



MINISTERIO DE
EDUCACIÓN PÚBLICA

GOBIERNO
DE COSTA RICA

Dirección de Educación
Técnica y Capacidades
Emprendedoras



»» Desarrollo de
Aplicaciones
de Software

Nivel
Undécimo

Juega para aprender

 “Motívate a la salud digital en tu aprendizaje”

Aprobado por el Consejo Superior de Educación, en la sesión 26-2025, acuerdo 202-26-2025 del 12/05/2025



DETCE
Dirección de Educación Técnica y Capacidades Emprendedoras



**MINISTERIO DE
EDUCACIÓN PÚBLICA**

**GOBIERNO
DE COSTA RICA**

Créditos

**Dirección de Educación
Técnica y Capacidades
Emprendedoras**

El Ministerio de Educación Pública (MEP), como autor del presente programa de estudio, se reserva los derechos morales y patrimoniales de esta obra, siendo responsabilidad de cualquier usuario o entidad reconocer esta condición para utilizar, reproducir o citar este programa y su texto.

Autoridades

José Leonardo Sánchez Hernández, Ministro de Educación Pública de Costa Rica.

Guiselle Alpízar Elizondo, Viceministra Académica.

Sofía Ramírez González, Viceministra Administrativa.

Dirección de Educación Técnica y Capacidades Emprendedoras (DETCE)

Alberto Calvo Leiva. Director de Educación Técnica y Capacidades Emprendedoras.

Joyce Mejías Padilla. Jefa Departamento de Especialidades Técnicas.

Rocío Quirós Campos. Jefa Unidad de Planificación y Diseño Curricular.



Equipo técnico

Elaboración del programa de estudio:

Heidy Cordonero Solano, Asesora Nacional de Informática Generalista.

Elaboration Subject Area English Oriented to Software Applications Development

Katherine Williams Jimenez, National English Advisor.

Coordinación general y revisión:

Rocío Quirós Campos, Jefa Unidad de Planificación y Diseño Curricular.

Fundamentación enfoque curricular del programa de estudio:

Rocío Quirós Campos, Jefa Unidad de Planificación y Diseño Curricular.

Validación de los elementos considerados en el diseño curricular:

Asesores Nacionales Unidad de Planificación y Diseño Curricular.



MINISTERIO DE
EDUCACIÓN PÚBLICA

GOBIERNO
DE COSTA RICA

Dirección de Educación
Técnica y Capacidades
Emprendedoras

Línea gráfica del formato utilizado en el programa de estudio.

Heidy Cordonero Solano, Asesora Nacional, DETCE.



Tabla de Contenidos

Presentación.....	10
Descripción de la Carrera Técnica.....	13
Modelo Pedagógico	17
Paradigma de la Complejidad.....	18
Humanismo	18
Racionalismo	19
Constructivismo Social.....	19
Educación para el Desarrollo Sostenible.....	27
Ciudadanía digital con equidad social.....	28
Ciudadanía planetaria con identidad nacional.....	28
Enfoque Curricular	38
Perfil de los Actores del Proceso de Aprendizaje	46
Estudiante	46
Competencia General	46
Competencias Específicas	47
Competencias Genéricas	47



Competencias para el Desarrollo Humano	49
Docente.....	52
<i>Diseño Curricular.....</i>	<i>55</i>
<i>Principios Didácticos y Estrategias Metodológicas para la Mediación Pedagógica</i>	<i>58</i>
<i>Orientaciones para la Realización de Actividades Pedagógicas Fuera de la Institución</i>	<i>71</i>
Práctica Profesional	72
Pasantía.....	72
Gira	73
Visita	73
<i>Planeamiento del Proceso de Aprendizaje</i>	<i>74</i>
Plan Anual.....	74
Plan de Práctica Pedagógica.....	76
<i>Evaluación del Proceso de Aprendizaje</i>	<i>80</i>
Trabajo Cotidiano.....	82
Tareas	82
Pruebas.....	83
Proyecto	84



Asistencia	84
.....	86
Estructura Curricular	87
Mapa Curricular	89
Malla Curricular.....	93
Subárea Emprendimiento e innovación aplicadas al desarrollo de aplicaciones de software.....	120
Descripción de la Subárea Emprendimiento e innovación aplicadas al desarrollo de aplicaciones de software	121
Subárea Desarrollo de componentes de software	142
Descripción de la Subárea Desarrollo de componentes de software	143
Subárea Bases de datos para proyectos de software.....	176
Descripción de la Subárea Bases de datos para proyectos de software.....	177
Subject Area English Oriented to Software Applications Development.....	200
Description.....	201
Rationale.....	205
Education for Sustainable Development	208
Global Citizenship with National Identity	208
Digital Citizenship with Social Equity	209
Common European Framework of Reference for Languages	211



<i>General Mediation Strategies and Pedagogical Approach</i>	213
<i>The Methodology Used in the Classroom</i>	222
<i>Curricular Design Template Elements</i>	225
Curriculum Template	228
<i>Planning</i>	230
Annual Learning Plan	230
Pedagogical Practice Plan	232
Task-Building Process	233
Pedagogical Practice Plan	239
<i>Evaluation of the Learning Process</i>	243
<i>Curricular Structure</i>	248
Curricular Design	261
Curricular Design	281
Curricular Design	302
Curricular Design	322
Curricular Design	339
Curricular Design	359



Curricular Design	395
References.....	413
Referencias Bibliográficas.....	421
Referencias Generales	421
Bibliografía complementaria	427
Apéndices	430
Glosario de Términos	431



Presentación

La Educación Técnica Profesional (ETP) es un subsistema del sistema educativo formal. Constituye un pilar en la preparación de técnicos y promueve el desarrollo social y económico del país a través de una oferta educativa flexible y dinámica. Proporciona igualdad de oportunidades en términos de acceso equitativo, no discriminatorio y ofrece dirección en dos sentidos: exploración vocacional en el Tercer Ciclo de la Educación General Básica (III Ciclo EGB) y formación en la carrera técnica seleccionada por la persona estudiante en Educación Diversificada.

De acuerdo con la Fundamentación Pedagógica de la Transformación Curricular (2015), la educación técnica tiene como uno de sus propósitos dar respuesta proactiva a la carencia de talento humano técnico nacional y mundial actual; “donde la educación es motor de cambio y catalizador para construir un mejor futuro, más sostenible y solidario” (p 15).

La ETP debe cumplir con un rol fundamental que faculte a las personas para la toma de decisiones informadas y asumir la responsabilidad de sus acciones individuales e incidencia en la colectividad actual y futura. Asimismo, el desarrollo de sociedades con integridad ambiental, viabilidad económica y justicia social – en el marco del respeto de la diversidad cultural y ética ambiental – cuya implementación debe ser el desarrollo de prácticas que posibiliten el aprovechamiento de las tecnologías de la información (TI) para disminuir la brecha social y digital.



En Costa Rica se visualiza la educación como un derecho humano y constitucional. El sistema educativo favorece la adquisición de conocimientos, habilidades y destrezas, valores y actitudes; además, promueve y estimula el desarrollo integral de las personas estudiantes y su participación en la sociedad civil y la vida económica del país.

La Dirección de Educación Técnica y Capacidades Emprendedoras (DETCE) es el órgano técnico del Ministerio de Educación Pública de la República de Costa Rica, responsable de promover programas de educación y formación de un talento humano especializado, cuya formación técnica y profesional sea el puente que potencie la vinculación con los mercados laborales o el emprendimiento.

Este programa de estudio favorece el desarrollo de procesos educativos con una estructura programática que incluye resultados de aprendizaje, de manera que la persona docente, como mediador pedagógico, pueda guiar en forma ordenada el proceso de construcción de conocimientos en el aula y el entorno, desarrolle competencias específicas, genéricas y para el desarrollo humano, con el propósito de que la persona estudiante se inserte exitosamente en el mundo laboral de la carrera técnica seleccionada o desarrolle su propio emprendimiento.

MACRO Currículum

Carrera técnica:

**Desarrollo de
aplicaciones de
software**

Componentes:

- Descripción de la carrera técnica.
- Fundamentación del modelo pedagógico.
- Enfoque curricular.
- Perfil de los principales actores del proceso de aprendizaje.
- Diseño curricular.
- Principios didácticos y estrategias metodológicas para la mediación pedagógica.
- Planificación de la mediación pedagógica.
- Evaluación de los aprendizajes.



Descripción de la Carrera Técnica

En la era de la hiperconexión, donde la tecnología ocupa un lugar protagónico en la vida diaria, surge la urgente necesidad de formar una nueva generación de técnicos que no solo dominen el desarrollo de aplicaciones de software, sino que también cultiven una relación sana, consciente y equilibrada con el mundo digital. Este programa de estudio nace con el propósito de que el estudiantado pueda aprender a aprender, de manera que puedan ser ciudadanos digitales felices, capaces de dejar una huella positiva en su propio aprendizaje y en el entorno que los rodea.

Aquí, cada línea de código se convierte en una herramienta para la creatividad, la expresión y la solución de problemas reales, pero también en una oportunidad para reflexionar sobre cómo la tecnología con un mal uso puede afectar la salud emocional, las relaciones sociales y la identidad.

Este programa invita a las y los estudiantes a decidir y disfrutar con temperancia; y así mantener una salud digital que potencie su capacidad de socializar en el mundo real, con empatía y propósito. Más que solo consumidores de tecnología, aspiramos a formar personas conscientes e íntegras, que puedan vivir plenamente en ambos mundos: el digital y el tangible, inspirados en la poderosa frase:

“Vive para ser libre y sé libre para vivir”, proponiendo un camino hacia el equilibrio y la innovación con sentido humano.

El propósito de la carrera técnica Desarrollo de aplicaciones de software, es la formación de técnicos en el nivel medio integrales (técnico 4 según el Marco Nacional de Cualificaciones de la EFTP) capaces de diseñar,



construir, probar y mantener aplicaciones de software innovadoras, éticas y funcionales, atendiendo las necesidades reales del entorno productivo, educativo y social. Este programa responde a las tendencias globales de transformación digital, automatización y economía basada en el conocimiento, y se fundamenta en estudios de la Unesco, el Foro Económico Mundial y el Banco Interamericano de Desarrollo (BID), que señalan al desarrollo de software como una de las habilidades clave del siglo XXI.

En el contexto nacional, informes del MICITT, CINDE, CAMTIC y el CONESUP respaldan la necesidad urgente de fortalecer la oferta de talento humano técnico especializado en tecnologías digitales, especialmente en programación, análisis de datos, infraestructura y experiencia de usuario.

El programa se organiza en subáreas que articulan el dominio técnico con habilidades transversales, potenciando la capacidad de las personas estudiantes para integrarse a equipos multidisciplinarios y emprender sus propios proyectos tecnológicos con habilidades lingüísticas.

En la subárea Herramientas para la producción de software, se abordan editores, sistemas de control de versiones, entornos de desarrollo integrado (IDE), automatización de pruebas, metodologías ágiles y plataformas de colaboración, esenciales para una producción eficiente y profesional.

La subárea Emprendimiento e innovación aplicados al desarrollo de aplicaciones de software, busca la adquisición de competencias para identificar oportunidades en el mercado digital, generar modelos de negocio sostenibles, aplicar pensamiento creativo y gestionar proyectos innovadores con impacto social y económico.



Con la implementación de la subárea Tecnologías de la información, se introducen conceptos fundamentales de hardware, redes, sistemas operativos y seguridad informática, habilitando al estudiantado en la comprensión de la infraestructura que sustenta el desarrollo de software moderno.

La subárea de Desarrollo de componentes de software incluye el uso de paradigmas de programación (orientado a objetos, funcional, eventos), diseño de algoritmos, estructuras de datos, documentación técnica y buenas prácticas de codificación para crear soluciones robustas y escalables.

La subárea Bases de datos para proyectos de software, cubre tanto bases de datos relacionales como NoSQL, diseño lógico y físico, consultas con SQL, modelado de datos, administración de bases de datos y su integración con aplicaciones mediante servicios o APIs.

Infraestructura y operaciones de servicios de software (DevOps), constituye una subárea que orienta la ejecución de prácticas de integración y entrega continua, contenedores, orquestación, monitoreo, automatización de despliegues y gestión en la nube, fortaleciendo la preparación para entornos de producción reales.

Experiencias de usuario y diseño de interacciones (UX/UI), tiene como objetivo explorar los fundamentos del diseño centrado en el usuario, la accesibilidad, prototipado, pruebas de usabilidad e interfaces gráficas, promoviendo productos que sean intuitivos, útiles y agradables para el usuario final.



Costa Rica se proyecta como un hub regional en tecnologías digitales. Sin embargo, según la Estrategia Nacional de Talento Humano 4.0 del MICITT y reportes de CINDE y CAMTIC, existe un déficit anual de miles de profesionales calificados en TI, especialmente en programación, desarrollo móvil y tecnologías emergentes.

La formación de técnicos en el nivel medio en desarrollo de aplicaciones de software responde directamente a cerrar esta brecha, facilitando la empleabilidad temprana, la continuidad académica en niveles universitarios y la posibilidad de emprender en el ecosistema digital. El enfoque por competencias y el uso de estándares internacionales de calidad garantizan una formación pertinente y alineada con las demandas de la Cuarta Revolución Industrial , (BRETE, 2025), (MICITT,2021), (CINDE, 2022) (BID, 2023).



Modelo Pedagógico

Las políticas educativa y curricular – aprobadas por el CSE – establecen el modelo educativo en el que se enmarcan los programas de estudio de la ETP. Al configurar las bases teóricas, las formas y los fines del aprendizaje, la persona docente y estudiante, el contexto y el saber se relacionan entre sí a partir del marco teórico de referencia que fundamenta el modelo pedagógico y el conjunto de intereses propios del contexto (social, institucional, individual y mercado) que median en el ejercicio de la educación o la formación de los individuos en la sociedad.

El modelo pedagógico concibe la educación como un proceso integral que se desarrolla a lo largo de la vida y favorece el progreso de la sociedad, facilitando la igualdad de condiciones de hombres y mujeres y el desarrollo pleno de sus potencialidades (Gómez et al., 2019).

El modelo pedagógico constituye el fundamento teórico y epistemológico que orienta y dirige el desarrollo de la educación, según contexto, guiando la acción en espacios áulicos. Desde el punto de vista inductivo, estos modelos y teorías se materializan mediante estrategias y acciones didácticas direccionadas a alcanzar los fines del aprendizaje, todo ello en el marco de la práctica en aula ejercida por las personas docentes.

En el caso del diseño curricular e implementación de los programas de estudio de la ETP, se sustentan en los pilares filosóficos establecidos en el modelo pedagógico planteado en la política educativa y curricular:



Paradigma de la Complejidad

Plantea que el ser humano es un ser autoorganizado y autorreferente; es decir, tiene conciencia de sí mismo y de su entorno, su existencia cobra sentido dentro de un ecosistema natural social- familiar y como parte de la sociedad. En cuanto a la adquisición de conocimiento, este paradigma toma en cuenta que las personas estudiantes se desarrollan en un ecosistema bionatural (que se refiere al carácter biológico del conocimiento en cuanto a formas cerebrales y modos de aprendizaje) y en un ecosistema social que condiciona la adquisición del conocimiento. El ser humano se caracteriza por tener autonomía e individualidad; establecer relaciones con el ambiente; poseer aptitudes para aprender, inventiva, creatividad, capacidad de integrar información del mundo natural y social y la facultad de tomar decisiones.

En el ámbito educativo, el paradigma de la complejidad permite ampliar el horizonte de formación, pues considera que la acción humana, por sus características, es esencialmente incierta, llena de eventos imprevisibles, que requieren que la persona estudiante desarrolle la inventiva y proponga nuevas estrategias para abordar una realidad que cambia a diario.

Humanismo

Se orienta hacia el crecimiento personal y por lo tanto aprecia la experiencia de la persona estudiante, incluyendo sus aspectos emocionales. Cada persona se considera responsable de su vida y de su autorrealización. La educación, en consecuencia, está centrada en la persona, de manera que sea ella



misma evaluadora y guía de su propia experiencia, a través del significado que adquiere su proceso de aprendizaje.

Cada persona es única, diferente; con iniciativa, con necesidades personales de crecer, con potencialidad para desarrollar actividades y solucionar problemas creativamente.

Racionalismo

El racionalismo se sustenta en la razón y en las verdades objetivas como principios para el desarrollo del conocimiento válido, ha sido fundamental en la conceptualización de las políticas educativas costarricenses.

Constructivismo Social

Propone el desarrollo máximo y multifacético de las capacidades e intereses de las personas estudiantes, según el aprendizaje en el contexto de una sociedad, tomando en cuenta las experiencias previas y las propias estructuras mentales de la persona que participa en los procesos de construcción de los saberes. Es parte y producto de la actividad humana en el contexto social y cultural donde se desarrolla la persona (CSE; MEP, 2016, p 8-10).

Los paradigmas epistemológicos fundamentan el modelo pedagógico y orientan los cambios pedagógicos desde el modelo conductista, centrado en la persona docente que enseña, a uno centrada en la persona estudiante. Este cambio requiere de un cambio fundamental en el papel del educador, desde un docente



transmisioncita a uno facilitador del aprendizaje. En este sentido, su función será orientar, guiar, moderar y facilitar el aprendizaje acudiendo al estudiantado y ofreciéndoles información cuando la necesitan. Su rol principal pasa de ser un protagonista, a ofrecerle al estudiantado diversas oportunidades de aprendizaje, colaborando con estos para que piensen de forma crítica, argumenten y reflexionen.

La persona estudiante dejará su papel pasivo, en el cual recibía información y luego memorizaba, pero de manera simultánea olvidaba rápidamente. El modelo establece que el estudiantado asuma un papel activo, que lo motive a aprender más, integrar los conocimientos, tener una actitud receptiva hacia el intercambio de ideas, compartir información y aprender de los demás, ser autónomo en el aprendizaje y trabajar con diferentes grupos gestionando los posibles conflictos que surjan (De Zubiría, J.2010).

La comparación entre el modelo conductista y el constructivismo social se presentan en la Tabla 1, según el objetivo del aprendizaje, el rol de la persona docente y estudiante, los contenidos, la metodología, los recursos educativos y la evaluación.



Tabla 1

Comparación entre los modelos pedagógicos conductista y constructivismo social

Aspectos por considerar	Modelo conductista	Modelo constructivismo social
Objetivo del aprendizaje	Plantea objetivos generales y específicos para la medición de los alcances y la obtención de cambios observables en el comportamiento de la persona estudiante.	Centrado en la construcción de los aprendizajes a través de la interacción social y la construcción conjunta del conocimiento.
Rol del estudiante	Pasivo, receptivo y orientado a la repetición para memorizar y repetir la conducta requerida por la persona docente.	Activo, participativo y protagonista en la construcción de su propio proceso de aprendizaje.
Rol del docente	Sujeto activo del proceso de aprendizaje, proveedor del conocimiento y creador de resultados de aprendizaje orientados a la repetición y memorización.	Facilitador del aprendizaje, promotor de la interacción social y autonomía del estudiante, diseñador de experiencias de aprendizaje y modelo de pensamiento crítico y metacognición.



Aspectos por considerar	Modelo conductista	Modelo constructivismo social
Contenidos	Tienden a ser estructurados y secuenciales, con un enfoque en la práctica repetitiva y el refuerzo de los comportamientos deseados.	Su selección y diseño fomentan la construcción activa del conocimiento del estudiante, a través de la interacción social y la participación en experiencias significativas y auténticas de aprendizaje.
Metodología	Rígida, poco flexible y emplea la enseñanza instruccional y programada. El aprendizaje se logra cuando se demuestra una respuesta apropiada ante un estímulo ambiental específico.	Emplea estrategias dirigidas a la construcción del conocimiento, como la resolución de problemas, la cual promueve el desarrollo de un aprendizaje significativo y el pensamiento crítico.
Recursos educativos	Se utiliza el material didáctico estructurado, ejercicios de práctica, pruebas y evaluaciones, modelos y ejemplos, programas de computadora y software educativo, refuerzos positivos, entre otros.	Proyectos colaborativos, aprendizaje basado en problemas, entornos de aprendizaje colaborativos, aprendizaje por descubrimiento, narrativas y cuentos, realimentación formativa, debates, otros.



Aspectos por considerar	Modelo conductista	Modelo constructivismo social
Evaluación	Parte de que todas las personas estudiantes son iguales, por lo que reciben la misma información; centrada en el logro de los objetivos, con predominio de la prueba escrita y oral para medir conocimientos y recopilar evidencias del rendimiento.	Se concibe como un proceso integral que va más allá de simplemente medir el conocimiento, sino para comprender cómo el estudiantado lo construye a través de la interacción social y la participación en experiencias significativas.

Los elementos del constructivismo social, aportados por Lev Vigotsky, proporcionan el marco referencial del modelo pedagógico seleccionado para el diseño e implementación de los planes de estudio propuestos para la ETP. A continuación, el detalle:

- toma en cuenta el nivel de desarrollo; es decir, el o la estudiante posee una zona de desarrollo real definida como las acciones que el estudiantado se encuentra en capacidad de desarrollar de forma independiente. En este sentido, resulta relevante destacar la importancia de la función diagnóstica de la evaluación en el proceso de aprendizaje, pues su aplicación nos permite obtener la información de la zona de desarrollo real con la que inician las personas estudiantes el nivel educativo.
- fomenta un rol activo del estudiantado en su aprendizaje, ya que no posee un rol pasivo respecto al proceso de su desarrollo, sino que es él quien, estimulado por el medio, compone y construye su propio



tejido, conceptual y simbólico, y desarrolla así las propias condiciones de su aprendizaje. Actúa sobre la realidad, la transforma y es transformado por ella.

La importancia de esta característica se acrecienta con la naturaleza de la ETP, pues durante el proceso de formación la persona estudiante tiene la oportunidad de aprender en entornos reales de trabajo, mediante la exposición a tareas auténticas, la estimulación del medio al que se ve expuesto durante la implementación de visitas técnicas, giras, pasantías y el desarrollo de la práctica profesional. Esto le permite ser artífice de su propio conocimiento y transformar su espacio.

- enfatiza la importancia de la interacción de la persona estudiante con el entorno y su relación con otros, ya que el factor social juega un papel determinante en la construcción del conocimiento. Desde la óptica de la ETP este aspecto es preponderante, ya que uno de sus fines es el desarrollo de competencias que le permitan al estudiante vincularse con éxito al mercado laboral. Cabe mencionar que las necesidades de los sectores productivos se caracterizan por ser dinámicas, vertiginosas y con un fuerte impacto ocasionado por el desarrollo de la inteligencia artificial, la revolución 4.0, la automatización y el uso de la tecnología.

En el contexto actual de la ETP, resulta imprescindible una mediación pedagógica que privilegie el contacto de las personas estudiantes con el entorno laboral, con el fin de promover el aprendizaje basado en actividades realistas que demanden el uso de herramientas y tecnología, la motivación en entornos empresariales y la experiencia de brindar solución a problemas del mundo real o laboral específico.



Adicionalmente, se debe considerar la construcción del conocimiento como parte de la interacción social con las personas y muy especialmente, el papel que ejercen algunos actores clave que participan del proceso educativo de este subsistema.

Evidentemente, la enseñanza de una carrera técnica debe tener lugar en el contexto de problemas del mundo real o de la práctica profesional. La mediación pedagógica seleccionada debe promover el autoaprendizaje y la ejecución de estrategias colaborativas y cooperativas, así como potenciar situaciones de aprendizaje lo más cercanas posibles al futuro contexto profesional del estudiantado. Para tal efecto, se deben brindar espacios donde las personas estudiantes se enfrenten a problemas reales, con un nivel de dificultad y complejidad similares al entorno laboral.

Así mismo, es importante indicar la importancia de los recursos educativos y la función de la persona docente. Constituyen el “andamiaje” de apoyo para la conducción del aprendizaje e independencia del estudiantado. Sin duda alguna, la educación dirigida a preparar a las personas para el mundo del trabajo requiere de recursos que brinden el soporte adecuado para el alcance de las competencias requeridas por el mercado laboral.

En este aspecto, la persona docente debe considerar con detenimiento las necesidades particulares de sus estudiantes, observar sus diferencias conceptuales, ritmos y estilos de aprendizaje, su inclusión y capacidades excepcionales. Del mismo modo, conforme la persona estudiante se vuelve más diestra, el o la docente retiran el andamiaje para que se desenvuelva de manera independiente.



Cabe considerar que, desde los fundamentos que plantea el constructivismo social, es de vital importancia el desarrollo de actividades y apoyos por parte del profesorado. Si analizamos la relación teórico-práctica que caracteriza la ETP, orientada a la adquisición de conocimientos, habilidades, destrezas, valores y actitudes en un campo profesional específico, la asistencia y soporte educativo del docente promueve que el estudiantado adquiera más posibilidades de actuación autónoma ante situaciones y tareas nuevas, cada vez más complejas.

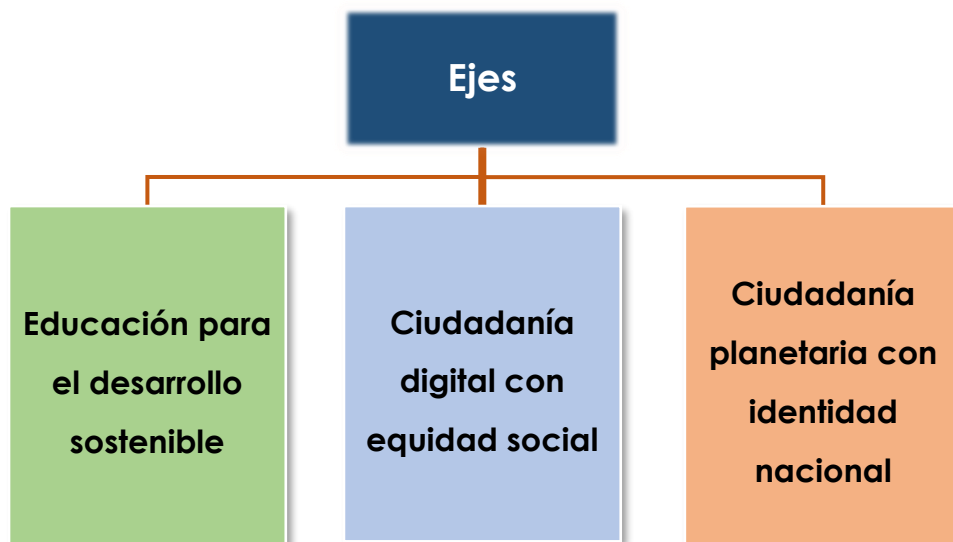
Este acompañamiento, por parte de la persona docente, es trascendental en el proceso educativo de una carrera técnica, ya que, durante la mediación pedagógica y la ejecución de visitas técnicas, giras, pasantías y prácticas profesionales en la empresa, las personas estudiantes pueden utilizar equipos, herramientas y tecnología en general, como parte de los recursos que brinda el andamiaje al proceso educativo, mediado con la supervisión y seguimiento de expertos.

En concordancia con los elementos que integran el modelo pedagógico, el diagrama 1 presenta los ejes transversales del diseño curricular, los cuales permean el plan de estudio propuesto y las situaciones desarrolladas en el contexto educativo.



Diagrama 1

Ejes de la política educativa y curricular del Ministerio de Educación Pública



Educación para el Desarrollo Sostenible

Este eje torna a la educación en la vía de empoderamiento de las personas, a fin de que tomen decisiones informadas, asuman la responsabilidad de sus acciones individuales e incidencia en la colectividad actual y futura. En consecuencia, contribuyan al desarrollo de sociedades con integridad ambiental, viabilidad económica y justicia social para las presentes y futuras generaciones.



Ciudadanía digital con equidad social

Eje que busca el desarrollo de un conjunto de prácticas orientadas a la disminución de la brecha social y digital, mediante el uso y aprovechamiento de las tecnologías digitales (CSE; MEP, 2016, p 10-12).

Ciudadanía planetaria con identidad nacional

Fortalece la toma de conciencia de la conexión e interacción inmediata que existe entre personas y ambientes en todo el mundo, así como la incidencia de las acciones locales en el ámbito global y viceversa. Además, implica retomar nuestra memoria histórica, con el propósito de ser conscientes de quiénes somos, de dónde venimos y hacia dónde queremos ir.

Desde la perspectiva de una educación enfocada en competencias, se integran las cuatro dimensiones que promueve la Transformación Curricular: Educar para una nueva ciudadanía (2015):

Formas de pensar: se refiere al desarrollo cognitivo de cada persona, por lo que implica las competencias relacionadas con la generación de conocimiento, la resolución de problemas, la creatividad y la innovación.

Formas de vivir en el mundo: conlleva el desarrollo sociocultural, las interrelaciones que se tejen en la ciudadanía global con el arraigo pluricultural y la construcción de los proyectos de vida.

Formas de relacionarse con otros: asociado con el desarrollo de puentes que se tienden mediante la comunicación y lo colaborativo.



Herramientas para integrarse al mundo: relacionado con la apropiación de las tecnologías digitales y otras formas de integración, así como la atención que debe prestarse al manejo de la información (MEP, 2015, p 33-37).

Adicionalmente, resulta imprescindible que la ETP – como pilar fundamental para la equidad, productividad y sostenibilidad del país – contribuya a la mejora de acceso igualitario a la educación, empleo, emprendimiento y trabajo decente.

Los elementos de mayor relevancia del modelo pedagógico de la ETP son: las políticas educativas vigentes, la gestión curricular y administrativa, el rol de la persona estudiante y docente y la mediación pedagógica.

Políticas educativas

Las políticas educativas se fundamentan en los pilares epistemológicos, los ejes, los principios y las dimensiones establecidas en las políticas educativas vigentes aprobadas por el CSE. Plantean un modelo educativo integral, humanista, racionalista y complejo, basado en el constructivismo social, sin dejar de lado la importancia de la aplicación de las normas técnicas.

Además, promueven la inclusión, la equidad de género, la creatividad, la innovación, la reflexión, el pensamiento crítico, el multilingüismo, las capacidades emprendedoras y el compromiso con la sostenibilidad, la sociedad costarricense y la ciudadanía planetaria y digital.



Gestión curricular

Los planes de estudio se diseñan con un enfoque por competencias desde la perspectiva formativa. Consideran el saber saber, saber hacer (estado del arte de la técnica), saber ser y saber convivir con los demás.

El diseño curricular parte de los estándares de cualificación, los cuales se implementan con una metodología basada en el análisis del contexto educativo y laboral – establecida por el Marco Nacional de Cualificaciones de la Educación y Formación Técnica Profesional de Costa Rica (MNC-EFTP-CR). La metodología brinda información de los requerimientos del sector productivo al que pertenece la cualificación, tanto en el contexto nacional como internacional.

La gestión curricular promueve una oferta educativa que responde a las necesidades de los sectores productivos, favorece la empleabilidad y la continuidad de los estudios en educación superior, en concordancia con los continuos avances de la tecnología, la inteligencia artificial y el impacto de la revolución 4.0. Por otra parte, promueve la gestión del talento humano docente, desarrollando las capacidades requeridas para el alcance de las competencias del estudiantado, según contexto.

Gestión administrativa

La gestión administrativa promueve la articulación de los actores que integran el Sistema Nacional de Educación y Formación Técnica Profesional (SINETEP) y establece alianzas estratégicas entre los diversos



actores de la EFTP (Educación y Formación Técnico Profesional). Asimismo, gestiona los recursos financieros necesarios para dotar a las personas estudiantes que así lo requieran, de incentivos económicos (becas), servicios de alimentación y transporte que garanticen su permanencia y éxito educativo.

Cabe mencionar que también promueve el desarrollo de procesos de formación en las personas docentes, de acuerdo con las necesidades del contexto.

Mediación pedagógica

Este elemento del modelo pedagógico de la ETP propone estrategias pedagógicas centradas en el aprendizaje, promueve que la persona estudiante construya conocimiento de forma autónoma – mediante su relación con otros colaboradores. Debe señalarse que también potencia el abordaje metodológico orientado a la acción mediante la implementación de metodologías activas, centradas en el estudiantado y caracterizadas por concebir el aprendizaje como proceso, y no únicamente como una recepción y acumulación de información.

En lo esencial, plantea que las actividades se basan en la interacción de la persona estudiante con los demás, el entorno y la cultura, estableciendo aprendizajes como consecuencia de su desarrollo y su relación con otros.

Resulta claro que plantea el desarrollo de actividades complejas requeridas para la vida y el mundo del trabajo, mediante la planificación y el diseño de situaciones de aprendizaje auténticas. Se considera



relevante para la implementación de la mediación pedagógica la aplicación de proyectos, simulaciones y experimentación activa.

La simulación es una técnica que permite recrear situaciones, establecer la factibilidad de un experimento y visualizar a un sistema físico, haciendo una conexión entre lo abstracto y la realidad. Evidentemente, generan un ambiente de aprendizaje interactivo, lo que permite a las personas estudiantes explorar la dinámica de un proceso.

En el caso de la experimentación activa, el estudiantado aprende y desarrolla capacidades a través de la experiencia en el mundo real. El aprendizaje constituye el proceso por el que se crea conocimiento mediante la transformación de la experiencia. Se fundamenta en la idea de que el conocimiento se produce a través de las acciones provocadas por una experiencia concreta, la cual se transforma en una conceptualización abstracta y permite aplicarse a nuevas situaciones, formando un proceso continuo e interactivo que genera nuevos aprendizajes.

La experimentación activa propicia el aprendizaje mediante el diseño de experimentos en laboratorio y la empresa. En este sentido, no basta con una experiencia para producir conocimiento, es necesaria la modificación de las estrategias cognitivas de la persona estudiante. Por lo tanto, la experiencia cobra sentido cuando se vincula con el conocimiento previo y se desarrollan andamiajes conceptuales que permitan aplicar el nuevo conocimiento a nuevas situaciones.



En el caso del proyecto como estrategia de aprendizaje, promueve que el estudiantado asuma una mayor responsabilidad de su propio aprendizaje y las competencias adquiridas en el proceso educativo para ser aplicadas en situaciones del contexto real. El proyecto facilita que la persona estudiante vivencie experiencias de aprendizaje para rescatar, comprender y aplicar los aprendizajes adquiridos, como herramienta para resolver problemas o proponer mejoras en el entorno en donde se desenvuelven. Así mismo, propicia que él o la estudiante se involucren en la solución de problemas y otras tareas significativas, permitiéndole trabajar de manera autónoma en la construcción de su propio aprendizaje.

En relación con la idea anterior, el proyecto impulsa la motivación en el estudiantado. Por ejemplo, cuando participa en actividades con una clara importancia en entornos empresariales y en los que se le facilita la aplicación de su aprendizaje, en la solución a problemas del mundo real o de un entorno laboral específico.

Por último, es conveniente acotar que el proyecto, en ambientes de aprendizaje de entornos reales de trabajo, permite al estudiante la utilización de equipos, recursos educativos tecnológicos, insumos, herramientas y otros de la empresa formadora.

Rol de la persona estudiante

La persona estudiante es el responsable directo en la construcción del conocimiento y cumple un papel activo y protagonista en el aprendizaje. De esta forma, demuestra capacidades para trabajar en equipo, argumentar, resolver problemas, respetar las ideas de otros, interactuar con otros y con su entorno para la construcción de aprendizajes significativos.



El o la estudiante crea y conduce su propia experiencia de aprendizaje, investiga y explora por sí mismo, comprometiéndose con la resolución de problemas reales y de su medio más cercano. En este sentido, asume con compromiso la actividad intelectual necesaria para la construcción del conocimiento.

Desde la perspectiva más general, la persona estudiante desarrolla capacidades de autorregulación y metacognición, que le permiten reflexionar sobre lo que sabe y cómo aprende. El propósito es que sea consciente de sí mismo como aprendiz, de forma que sea capaz de controlar la cognición y motivación para mejorar su aprendizaje. Las personas estudiantes autorreguladas saben cómo planificar eficazmente su aprendizaje y cómo monitorear su comprensión de forma eficiente, saben cuándo no entienden, tienen estrategias que les permite revisar y corregir los aspectos que no han comprendido y también cómo evaluar su aprendizaje con precisión y eficacia.

Por consiguiente, comparte conocimientos, habilidades, destrezas, valores y actitudes con él o la docente y el estudiantado, propiciando situaciones de aprendizaje multidireccionales y dinámicas, que surgen de su interacción con el entorno empresarial.

Rol de la persona docente

La persona docente es responsable de guiar y orientar el proceso de aprendizaje, promover la innovación, el desarrollo y autonomía del estudiantado, así como enseñar a aprender a aprender, mediante estrategias que estimulen la creatividad, favorezcan el movimiento, la exploración, la construcción y la motivación, en respuesta a la mediación pedagógica.



Se encarga de mantener comunicación con la coordinación con la empresa del centro educativo y el sector empresarial, en relación con el desempeño del estudiante durante el desarrollo de actividades pedagógicas fuera del centro educativo. Adicionalmente, brinda y da seguimiento a los apoyos educativos que en materia de estrategias metodológicas y de evaluación requiera la persona estudiante.

Resulta claro que la persona docente guarda confidencialidad de la información de carácter industrial o comercial, a la que tenga acceso durante el desarrollo de actividades pedagógicas fuera del centro educativo.

El o la docente propicia el desarrollo de emociones positivas en la clase o más específicamente, motiva a través de la curiosidad, la indagación y el papel activo del estudiante como insumo fundamental para el logro de la atención ejecutiva, la formación de nuevas redes neuronales (neuro plasticidad) y la consolidación de memorias de largo plazo. Todo ello en concordancia con lo derivado de investigaciones actuales en el ámbito de las neurociencias cognitivas.

Se plantea la necesidad de que la persona docente promueva el aprendizaje autorregulado y maximice el compromiso cognitivo del estudiantado, comprendiendo la naturaleza de las actividades de aprendizaje propuestas y los lineamientos utilizados al presentar esas actividades de aprendizaje. Además, debe realizar el proceso de evaluación diagnóstica, formativa y sumativa.



Rol del centro educativo

Es el responsable de propiciar mecanismos para la planificación y el financiamiento de la ETP, disponer de infraestructura, equipamiento, herramientas e insumos que faciliten el mejoramiento y fortalecimiento de la calidad del servicio educativo y la mediación pedagógica de las carreras técnicas, en concordancia con las demandas del contexto.

Al centro educativo le corresponde establecer comunicación con los sectores productivos para el desarrollo de visitas técnicas, giras, pasantías y prácticas profesionales, así realimentar el proceso educativo. Además, promover y supervisar el desarrollo de la evaluación educativa y la mediación pedagógica de calidad, de conformidad con lo establecido en las políticas educativas y normativas vigentes.

Se encarga de establecer puentes de comunicación efectivos con la persona encargada del estudiante e implementar protocolos que aseguren su éxito académico y permanencia en el centro educativo. Por otra parte, gestionar los procesos administrativos con otras dependencias del MEP que garanticen el funcionamiento de la institución educativa, los mecanismos de control y seguimiento requeridos.

En otro orden de ideas, es importante recalcar que el diseño curricular de los programas de estudio responde a las necesidades de la EFTP demandadas por el contexto laboral actual. En el marco de la atención de las recomendaciones dadas al país por la Organización para la Cooperación y el Desarrollo Económicos (OCDE), se implementa el MNC-EFTP-CR, el cual constituye la estructura reconocida nacionalmente, responsable de



normar las cualificaciones y las competencias asociadas a partir de un conjunto de criterios técnicos contenidos en los descriptores.

Cabe resaltar que por primera vez los planes de estudio de las carreras técnicas tienen los estándares de cualificación como uno de sus insumos, por lo que una vez que se implementen, el diploma de técnico en el nivel medio tendrá equivalencia con el nivel de cualificación 4, establecido en el MNC-EFTP-CR.



Enfoque Curricular

Las nuevas tendencias que hoy caracterizan la organización del mercado de trabajo y la demanda de nuevos perfiles profesionales, en el marco de la globalización económica y de la sociedad de la información y el conocimiento, provocaron una transformación en materia de conocimientos, habilidades, destrezas, valores y actitudes requeridos por el talento humano técnico, el cual representa uno de los perfiles de mayor demanda según los empleadores, tanto en el mercado laboral, nacional como internacional.

Posiciones especializadas como técnicos, representantes de ventas, electricistas, mecánicos, personal de apoyo de oficina e ingenieros se han clasificado entre los primeros cinco puestos más difíciles de cubrir en los últimos diez años en Costa Rica. La escasez de talento humano disponible y la falta de competencias técnicas y competencias para el desarrollo humano son las principales razones por las que los empleadores no encuentran el talento adecuado a sus organizaciones (Manpower Group, 2018).

Por otra parte, el Banco Mundial, la Organización Internacional del Trabajo (OIT) y la UNESCO (2023) son del criterio que las tendencias asociadas a la Industria 4.0 inciden en la demanda de competencias, la distribución de oportunidades económicas, la evolución laboral de los mercados, el progreso tecnológico, la inteligencia artificial, la transformación demográfica y el cambio climático. Ante este panorama, se requiere una ETP de calidad para garantizar la transición exitosa al mercado laboral.

Otro factor importante que impacta la ETP es la inteligencia artificial, una de las áreas de la tecnología que más cambios vertiginosos ha provocado en la vida social, económica y cultural de las personas y los países.



Su papel es relevante, pues forma parte de la preparación requerida por las personas estudiantes para enfrentar el dinámico mundo del trabajo, contribuir al empleo y la productividad.

De la misma forma, la pandemia provocada por el COVID-19 aceleró el desarrollo de competencias digitales de la EFTP, trayendo consigo oportunidades, pero también evidenciando las limitaciones que deben superarse para que estas innovaciones alcancen todo su potencial y contribuyan a la resiliencia del sistema ante futuras interrupciones.

El enfoque por competencias – desde la corriente o perspectiva formativa – tiene un respaldo epistemológico vinculado al constructivismo, neoconstructivismo, cognitivismo y social constructivismo. Constituye uno de los factores principales para dinamizar la economía nacional y reconoce que las personas aprenden a construir el sentido de su existencia mediante hechos y experiencias ya existentes, lo que permite elaborar nuevos conocimientos.

El enfoque por competencias, basado en la perspectiva social constructivista, demanda una vinculación directa con el desarrollo integral de las personas. El aprendizaje de una competencia no puede aislarse del desarrollo de la persona, su comunidad o entorno laboral-social. Bajo esta corriente se reconoce que el conocimiento se construye a partir de la propia experiencia de quien aprende, de la información que recibe y la manera como lo procesa, coteja, integra, reconstruye e interpreta, pero, sobre todo, de cómo la comparte con los demás.



En el enfoque por competencias se busca que la persona estudiante desarrolle sus propias aptitudes o capacidades con la intención de alcanzar un desarrollo integral a lo largo de la vida, que le permita insertarse exitosamente en el sector empleador o continuar estudios de educación superior. Según López (2016) “La palabra competencia es de naturaleza polisémica, por lo que su abordaje requiere precisar la perspectiva de su enfoque, ya que actualmente es común encontrar una gran variedad de clasificaciones (p. 43).

Dentro de este marco del enfoque por competencias, Ramírez (2020) considera que:

trasciende el planteamiento educativo tradicionalista que privilegiaba la habilidad memorística, de modo que afronta a las personas a aplicar el conocimiento en distintas situaciones; valida el aprendizaje como un proceso escalonado e integral en la que los errores forman parte; da énfasis a procesos más integrales en los que para la adquisición y asimilación de saberes se integran al saber conocer, el saber hacer, saber ser y el saber convivir. (p. 5)

En relación con la idea anterior, Jacques Delors planteó que la educación debe estructurarse en torno a cuatro aprendizajes fundamentales que en el transcurso de la vida serán para cada persona, en cierto sentido, los pilares del conocimiento: aprender a conocer, es decir, adquirir los instrumentos de la comprensión; aprender a hacer, para poder influir sobre el propio entorno; aprender a vivir juntos, para participar y cooperar con los demás en todas las actividades humanas; aprender a ser, un proceso fundamental que recoge elementos de los tres anteriores. Por supuesto, estas cuatro vías del saber convergen en una sola, ya que hay entre ellas múltiples puntos de contacto, coincidencia e intercambio (Delors, 1994).



Para hacer posible el desarrollo en la vida de las personas, su proceso de formación deberá estar asociado, no solo en la adquisición de datos e información, sino en la articulación e integración de los saberes o aprendizajes: saber conocer, saber hacer, saber estar y saber ser.

Las competencias nos remiten a la acción. Para Perrenoud (2008) “Una competencia es concebida como la capacidad de movilizar varios recursos cognitivos para hacer frente a un tipo determinado de situaciones”. Roegiers (2010) las “considera como un conjunto ordenado de capacidades (actividades) que se ejercen sobre los contenidos en una categoría determinada para resolver los problemas planteados por estos (López, p. 67).

Las competencias movilizan saberes, maneras de hacer y actitudes; cuando la persona tiene la competencia, en ese momento actualiza lo que sabe en un contexto singular. En este sentido, es importante contemplar la motivación como elemento presente en el desarrollo de las competencias, pues es considerada como una dimensión humana basada en el aprender. Es decir, la persona estudiante motivada ensaya comportamientos adecuados ante experiencias distintas, pues a partir de los errores cometidos previamente, evade las respuestas que no surtieron efecto en situaciones específicas y replica aquellas con resultados exitosos (Ramírez, 2020).

Por consiguiente, cuando se habla del desarrollo de competencias se hace una alusión directa al aprendizaje. Desde esta perspectiva, la investigación actual en el ámbito de las neurociencias cognitivas deja en claro que el desarrollo de emociones positivas en la clase o más específicamente de la motivación, a través de la



curiosidad, la indagación y el papel activo de los educandos, constituye un insumo fundamental para el logro de la atención ejecutiva, la formación de nuevas redes neuronales (neuro plasticidad) y la consolidación de memorias de largo plazo, todos ellos considerados como procesos inherentes al aprendizaje.

De acuerdo con estas ideas, queda claro que una competencia puede ser definida como el saber en la acción (López, 2016). Castillo y Cabrerizo (2010) definen una competencia como:

...la capacidad de aplicar los conocimientos -lo que se sabe- junto con las destrezas y habilidades -lo que se sabe hacer- para desempeñar una actividad profesional, de manera satisfactoria y en un contexto determinado, de manera satisfactoria -sabiendo ser- uno mismo y sabiendo estar con los demás. (p. 64)

Tobón (2007) define las competencias como:

... procesos complejos de desempeño con idoneidad en determinados contextos, integrando diferentes saberes (saber ser, saber hacer, saber conocer y saber convivir), para realizar actividades y/o resolver problemas con sentido de reto, motivación, flexibilidad, creatividad, comprensión y emprendimiento, dentro de una perspectiva de procesamiento metacognitivo, mejoramiento continuo y compromiso ético, con la meta de contribuir al desarrollo personal, la construcción y afianzamiento del tejido social, la búsqueda continua del desarrollo económico-empresarial sostenible, y el cuidado y protección del ambiente y de las especies vivas. (p. 17)



Esta definición muestra seis aspectos esenciales en el concepto de competencias desde el enfoque complejo: procesos, complejidad, desempeño, idoneidad, metacognición y ética. Significa que en cada competencia se hace un análisis de alguno de los aspectos centrales para orientar el aprendizaje y la evaluación, lo cual tiene implicaciones en la didáctica, así como en las estrategias e instrumentos de evaluación.

Tobón (2007) menciona que las competencias son un enfoque para la educación y no un modelo pedagógico. Son un enfoque porque solo se focalizan en determinados aspectos conceptuales y metodológicos de la educación y la gestión del talento humano; por ejemplo: 1) integración de saberes en el desempeño, como el saber ser, el saber hacer, el saber conocer y el saber convivir; 2) construcción de los programas de formación acorde con la filosofía institucional y los requerimientos disciplinares, investigativos, laborales, profesionales, sociales y ambientales; 3) orientación de la educación por medio de criterios de calidad en todos sus procesos; 4) énfasis en la metacognición en la didáctica y la evaluación de las competencias; y 5) empleo de estrategias e instrumentos de evaluación de las competencias mediante la articulación de lo cualitativo con lo cuantitativo (p. 18-19).

Por su parte, Estévez y Robles (2013) definen la competencia “como la capacidad de poner en movimiento (aplicar) conocimientos (saberes), habilidades (saber hacer) y actitudes (implica valores) de modo pertinente para resolver problemas o realizar tareas en contextos y situaciones específicas” (p. 8).



Al trabajar bajo un enfoque por competencias, lo primero que se deberá aclarar son las metas o propósitos propuestos. Cuando el docente planea es fundamental que fije las metas, determine los resultados esperados e identifique el tipo de competencias por desarrollar.

Para Adam (2004) los resultados de aprendizaje:

... son enunciados acerca de lo que se espera que el estudiante sea capaz de hacer, comprender o demostrar una vez terminado un proceso de aprendizaje. Describen de manera integrada los conocimientos, habilidades y actitudes que los estudiantes adquirirán en un proceso de formación. Dichos resultados deben ser observables o medibles, y se redactan usando un verbo dinámico, es decir que se refiere a una acción, no a un estado. (p. 19)

En relación con el contexto de la ETP y hacia dónde se dirige la formación, Muñoz (2012) es del criterio que “el enfoque por competencias se concentra en el desarrollo de una formación técnica, que las personas la puedan desarrollar de manera eficiente y eficaz y en perspectiva de competitividad y de innovación científico/tecnológica o de gestión técnica y algorítmica del conocimiento” (p. 21).

El enfoque por competencias, propuesto en este programa de estudio, considera como parte de los elementos del diseño curricular el desarrollo de competencias específicas, genéricas y para el desarrollo humano.



Las competencias específicas tienen que ver con el conocimiento concreto de cada área temática o campo disciplinar. Las competencias genéricas constituyen parte del dominio que él o la estudiante debe tener sobre el conjunto de conocimientos teóricos necesarios que sustentan el campo disciplinar incluyendo funciones cognitivas, metodológicas, tecnológicas y lingüísticas. Las competencias para el desarrollo humano se refieren a la capacidad de mantener una óptima relación social y están vinculadas con la cooperación al llevar a cabo proyectos comunes o de autoconocimiento. Así mismo se vinculan con la capacidad de alcanzar una visión de conjunto e implican la comprensión, conocimiento y sensibilidad de las personas. Se le considera como la capacidad de actuar de manera flexible y disposición del cambio ante la presencia de nuevas situaciones (López, 2017, p. 40).



Perfil de los Actores del Proceso de Aprendizaje

Estudiante

Bajo el enfoque por competencias y los fundamentos establecidos en las políticas educativas y directrices emanadas por el CSE, en materia de ETP, se espera que cada estudiante, al finalizar su proceso formativo en la carrera técnica, desarrolle las siguientes competencias:

Competencia General

Se sustenta en el estándar de cualificación que sirvió de insumo para la elaboración del programa de estudio. Describe la función principal del técnico en el nivel medio, según el campo disciplinar en el que se educó. Este parte del análisis del contexto educativo y laboral y de la información suministrada por informantes clave y fuentes de información nacionales e internacionales.

- Desarrollar aplicaciones de software, utilizando diferentes paradigmas de programación, patrones, técnicas, herramientas estándar de diseño y bases de datos, considerando requerimientos acordados con el cliente, con ética y profesionalismo, coordinando con el personal cualificado la solución de problemas en un ambiente de sana convivencia.



Competencias Específicas

Relacionadas con el conocimiento concreto de cada área temática o campo disciplinar.

- Realizar el desarrollo y actualización de componentes de software, utilizando plataformas y lenguajes de programación vigentes, mediante la aplicación de marcos de referencia, según criterios de seguridad y requerimientos técnicos establecidos.
- Aplicar principios de comunicación entre componentes de software, mediante la aplicación de marcos de referencia, según criterios de seguridad y requerimientos técnicos establecidos.
- Determinar necesidades de infraestructura en el despliegue y gestión de los servicios, según requerimientos técnicos establecidos.
- Realizar la gestión del ciclo de vida de bases de datos en proyectos de software mediante el uso de lenguajes de programación asociados a la base de datos y aplicación de marcos de referencia considerando criterios de seguridad y requerimientos técnicos establecidos.
- Aplicar técnicas y herramientas estándar en el diseño de interacciones y experiencias de usuario de los componentes de software, considerando disposiciones de privacidad, seguridad y normativa vigente.

Competencias Genéricas

Constituyen parte del dominio que la persona estudiante debe tener sobre el conjunto de conocimientos teóricos necesarios que sustentan el campo disciplinar.



- Identificar oportunidades de negocios y aplica metodologías para la construcción de modelos de negocios.
- Elaborar planes de negocios aplicando metodologías vigentes en el mercado.
- Desarrollar las etapas correspondientes para la creación de empresas de práctica y de su proyecto de vida, tomando en consideración sus competencias, recursos, el entorno y su compromiso local y social.
- Utilizar herramientas y tecnologías digitales mediante la aplicación de software de código abierto y licenciado, la automatización y el análisis de datos y su transmisión a través del Internet; así como la evaluación de alternativas para la protección e integridad de los datos mediante el uso de tecnologías.
- Promover y verificar acciones que respondan a la normativa ambiental.
- Aplicar las normas de salud ocupacional, según protocolos establecidos.
- Aplicar normas de aseguramiento de la calidad establecidas a nivel nacional e internacional.
- Coordinar acciones con equipos de trabajo, de manera asertiva y propositiva.
- Proponer soluciones creativas e innovadoras a procesos específicos del campo de formación técnica.
- Demostrar habilidad y destreza en las tareas propias de la carrera.
- Comprender, interpretar y comunicar información técnica propia de su campo de formación.



- Dirigir procesos de producción, cumpliendo las instrucciones de los técnicos superiores.
- Elaborar proyectos de la carrera.
- Demostrar calidad en su trabajo.
- Aplicar sistemas de mantenimiento preventivo y correctivo en equipo, maquinaria y herramienta, propias de la carrera, cuando corresponda.
- Organizar el espacio de trabajo, aplicando normas técnicas propias de la carrera.
- Utilizar los materiales, equipos, maquinarias y herramientas propios de su área de formación técnica, conforme los protocolos y especificaciones técnicas establecidas.

Competencias para el Desarrollo Humano

Se definen como competencias no específicas de una ocupación, necesarias para el desarrollo integral de una persona, un profesional o un ciudadano. Se adquieren durante el desarrollo del proceso de mediación pedagógica, en el desempeño del campo disciplinar y a lo largo de la vida.

- Desempeña las labores propias de su área de formación técnica con:
 - *autocontrol*: capacidad de control o dominio sobre uno mismo.



- *compromiso ético*: capacidad o voluntad para hacer el bien a través de relaciones morales entre humanos.
 - *discernimiento*: capacidad de comprender o declarar la diferencia entre varias cosas de un mismo asunto, involucra juicios morales o de actuación, resueltos con conciencia, aplicando un proceso lento de concentración para la toma de decisiones con ética y moral.
 - *responsabilidad*: capacidad de analizar procesos e identificar y comprender el asunto para proponer un planteamiento eficaz y viable.
-
- Propone soluciones a los problemas que se presentan en el campo laboral mostrando capacidad para el análisis de procesos e identificación y comprensión de planteamientos eficaces y viables.
 - Aplica los principios de atención al cliente.
 - Demuestra capacidad para ser atento con otro aplicando las políticas de la empresa, relacionándose de manera efectiva con el fin de resolver la necesidad, el servicio o producto planteado.
 - Atiende al usuario con proactividad y asertividad.
 - Se comunica correctamente tanto en forma oral como escrita. Demuestra capacidad de producir un canal de comunicación audible o visual para transmitir información en forma precisa
 - Demuestra capacidad para aprender por él mismo, sin necesidad de un mediador (autoaprendizaje).



- Se comunica asertivamente. Comunica información clara y objetiva en relación con puntos de vista, deseos y sentimientos, con honestidad y respecto a las otras personas.
- Trabaja en equipo de manera responsable y ordenada.
- Muestra capacidad de negociación. Expone puntos de vista con el propósito de obtener un acuerdo o resultados.
- Evidencia innovación y creatividad. Desarrolla productos o procesos de manera novedosa y creativa.
- Demuestra liderazgo en el desempeño de su área de formación técnica para el logro de las metas y objetivos de la organización y el bien común.
- Manifiesta capacidad para anticiparse a problemas o necesidades futuras, por iniciativa propia, en el ámbito de su área de formación técnica.
- Evidencia pensamiento crítico. Interpreta las opiniones o afirmaciones con argumentos válidos o veraces, aplicados al contexto de la vida cotidiana.
- Ética y bioética. Capacidad de las personas y las comunidades para enfrentar, analizar y resolver problemas morales y éticos relacionados con la vida humana, la salud y la biotecnología.
- Otras que el sector productivo y educativo requieran.



Docente

Constituye un facilitador de la información y el conocimiento. Para ello requiere de una verdadera disposición y compromiso para ser un promotor efectivo del desarrollo de las competencias. A continuación, algunas de las características del docente en un enfoque por competencias:

- Muestra inquietud por investigar, conocer y desarrollar conocimientos nuevos relacionados con su carrera técnica.
- Muestra conocimiento de la realidad nacional e internacional que se relaciona con el campo de acción de su carrera.
- Evalúa detenidamente su propio aprendizaje y experiencias.
- Reconoce sus capacidades y limitaciones, en busca de un continuo desarrollo personal.
- Domina y estructura los saberes para facilitar experiencias de aprendizaje significativo.
- Reconoce con profundidad las competencias, los contenidos y los enfoques que se establecen para la enseñanza, así como las interrelaciones y la racionalidad del plan de estudios.
- Posee competencias de pensamiento crítico, sistémico, divergente y reflexivo enmarcado en procesos éticos válidos ante la sociedad.



- Participa responsablemente en el proceso de desarrollo de competencias.
- Posee la habilidad de aprender a aprender.
- Promueve estrategias que motiven al estudiante a adquirir un aprendizaje significativo.
- Diseña, organiza y propone estrategias y actividades didácticas, adecuadas a los niveles y formas de desarrollo de competencias, que deben ser adquiridas por el o la estudiante, interrelacionando las características propias del medio social y cultural.
- Participa en el mejoramiento de la calidad educativa.
- Posee capacidad de expresarse en forma clara, sencilla y correcta en forma verbal y escrita, tanto en el ámbito técnico, como en el social cotidiano.
- Sabe escuchar los diferentes puntos de vista y atender las necesidades de expresión de los aprendientes e iguales en un marco de reflexión positiva.
- Aborda correctamente los procesos de solución de conflictos entre pares, promoviendo el diálogo, comprometiéndose con los ideales de la educación costarricense.
- Guía del desarrollo intelectual de las personas estudiantes.
- Genera estrategias de evaluación que motiven el aprendizaje significativo.



- Explora conocimientos y potenciales del alumno para el desarrollo de competencias.
- Trabaja en equipo.
- Expone empatía, sensibilidad y respeto por las necesidades y sentimientos de los demás.
- Posee sentido de equidad social, justicia, respeto, imparcialidad, integridad y honradez.
- Plantea, analiza y resuelve problemas; enfrentando desafíos intelectuales en los que genera respuestas propias a partir de sus conocimientos y experiencias.
- Posee capacidad de orientar a sus estudiantes para que estos adquieran la competencia de analizar y de resolver problemas.
- Identifica estilos de aprendizaje para optimizar y estimular las competencias.
- Determina su propio estilo en cuanto al proceso enseñanza aprendizaje usando múltiples fuentes de información e innovación.



Diseño Curricular

Dentro de los elementos del diseño curricular, el programa de estudio considera el desarrollo de las competencias específicas o técnicas propias del área de formación técnica, además de las competencias para el desarrollo humano y el eje de la política educativa “Persona centro del proceso educativo y sujeto transformador de la sociedad”, la cual permea todo el proceso educativo de la carrera técnica o carrera seleccionada por el o la estudiante.

Los resultados de aprendizaje son enunciados asociados con lo que se espera que la persona estudiante sea capaz de hacer, comprender o demostrar, una vez terminado el proceso de aprendizaje. Los saberes esenciales son el conjunto de conocimientos técnicos, teóricos, metodológicos del campo disciplinar y de otras disciplinas requeridas para el proceso de aprendizaje en su área de formación técnica y para la vida. Estos deben desarrollarse para el logro de los resultados de aprendizaje determinados en la propuesta curricular.

Los indicadores de logro constituyen enunciados que expresan el camino hacia el cumplimiento del estándar, reflejan los propósitos, metas y aspiraciones a alcanzar por la persona estudiante, desde el punto de vista afectivo, cognitivo e instrumental. Son indicadores para la macro evaluación que permiten visualizar y evidenciar el nivel de logro alcanzado por el estudiantado como producto del abordaje pedagógico desarrollado por el o la docente.



A continuación, el formato establecido en el diseño curricular de este programa de estudio.

Tabla 2

Información administrativa

Carrera técnica¹:	Campo detallado²:
Subárea:	Nivel:
Unidad de estudio:	Tiempo estimado:
Competencia para el desarrollo humano:	Eje política educativa³:

¹ Nombre de la cualificación del estándar aprobado por el MNC-EFTP-CR.

² Según la Clasificación Internacional Normalizada de Educación (CINE).

³ Política educativa “Persona centro del proceso educativo y sujeto transformador de la sociedad”.



Tabla 3

Planificación Curricular de la Unidad de Estudio

Resultados de aprendizaje	Saberes esenciales	Indicador de logro ⁴
1.		
2.		
3.		
4.		

⁴ Indicadores para la macroevaluación.



Principios Didácticos y Estrategias Metodológicas para la Mediación Pedagógica

La educación ocupa un lugar central en la agenda de los países y esto se debe a razones como los rápidos avances en las tecnologías de la información y la comunicación, el cambio hacia economías basadas en el conocimiento y el énfasis en las habilidades críticas y capacidades requeridas al ciudadano del siglo XXI. Bajo esta premisa, el sistema educativo y la persona docente en particular deben facilitar una mediación pedagógica que permita la adquisición de conocimientos, el desarrollo de competencias y las herramientas que requiere una persona para su desempeño en la sociedad actual.

Las nuevas generaciones están influidas de modo directo e indirecto por las tecnologías de la información y las telecomunicaciones, lo que hace, entre otros factores, que aprendan en modo distinto a las generaciones precedentes. No basta con emplear recursos tecnológicos para satisfacer necesidades de aprendizaje y formación. El reto está en que las nuevas tecnologías constituyan un medio para formar a las nuevas generaciones de ciudadanos con los valores que demanda la sociedad.

Por esta razón, el método de aprendizaje constituye un factor clave en la creación de nuevos ambientes de aprendizaje; es la vía o camino para la presentación de la información, los pasos que se siguen y hacen que las personas estudiantes participen de modo activo e interactivo, crítico, reflexivo, creativo, comprometido y responsable. El estudiantado no es solo receptor de la información sistematizada y presentada por otros, sino todo lo contrario, participa en la construcción del conocimiento y contribuye al aprendizaje de los demás miembros de su grupo.



Dentro de este orden de ideas, John Biggs propone el alineamiento constructivo, el cual constituye un modelo pedagógico que responde a la pregunta cómo enseñar para que todos los miembros de la clase aprendan más profundamente y cómo revitalizar el sentido de enseñar más allá de transmitir contenidos. Su modelo conceptual propone una manera diferente de delimitar y expresar qué se enseña, cómo se enseña y qué se evalúa.

Biggs señala que la enseñanza “forma un sistema complejo, el cual incluye a nivel del aula al profesor, los estudiantes, el contexto, las actividades de aprendizaje y sus resultados” (Biggs, 1996, p. 350). Estos elementos necesitan estar alineados si queremos fomentar el aprendizaje de los estudiantes: “cuando hay alineamiento entre lo que queremos, cómo enseñamos y cómo evaluamos, es probable que la enseñanza sea mucho más eficaz que cuando no lo hay” (Biggs, 2004, p.46).

Para Espejo y Sarmiento (2017), “Este alineamiento tiene lugar en un contexto, o bajo ciertos factores situacionales que no podemos olvidar al diseñar un curso” (p. 18). Esto significa que el profesorado debe partir de los resultados de aprendizaje del curso que dicta y posteriormente, diseñar un sistema de evaluación y actividades de enseñanza-aprendizaje que sean: a) coherentes entre sí, y b) coherentes con los resultados de aprendizaje antes descritos. Esto implica que en realidad la evaluación no debe tratarse como algo aislado de las metodologías de enseñanza aprendizaje, sino como parte integrante.

Según lo expuesto en el Diagrama 1, el alineamiento constructivo requiere que las personas docentes conozcan, con claridad y precisión, los elementos centrales del planeamiento educativo.



Diagrama 2

Interconexión entre los tres elementos centrales del planeamiento curricular



Los resultados de aprendizaje esperados (RAEs) o competencias (antes llamados objetivos o metas: ¿qué esperamos que las personas estudiantes logren en sus carreras, cursos o clases?

- Las actividades de enseñanza y aprendizaje (AEAs): ¿qué van a hacer nuestros estudiantes para alcanzar los resultados esperados y qué vamos a hacer nosotros para apoyarlos?



- Los medios de evaluación: ¿cómo vamos a evaluar si nuestros estudiantes alcanzaron los resultados esperados?

En concordancia con el modelo del alineamiento constructivo, un abordaje metodológico orientado a la implementación de la mediación pedagógica es requerido para la EFTP. Este modelo se caracteriza por alejarse de los procedimientos sistemáticos – relacionados con estructuras teóricas específicas – y en una didáctica que facilite la conexión entre el conocimiento y la acción.

Los métodos orientados a la acción emplean estrategias didácticas que vinculen a la persona estudiante con situaciones de la vida y el trabajo. En este contexto, la didáctica orientada a la acción considera la resolución de problemas e incluye la planificación, la ejecución, el control y la evaluación. Por esta razón, no basta con llevar a cabo acciones según las instrucciones, debido a que el propósito central de este enfoque pedagógico es el desarrollo de la competencia de acción.

Estos métodos incluyen el aprendizaje relacionado con el contenido, el aprendizaje metódico para la resolución de problemas, el aprendizaje social-comunicativo y el aprendizaje afectivo-ético. Algunas estrategias orientadas a la acción que la persona docente puede implementar en su mediación pedagógica son: proyectos, situaciones simuladas, juegos empresariales, estudios de caso, juegos de rol, entre otros.

En este sentido, los métodos se basan en el desarrollo de actividades complejas requeridas para la vida y el mundo del trabajo y que el estudiantado ejecuta de forma independiente. Algunos ejemplos de métodos orientados a la acción son las simulaciones, los juegos de empresa, los estudios de casos, los juegos de rol y el



método del texto guía. Este último permite estimular y estructurar los procesos de aprendizaje; comprende preguntas orientadoras, principios rectores, planes de trabajo y fichas de control.

Otra recomendación son los talleres de escenarios y futuro, considerados como sesiones de trabajo colaborativo diseñadas para explorar posibles futuros, identificar tendencias emergentes y desarrollar estrategias para adaptarse a ellos. Estos talleres suelen utilizar métodos de pensamiento crítico y creativo para ayudar a los participantes a visualizar y comprender diferentes escenarios futuros, desde los más probables hasta los más improbables. (OpenAI, 2024)

Es importante señalar que la incorporación de métodos de una didáctica orientada a la acción, el desarrollo de una mediación pedagógica con metodologías activas, la planificación y el diseño de situaciones de aprendizaje auténticas promueven un aprendizaje basado en actividades realistas y brindan información clara de los conocimientos y capacidades desarrolladas por las personas estudiantes. Por otra parte, propician la motivación, ya que el estudiantado se compromete en actividades de importancia en entornos empresariales y que le facilitan la aplicación de su aprendizaje en la solución a problemas del mundo real o entorno laboral específico.

Las metodologías activas se centran en el estudiantado y se caracterizan por concebir el aprendizaje como proceso y no únicamente como una recepción y acumulación de información. Otro elemento que fundamenta su aplicación es el aprendizaje autodirigido, es decir el desarrollo de habilidades metacognitivas



que promueven un mejor y mayor aprendizaje, promueven el trabajo en equipo, la discusión, la argumentación y la evaluación constante de lo que aprenden.

Estas metodologías enfatizan que la enseñanza debe tener lugar en el contexto de problemas del mundo real o de la práctica profesional. Se deben presentar situaciones lo más cercanas posibles al contexto profesional en que la persona estudiante se desarrollará en el futuro. La contextualización de la enseñanza promueve la actitud positiva hacia el aprendizaje y motivación; además, le permite al estudiante enfrentarse a problemas reales, con un nivel de dificultad y complejidad similares a los que encontrará en la práctica profesional.

El Compendio de estrategias para la mediación pedagógica de la ETP, elaborado por la Unidad de Planificación y Diseño Curricular del Departamento de Especialidades Técnicas, abarca metodologías activas que la persona docente y mentora pueden implementar; entre ellas:

- **Aula invertida:** concebida como un modelo pedagógico que plantea la necesidad de transferir parte del proceso de enseñanza y aprendizaje fuera del aula, con el fin de utilizar el tiempo de clase para el desarrollo de procesos cognitivos de mayor complejidad que favorezcan el aprendizaje significativo.
- **Aprendizaje reflexivo basado en la indagación:** similar al aprendizaje basado en proyectos; sin embargo, el rol del profesorado es diferente. En el aprendizaje reflexivo o basado en la indagación, la persona estudiante explora un tópico y elige el tema, desarrolla el plan de investigación y llega a conclusiones, aunque la persona docente esté disponible para proporcionar ayuda y orientación cuando sea necesario.



- **Aprendizaje basado en problemas:** si bien esta estrategia se inicia con la formulación del problema planteado por el estudiantado o la persona docente, su propósito no solo se centra en la resolución del problema, sino en el proceso de fundamentar la posible solución. Esto se aprecia cuando se asigna el mismo problema a varios grupos. Al presentar las soluciones se observa cuál estrategia o argumentación se adoptó en cada uno de los equipos.
- **Aprendizaje basado en proyectos:** se define el proyecto como el conjunto de actividades articuladas entre sí, con el fin de generar productos, servicios o comprensiones capaces de resolver problemas o satisfacer necesidades e inquietudes, según los recursos y el tiempo asignado. Es una estrategia metodológica de diseño y programación que implementa un conjunto de tareas basadas en la resolución de preguntas o problemas (retos), mediante un proceso de investigación o creación por parte del estudiantado que trabaja de manera relativamente autónoma, con un alto nivel de implicación y cooperación y que culmina con un producto final presentado ante los demás.
- **Aprendizaje basado en retos:** tiene sus raíces en el aprendizaje vivencial y tiene como principio fundamental que los y las estudiantes aprendan mejor cuando participan de forma activa en experiencias abiertas de aprendizaje, que cuando actúan de manera pasiva en actividades estructuradas.
- **Taller:** constituye una metodología que integra la teoría y la práctica. Se caracteriza por la investigación, el aprendizaje por descubrimiento y el trabajo en equipo que requiere del acopio y sistematización de material especializado, acorde con el tema tratado y cuyo fin es la elaboración de un producto tangible. Enfoca sus acciones hacia el saber hacer, es decir, hacia la práctica de una actividad. La persona docente ya no enseña en el sentido tradicional, sino que es un asistente técnico que ayuda a aprender y



el estudiantado aprende haciendo. Puede organizarse con el trabajo individualizado del estudiante, en parejas o en pequeños grupos, siempre y cuando el trabajo que se realice trascienda el simple conocimiento, convirtiéndose de esta manera en un aprendizaje integral que implica la práctica.

- **Proyecto:** enfrenta al estudiantado a situaciones que los llevan a comprender y aplicar lo que aprenden, como una herramienta para resolver problemas. Estas experiencias en las que se ven involucrados hacen que aprendan a manejar y usar los recursos disponibles como el tiempo y los materiales; además, desarrollan y perfeccionan habilidades académicas y sociales a través de la mediación pedagógica. La técnica de proyectos se aboca a conceptos fundamentales y principios de la disciplina del conocimiento y no a temas selectos. La situación en que trabaja el estudiantado es, en lo posible, orientada a la vida real y al contexto laboral, frecuentemente con dificultades reales por enfrentar y con una realimentación constante.
- **Aprendizaje cooperativo:** reviste de importancia como metodología para el desarrollo de estrategias de mediación pedagógica bajo el enfoque por competencias. Es la interdependencia que se logra a partir de las relaciones de cooperación entre los implicados en un aprendizaje. Ello no implica suprimir el trabajo individual, es necesario prepararse mejor para el esfuerzo grupal, con el objeto de alcanzar entre todos la tarea. Cooperar es compartir una experiencia vital significativa que exige trabajar juntos para lograr beneficios mutuos. La cooperación implica resultados en conjunto, mediante la interdependencia positiva que involucra a todos los miembros del equipo en lo que se hace, y en cuyo proceso cada uno aporta su talento (Ferreiro, 2007).



- **Aprendizaje basado en la experiencia:** la necesidad de adquirir competencias acordes con la exigencia competitiva de las empresas y las condiciones cambiantes del contexto es una realidad actual en nuestra sociedad. Es necesario promover habilidades relacionadas con la resolución de problemas, el aprendizaje autónomo, la capacidad para tomar decisiones, autodirigir las acciones y analizar su impacto. Para alcanzar las competencias anteriormente citadas, el aprendizaje experiencial es una herramienta muy útil en la formación del trabajo, ya que le permite al estudiante adquirir conocimiento con eficacia y en corto tiempo.
Este enfoque educativo se basa en el aprendizaje activo y la aplicación práctica del conocimiento. A diferencia de los de orientación más tradicional y centrados en la transmisión de información de manera pasiva, las personas estudiantes aprenden mejor cuando se involucran en experiencias prácticas y significativas que demandan su participación, conexión con el mundo real y aprendizaje reflexivo. En el aprendizaje basado en la experiencia, las personas (individualmente o en grupo) realizan determinadas acciones y observan los efectos, construyen el conocimiento de forma profunda y aumentan la comprensión, la eficacia y eficiencia al aplicar las competencias aprendidas.
- **Simulación:** son experiencias de aprendizaje enfocadas en el reto, desafío y aventura, presentando de manera simplificada y resumida modelos de situaciones reales y complejas que someten al estudiantado a la toma de decisiones, liderazgo, comunicación, planificación y delegación. La simulación es una técnica muy útil para lograr un aprendizaje significativo y recrear experiencias que serían imposibles de vivenciar en la realidad, tal como ocurre por ejemplo con los hechos del pasado. El estudiantado puede representar situaciones a las que se enfrenta en el trabajo o que esperan encontrar en el futuro. Se les



puede encomendar la tarea de gestionar una empresa, a partir de una situación dada, o la gestión de una función específica dentro de un ambiente simulado.

Las simulaciones basadas en la realidad facilitan el cambio de actitudes y habilidades, con el objetivo de que ese cambio tenga un impacto directo en el desempeño laboral. Produce un alto grado de motivación y la participación del estudiante. Desarrolla habilidades y destrezas, estimula el espíritu crítico, permite visualizar las consecuencias de su accionar y aplica en forma práctica los conocimientos teóricos adquiridos.

Las simulaciones son una herramienta altamente efectiva para implementar el aprendizaje experiencial. Ofrecen a las personas estudiantes la oportunidad de participar activamente, practicar habilidades y aplicar conocimientos en situaciones reales o simuladas. En definitiva, son de beneficio para el aprendizaje presencial y el aprendizaje en línea significativo y duradero.

- **Demostración:** técnica empleada para enseñar y evaluar habilidades, herramientas y aprendizajes específicos. Implica que el estudiantado exponga, explique o aplique ante la persona docente y una audiencia particular, el procedimiento, el proceso de un tema o el tópico bajo estudio, en forma concreta. Es decir, mediante una demostración la persona estudiante realiza una ejecución real o simulada ante otros. La demostración permite valorar la apropiación, comprensión o capacidad para aplicar una teoría, método, técnica o algún instrumento; además, apreciar la definición propia de conceptos, actitudes y habilidades relacionadas con la resolución de problemas, el pensamiento crítico y la comunicación efectiva, lo que constituye un monitor de su propio aprendizaje y fomenta la metacognición.



La ETP promueve la utilización de metodologías activas y la exposición de la persona estudiante a entornos de aprendizaje reales, propios de la práctica profesional, lo cual le brinda una visión más compleja de este espacio. De acuerdo con el modelo pedagógico, brinda la oportunidad de desarrollar tareas auténticas vinculadas de modo significativo al entorno.

En este contexto, el rol de la persona docente es proveer entornos de aprendizaje que propicien el desarrollo de capacidades y fomenten la reflexión en torno a la experiencia, la negociación social (aprendizajes cooperativos), sin dejar de tomar en consideración las características propias del estudiantado. El aprendizaje debe entenderse como la reconstrucción de saberes culturales, partiendo de los conocimientos previos y permitiendo su reorganización interna.

Con la finalidad de facilitar la mediación pedagógica que realizan las personas docentes, se presentan algunas orientaciones didácticas y pedagógicas para la aplicación de currículos basados en enfoque por competencias:

- Articulación de resultados de aprendizaje, saberes esenciales, actividades y sistema de evaluación como línea de trabajo por seguir.
- Aplicación de métodos variados que resulten apropiados para la adquisición de aprendizajes de diferente naturaleza: conceptos, teorías, habilidades, actitudes y valores. La diversidad de métodos permite acceder desde varias perspectivas al objeto de aprendizaje, de manera que se pueda apprehender de



forma integral. Sin embargo, no se debe dispersar la atención del estudiante con una diversidad de metodologías cambiantes.

- Inclusión de metodologías variadas dentro de un marco coherente y que responda a las características antes mencionadas. Ninguna estrategia docente es la solución única, sino más bien una excusa para invitar a las personas estudiantes a actuar y, sobre la base de sus producciones, crear oportunidades de intercambio y reflexión.
- Selección de actividades de contexto que la persona estudiante puede reconocer como socialmente valoradas y un medio para estimular su interés y motivación.
- Un entorno que facilite un aprendizaje de calidad caracterizado, entre otros elementos, por coordinar los resultados de aprendizaje y el método docente con las estrategias, técnicas y actividades de evaluación (metodología de evaluación), de modo que todo el proceso de mediación pedagógica sea coherente y los actores de dicho proceso (docentes y estudiantes) sean copartícipes del mismo.
- Implementación de las tecnologías de Información y comunicación para crear entornos virtuales y simular condiciones laborales reales (CSUCA, 2018, p. 86).

El papel de la persona docente – como actor clave de la ETP – es fundamental para el alcance de aprendizajes significativos. En su rol en el proceso educativo, se espera que:



- Sea experto en su campo profesional y especialista en el diseño de procesos de enseñanza-aprendizaje que respondan individualmente a una gran variedad de necesidades.
- Sea un actor relevante en la preparación de jóvenes y adultos para el mercado laboral, mediante la enseñanza no solo de competencias profesionales, sino también de las transversales, genéricas y para el desarrollo humano.
- Apoye la transición de la “escuela al mundo del trabajo” de las personas estudiantes con diversos antecedentes, incluidos los que tienen dificultades con los estudios académicos y los adultos que necesitan adquirir nuevas competencias, actualizarlas o mejorarlas.
- Prepare al estudiantado para el mundo laboral combinando sus diferentes conocimientos.
- Promueva el aprendizaje permanente, la formación integral y el desarrollo individual.
- Evalúe y reconozca individualmente las necesidades, experiencias y exigencias de sus estudiantes, integrándolas en la mediación pedagógica.
- Facilite la adaptación a las exigencias y al mundo del trabajo en constante cambio, en aspectos como la digitalización, automatización, procesos en la empresa, heterogeneidad, entre otros,
- Sea mediador entre el mercado laboral y la cualificación profesional (OCDE, 2021).



Orientaciones para la Realización de Actividades Pedagógicas Fuera de la Institución

El documento Orientaciones y lineamientos para el desarrollo de actividades pedagógicas fuera del centro educativo en la ETP tiene como finalidad orientar y dar a conocer los requisitos para realizar visitas, giras, pasantías y práctica profesional en las asignaturas del área técnica del plan de estudios de la ETP que se imparten en los colegios técnicos profesionales, IPEC y CINDEAS que ofertan carreras técnicas.

Las actividades pedagógicas fuera del centro educativo constituyen el medio idóneo para fortalecer y desarrollar conocimientos, habilidades, destrezas y actitudes en las personas estudiantes, a través de la relación con el entorno y una realidad concreta.

Para la implementación de estas actividades, todos los actores deben cumplir con lo que establece el documento citado. Sus disposiciones son de acatamiento obligatorio y de aplicación inmediata en los colegios técnicos profesionales e instituciones públicas que imparten carreras de la ETP. Asimismo, toda actividad pedagógica fuera de la institución educativa debe corresponder únicamente con el desarrollo o complemento de los planes y programas de estudio y cumplir con las disposiciones ministeriales y legislación vigente.

Orientaciones y lineamientos para el desarrollo de actividades pedagógicas fuera del centro educativo en la ETP establece las actividades pedagógicas contempladas en los programas de estudios vigentes y el proceso de aprendizaje del estudiante de la ETP:



Práctica Profesional

Es una actividad de índole curricular que realizan las personas estudiantes en forma individual, cuando cursan el último nivel en los colegios técnicos profesionales, colegios técnicos profesionales nocturnos, secciones técnicas nocturnas de colegios técnicos profesionales e IPEC y CINDEA que imparten carreras técnicas.

La práctica profesional está directamente relacionada con la carrera técnica cursada. Su objetivo es aplicar y complementar los conocimientos adquiridos por la persona estudiante durante su formación técnica, favorecer la adquisición de competencias para el ejercicio de actividades profesionales, facilitar su empleabilidad y fomentar su capacidad de emprendimiento.

Esta actividad se rige por lo establecido en el Reglamento de Requisitos de Graduación para optar por el Título de Técnico en el Nivel Medio en las carreras aprobadas por la DETCE. Se puede realizar en empresas, instituciones y entidades públicas o privadas, en el ámbito nacional o internacional.

Pasantía

Actividad de índole curricular y de carácter obligatorio, que forma parte del proceso de enseñanza y aprendizaje que se realiza en organizaciones públicas o privadas. Su objetivo es lograr que la persona estudiante vivencie la realidad inherente a su carrera y facilite, de esta manera, su incorporación al sector productivo.



Gira

Actividad pedagógica que constituye un medio alternativo y vivencial de aprendizajes significativos, un espacio de formación constante para la persona estudiante, a partir de diversas vivencias en contextos particulares y guiados por la persona docente.

Visita

La visita es un recorrido con fines de aprendizaje que el estudiantado de la ETP realiza de forma individual o grupal, bajo la orientación y acompañamiento del docente, guías especiales o ambos, a un lugar seleccionado previamente como museo, zona histórica o arqueológica, galería, parque, reserva, oficina pública, empresa, laboratorio, fábrica, taller, comunidad, montaña, entre otros. Lo anterior de conformidad con la naturaleza de la carrera técnica que cursa la persona estudiante y lo establecido en el respectivo programa de estudio (MEP, 2021, p 8-16).



Planeamiento del Proceso de Aprendizaje

Plan Anual

El plan anual se realiza a partir del programa de estudio vigente y constituye el cronograma en el que se representan las unidades de estudio – con sus respectivos resultados de aprendizaje – en los meses y semanas que componen el curso lectivo.

La persona docente debe elaborar un plan anual por cada subárea. Para tal efecto, indica las semanas y horas destinadas al desarrollo de cada una de las unidades de estudio y resultados de aprendizaje que componen el programa de estudio de la subárea. Adicionalmente, debe respetar la secuencia lógica indicada en el programa para el abordaje del proceso educativo.

Para elaborar el plan anual, el o la docente consideran la información contenida en la estructura, mapa y malla curricular del programa de estudio de la subárea(s) a su cargo.

El plan anual se entrega a la persona directora del centro educativo, de manera física o digital, según lo establezca la administración al inicio del curso lectivo. A continuación, el formato del plan anual aprobado por el CSE:



Ilustración 1

Tabla para la Elaboración del Plan Anual

Plan Anual

Centro educativo:																																																				
Carrera técnica:																																																				
Subárea:																																									Nivel:											
Docente:																																									Curso lectivo:											
Unidades de estudio y resultados de aprendizaje	Feb				Mar				Abr				May				Jun				Jul				Ago				Set				Oct				Nov				Dic				Tiempo (horas)							
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4								
Recursos educativos:																																																				



Plan de Práctica Pedagógica

El plan de práctica pedagógica se prepara de forma mensual. Es un documento de uso diario y se entrega al director o directora, de manera física o digital, cuando la administración del centro educativo lo juzgue oportuno, de manera que se pueda comprobar que su desarrollo es congruente con lo planificado en el plan anual preparado por la persona docente al inicio del curso lectivo.

Su formato contempla dos secciones: administrativa y técnica. En la primera parte la persona docente incluye el nombre del centro educativo, su nombre y apellidos, el nivel, la carrera técnica que imparte, modalidad (agropecuario, comercial y servicios e industrial), el campo detallado, la subárea, la unidad de estudio, el tiempo estimado, la competencia para el desarrollo humano y el eje de la Política Educativa.

Cabe mencionar que, el campo detallado se indica según la Clasificación Internacional Normalizada de la Educación (CINE). En el caso de la subárea, la unidad de estudio y el tiempo estimado, deben tener concordancia con lo establecido en el plan anual, así como en la estructura, mapa y malla curricular del programa de estudio.

La competencia para el desarrollo humano y los ejes de la política educativa se desarrollan a lo largo de todo el programa de estudio y son elementos que forman parte del desarrollo de la sección técnica del plan de práctica pedagógica.



La persona docente debe trasladar los resultados de aprendizaje y saberes esenciales del programa de estudio, según la subárea y unidad de estudio correspondiente. La experiencia del docente determina el tipo de estrategia y técnica pedagógica que empleará para la mediación. En este sentido, se contemplan la que utilizará como docente para su abordaje en el aula y la que ejecutará la persona estudiante.

La persona docente se encarga de generar los indicadores de logro que espera observar en el estudiantado, como producto de las estrategias de mediación empleadas y las evidencias de conocimiento, desempeño o producto, según corresponda. Los indicadores de logro, establecidos en el plan de práctica pedagógica, deben tener concordancia con la información incluida en los instrumentos técnicamente elaborados para el proceso de evaluación y, en el caso de las evidencias, deben observarse en el portafolio de evidencias del estudiante.

Con respecto al tiempo estimado, la persona docente lo determina en horas y se refiere al periodo requerido para el abordaje de cada uno de los resultados de aprendizaje, respetando lo establecido en el plan anual.

El eje de la política educativa corresponde a la política curricular “Educar para una nueva ciudadanía”. Según la Ilustración 1, en la parte inferior del plan anual la persona docente indica los recursos de espacio físico, materiales, equipo y herramientas que utilizará para el desarrollo del plan de práctica pedagógica.

A continuación, se detalla el formato del plan de práctica pedagógica, según lo aprobado por el CSE en el programa de estudio.



MINISTERIO DE
EDUCACIÓN PÚBLICA

GOBIERNO
DE COSTA RICA

Dirección de Educación
Técnica y Capacidades
Emprendedoras

Plan de Práctica Pedagógica

Centro educativo:

Nombre del docente:

Nivel:

Carrera técnica:

Modalidad:

Campo detallado⁵:

Subárea:

Unidad de estudio:

Tiempo estimado:

Competencias para el desarrollo humano:

Eje Política Educativa⁶:

⁵ Según la Clasificación Internacional Normalizada de la Educación (CINE).

⁶ Política Curricular “Educar para la nueva ciudadanía”.



Tabla 4

Formato del Plan de Práctica Pedagógica

Resultados de aprendizaje	Saberes esenciales	Estrategias para la mediación pedagógica	Evidencias	Tiempo estimado (horas)
1.		Docente Estudiante	Conocimiento Desempeño Producto	
2.		Docente Estudiante	Conocimiento Desempeño Producto	



Evaluación del Proceso de Aprendizaje

Hablar de evaluación por competencias significa incorporar nuevas estrategias e implementar una evaluación orientada al aprendizaje, centrada en la participación del estudiante, dirigida a situaciones de naturaleza auténtica cada vez más cercanas a la vida real. Por lo anterior, la competencia es contextual, refleja la relación entre las habilidades de las personas y las actividades que desempeñan en una situación particular en el mundo real (López, 2014).

La evaluación en un enfoque por competencias es continua, dinámica, holista y dirigida al análisis de los niveles de desempeño alcanzados por la persona estudiante. Es decir, cumple una función de autorregulación que le permite al estudiante generar un monitoreo personal de su aprendizaje.

Desde esta perspectiva, la competencia predice el desempeño, está directamente vinculada con procesos prácticos del estudiante y no tanto con el cúmulo de datos. Mediante la evaluación se identifican y registran los atributos de la competencia que se pretende desarrollar a través de los procesos y las evidencias generadas por el estudiantado, con la intención de valorar la evolución del dominio y su respectiva transferencia. El o la docente deben plantear juicios basados en el proceso y las evidencias de sus estudiantes, por medio de la observación y el análisis de la evolución del dominio de niveles.

La evaluación debe estar alineada al currículum y acorde con los resultados de aprendizaje, las estrategias de mediación por desarrollar durante el proceso educativo y el sistema de valoración de los conocimientos, desempeños y productos deseados, según los indicadores de logro establecidos. Es importante señalar



también que ofrece estrategias que posibilitan conocer a profundidad los resultados obtenidos por las personas estudiantes.

Mediante la evaluación basada en competencias, las personas estudiantes ofrecen al docente, padres de familia, compañeros (as) y comunidad en general, las “evidencias” de su desempeño con nuevas herramientas y métodos de evaluación, las cuales se apoyan en una perspectiva de corte constructivista y centran su dinámica en los procesos.

Una vez seleccionadas las estrategias de mediación pedagógica, se definen los instrumentos de evaluación. En ellos se incluyen los indicadores de logro y los criterios de desempeño para valorar la situación de aprendizaje, pues permiten al docente emitir juicios sobre lo alcanzado por cada estudiante.

Para ser objetivo en la emisión de juicios de valor, es importante establecer los indicadores de logro y las evidencias asociadas a los niveles de valoración establecidos, con el propósito de que al finalizar se proceda con el análisis de la información recolectada y se determine si se han alcanzado las competencias y en cuáles niveles. Lo anterior permite la toma de decisiones respecto al desarrollo de las competencias por parte de cada estudiante.

El Reglamento de Evaluación de los Aprendizajes (REA), mediante decreto ejecutivo, rige la evaluación costarricense y establece los componentes para cada una de las modalidades del sistema educativo. De esta manera, para obtener el promedio (por periodo) de cada asignatura o subárea que cursa la persona estudiante, se suman los valores porcentuales de cada componente de la calificación.



En el caso de los talleres exploratorios y subáreas correspondientes a la ETP, tanto en modalidades diurnas, nocturnas y plan a dos años, el REA establece y asigna un valor porcentual a los siguientes componentes de la calificación:

Trabajo Cotidiano

Se refiere a las actividades educativas que realiza el estudiantado, con la guía y orientación de la persona docente, según el planeamiento didáctico y el programa de estudios. Para su calificación, se deben utilizar instrumentos técnicamente elaborados, en los que se registre información relacionada con el desempeño del estudiante.

La información para calificar el trabajo cotidiano se recopila durante el transcurso del período y el desarrollo de las lecciones, como parte del proceso de enseñanza - aprendizaje y no como producto. Asimismo, debe reflejar el avance gradual del estudiante en sus aprendizajes.

En el caso de las asignaturas de las carreras técnicas del Plan de Estudios de Educación de Adultos y la Educación Diversificada Técnica, el trabajo cotidiano incluye la realización del portafolio de evidencias.

Tareas

Se refiere a los trabajos cortos asignados al estudiantado, con el propósito de reforzar o repasar aprendizajes esperados, según la información recopilada durante el trabajo cotidiano. Por tal razón, es indispensable que



sean ejecutadas únicamente por la persona estudiante, de tal forma que pueda fortalecer su propio aprendizaje.

Las tareas no deben asignarse para ser desarrolladas en horario lectivo y en períodos de vacaciones, entiéndase Semana Santa y medio año, o período de pruebas calendarizadas en el centro educativo.

Pruebas

Son un instrumento de medición cuyo propósito es que el estudiantado demuestre la adquisición de habilidades cognitivas, psicomotoras o lingüísticas. Pueden ser escritas, de ejecución u orales. Para su construcción, se seleccionan los aprendizajes esperados e indicadores, de acuerdo con el programa de estudio vigente y del nivel correspondiente.

A menos que la persona docente lo juzgue necesario, las pruebas no deben tener carácter acumulativo durante un mismo período. La prueba escrita debe ser resuelta individualmente y aplicarse ante la presencia del docente o, en su defecto, en presencia del funcionario (a) que la persona directora designe. En lo que se refiere a la prueba oral y de ejecución, debe aplicarse ante él o la docente a cargo de la asignatura o subárea. Las pruebas cortas deben tener carácter formativo, salvo el caso de las aplicadas al estudiantado con necesidades educativas.



Proyecto

Consiste en un proceso de construcción de aprendizajes, guiado y orientado por la persona docente. Parte de la identificación de contextos del interés del estudiante y se relaciona con contenidos curriculares o resultados de aprendizaje, valores, actitudes, aprendizajes obtenidos y prácticas propuestas en cada unidad temática del programa de estudio o subárea de la carrera técnica.

El propósito del proyecto es que el estudiantado aplique lo aprendido en la realización reflexiva de un conjunto sistemático de acciones de interés, circunscrito en un contexto determinado del entorno sociocultural.

Se realiza de manera individual o grupal. Para su evaluación, la persona docente debe entregar al estudiantado los indicadores y criterios acordes con las etapas definidas y considerar tanto el proceso como el producto, así como evidenciar la autoevaluación y coevaluación.

Asistencia

La asistencia se define como la presencia de la persona estudiante en las lecciones y en todas aquellas otras actividades escolares a las que fuere convocado. Las ausencias y las llegadas tardías podrán ser justificadas o injustificadas (MEP, 2018, Art. 25-30).

Existe una gama de estrategias y herramientas que la persona docente puede utilizar como parte del proceso de evaluación de los componentes de evaluación citados. En el caso del trabajo cotidiano se cita el mapa



conceptual, portafolio de evidencias, línea de tiempo, mapa mental, mapas cognitivos, video foro, proyectos, collage, plenarias, entre muchas otras.

La persona docente debe confeccionar los instrumentos de evaluación técnicamente elaborados, que muestren los indicadores y permitan visualizar el nivel de logro del estudiante, según la normativa vigente y las directrices ministeriales emanadas.

Las pruebas escritas y de ejecución constituyen instrumentos de evaluación de gran importancia para la valoración del desempeño del estudiante. Deben confeccionarse de acuerdo con los lineamientos técnicos establecidos por el Departamento de Evaluación de los Aprendizajes del MEP.

El portafolio de evidencias es una herramienta valiosa, ya que permite observar las evidencias del proceso de aprendizaje de las personas estudiantes en el desarrollo de las competencias, según los lineamientos establecidos por la DETCE.

MICRO

Currículum

Carrera técnica:

**Desarrollo de
aplicaciones de
software**

COMPONENTES:

- Estructura curricular
- Mapa curricular
- Malla curricular
- Sílabos



Estructura Curricular

Tabla 5

Número de horas por subárea y nivel educativo

Subárea	Horas semanales 10° año	Horas anuales 10° año	Horas semanales 11° año	Horas anuales 11° año	Horas semanales 12° año	Horas anuales 12° año
Herramientas para la producción, transformación digital y seguridad de datos	4	160	-	-	-	-
Tecnologías de información	4	160	-	-	-	-
Infraestructura y operaciones de servicios de software	4	160	-	-	-	-
Emprendimiento e innovación aplicadas al desarrollo de aplicaciones de software	-	-	4	160	-	-
Bases de datos para proyectos de software	-	-	8	320	-	-
Desarrollo de componentes de software	8	320	8	320	12	300



Subárea	Horas semanales 10° año	Horas anuales 10° año	Horas semanales 11° año	Horas anuales 11° año	Horas semanales 12° año	Horas anuales 12° año
Experiencias de usuario y diseño de interacciones para componentes de software	-	-	-	-	8	200
English Oriented to Software Applications Development	4	160	4	160	4	100
Total 2840 horas⁷	24	960	24	960	24	600

⁷ Incluye las 320 horas de la práctica profesional de duodécimo nivel.



Mapa Curricular

Nivel: Décimo

Tabla 6

Unidades de estudio por subárea

Subárea	Unidad de estudio 1	Unidad de estudio 2	Unidad de estudio 3	Unidad de estudio 4	Unidad de estudio 5
Herramientas para la producción, transformación digital y seguridad de datos	Alfabetización digital para el bienestar y la seguridad en línea	Transformación digital, seguridad y análisis de la información	Herramientas para la producción de documentos	NA	NA
Tecnologías de información	Fundamentos de tecnologías de la información	Fundamentos de ciberseguridad	NA	NA	NA
Desarrollo de componentes de software	Herramientas lógicas y robótica	Algoritmos y diagramas de flujo	Software y su ingeniería	Levantamiento de requerimientos	Introducción a la programación



Subárea	Unidad de estudio 1	Unidad de estudio 2	Unidad de estudio 3	Unidad de estudio 4	Unidad de estudio 5
Infraestructura y operaciones de servicios de software	Virtualización y computación en la nube	Introducción a las redes	NA	NA	NA

NA: No aplica.



Nivel: Undécimo

Tabla 7

Unidades de estudio por subárea

Subárea	Unidad de estudio 1	Unidad de estudio 2	Unidad de estudio 3	Unidad de estudio 4
Emprendimiento e innovación aplicadas a la configuración y administración de servicios en la nube	Oportunidades de negocios	Modelo de negocios	Creación de la empresa	Plan de vida
Desarrollo de componentes de software	Desarrollo ágil de software	Calidad de software	Programación orientada a objetos	Estructura de datos
Bases de datos para proyectos de software	Introducción a los datos	Diagramas, estructuras y modelos de bases de datos	Bases de datos relacionales	Bases de datos NoSQL



Nivel: Duodécimo

Tabla 8

Unidades de estudio por subárea

Subárea	Unidad de estudio 1	Unidad de estudio 2	Unidad de estudio 3
Desarrollo de componentes de software	Programación orientada a objetos avanzada	Aplicaciones móviles	Comunicación entre componentes de software
Experiencia de usuario diseño de interacciones para componentes de software	Experiencias de usuario (UX)	Diseño de interacciones (IxD)	NA

NA: No aplica.



Nivel: Décimo

Subárea: Herramientas para la producción, transformación digital y seguridad de datos

Tabla 9

Resultados de aprendizaje por unidad de estudio y tiempo estimado

Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
Alfabetización digital para el bienestar y la seguridad en línea	32	<ul style="list-style-type: none">• Examinar el concepto de alfabetización digital y su impacto en la vida cotidiana.• Implementar buenas prácticas que promuevan el uso saludable y responsable de la tecnología y su impacto en la salud mental y física de las personas.• Aplicar técnicas de autocontrol para evitar la adicción digital y equilibrar el uso de la tecnología y propiciar el bienestar personal.• Aplicar los principios de ciberseguridad y el comportamiento ético en entornos digitales.



Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
		<ul style="list-style-type: none">• Investigar sobre el uso de herramientas digitales que potencien la creatividad y el aprendizaje de forma equilibrada.• Diseñar el plan personal de salud digital.
Transformación digital, seguridad y análisis de la información	40	<ul style="list-style-type: none">• Explorar el concepto y la importancia de la transformación digital en diferentes sectores.• Identificar los principios de la seguridad informática y medidas de protección de datos.• Analizar información digital utilizando herramientas tecnológicas para la toma de decisiones• Desarrollar estrategias de autoaprendizaje para actualización de contenidos en entornos digitales cambiantes.
Herramientas para la producción de documentos	88	<ul style="list-style-type: none">• Aplicar las funciones básicas del procesador de textos en la elaboración de documentos.• Utilizar las herramientas que presenta la hoja electrónica para la elaboración de documentos.• Generar presentaciones con los elementos básicos del editor, para la presentación de documentos de forma dinámica.



Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
		<ul style="list-style-type: none">• Explicar los elementos que integran el entorno web.• Aplicar las herramientas colaborativas para la elaboración de documentos en la nube.• Emplear técnicas de navegación y el uso de plataformas de comunicación y colaboración, adoptando conductas seguras.• Implementar procesos de autoaprendizaje que propicien el uso herramientas ofimáticas mediante software de código abierto y licenciado.• Utilizar las tecnologías como recurso, profundizando y dinamizando el aprendizaje, en respuesta a situaciones de la vida cotidiana.



Subárea: Tecnologías de información

Tabla 10

Resultados de aprendizaje por unidad de estudio y tiempo estimado

Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
Fundamentos de tecnologías de la información	80	<ul style="list-style-type: none">• Emplear los componentes requeridos para la reparación, actualización, armado y desarmado de computadoras personales aplicando los principios de salud ocupacional.• Instalar componentes para la actualización de la computadora, realizando la configuración, según las necesidades del usuario.• Determinar cómo las computadoras se comunican en la red.• Resolver problemas que se presentan en equipos portátiles y otros dispositivos, considerando las características técnicas de los equipos portátiles y otros dispositivos, la conectividad y la configuración de acuerdo con las necesidades del usuario.• Instalar sistemas operativos licenciados y de código abierto, según las tendencias del mercado actual.



Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
		<ul style="list-style-type: none">• Analizar aspectos del entorno, requeridos para la implementación de seguridad en equipos, los datos, la red y la función del profesional de tecnologías de información.• Desarrollar las labores de manera responsable, según la planificación, instrucciones y normas establecidas.• Argumentar sobre el impacto ambiental y tecnológico que genera el uso de las tecnologías de información en la sociedad.
Fundamentos de ciberseguridad	80	<ul style="list-style-type: none">• Explicar las amenazas, vulnerabilidades, ataques y problemas presentes en la seguridad de la información en la era digital.• Emplear protección en los sistemas operativos y dispositivos finales en la era de la ciberseguridad de la información.• Explicar las mejores prácticas para la ciberseguridad de la información en la era digital.• Explicar la seguridad en la nube, criptografía, monitoreo y la gobernanza en la ciberseguridad de la información.



Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
		<ul style="list-style-type: none">• Explicar en qué consiste la inteligencia contra amenazas, gestión de riesgos, evaluación de vulnerabilidades, análisis y respuesta ante incidentes de ciberseguridad.• Utilizar técnicas que propicien el desarrollo de la capacidad proactiva.• Interpretar los objetivos para el desarrollo sostenible según lo establecido por la Asamblea General de las Naciones Unidas en la agenda 2030.



Subárea: Desarrollo de componentes de software

Tabla 11

Resultados de aprendizaje por unidad de estudio y tiempo estimado

Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
Herramientas lógicas	48	<ul style="list-style-type: none">• Resolver problemas utilizando los sistemas numéricos y el razonamiento lógico.• Aplicar la lógica proposicional y la lógica de predicados en la determinación de validez de la proposición dada.• Emplear algoritmos, matrices, álgebra de matrices y mapas de Karnaugh en la resolución de problemas.• Implementar acciones orientadas a la resolución de problemas en situaciones propias del área técnica y de la vida cotidiana.• Desarrollar estrategias matemáticas y tecnológicas que le permitan a la persona estudiante sentirse parte de la ciudadanía digital en el mundo globalizado.
Algoritmos y diagramas de flujo	48	<ul style="list-style-type: none">• Aplicar algoritmos y diagramas de flujo estructurado como herramientas para resolución lógica de problemas computacionales.



Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
		<ul style="list-style-type: none">• Aplicar técnicas de diagramación en la resolución de problemas, utilizando simbología para la construcción de algoritmos, ciclos y estructuras condicionales.• Analizar la importancia de la toma de decisiones en búsqueda del logro de las metas propuestas y la sana convivencia.• Contrastar las implicaciones económicas, socioculturales y éticas del uso de la tecnología.
Software y su ingeniería	40	<ul style="list-style-type: none">• Explicar la importancia de la ingeniería del software y los modelos de procesos en el desarrollo aplicaciones de software.• Explicar el diseño desde el contexto de la ingeniería del software para el desarrollo de aplicaciones de software.• Examinar el diseño arquitectónico de software para el desarrollo de aplicaciones.• Explicar la importancia del diseño en el nivel de componentes de la ingeniería de software presentes en el desarrollo de aplicaciones de software.



Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
		<ul style="list-style-type: none">• Ejemplificar el uso de la resolución de problemas en situaciones de la vida real y la carrera técnica.• Argumentar la importancia de la ciudadanía planetaria en la actualidad.
Levantamiento de requerimientos	64	<ul style="list-style-type: none">• Explicar la importancia de la ingeniería de requerimientos para el desarrollo de aplicaciones de software.• Utilizar el modelado de requerimientos para el desarrollo de aplicaciones de software.• Emplear el modelado de requerimientos orientado al flujo, comportamiento para el desarrollo de aplicaciones de software.• Implementar acciones que favorezcan la realización de actividades de manera colaborativa con el propósito de alcanzar el cumplimiento de metas comunes.• Argumentar acerca de cómo la robótica se ha desarrollado a partir del uso de las tecnologías de información.



Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
Introducción a la programación	120	<ul style="list-style-type: none">• Aplicar los principios fundamentales, características de los lenguajes y elementos de la programación, en el uso e instalación de entornos de desarrollo.• Implementar soluciones a problemas básicos mediante la aplicación de la lógica de programación, utilizando estructuras, el diseño de algoritmos y diagramas de flujo.• Programar aplicaciones de software que integre tipos de datos, variables y estructuras de programación, para la resolución de problemas en contextos reales o simulados.• Implementar los principios de programación en la resolución de problemas integrando funciones y modularidad.• Utilizar archivos para la extracción, modificación y resguardo de la información, aplicando prácticas seguras y generalidades básicas de la Programación Orientada a Objetos (POO).• Demostrar conductas que reflejen compromiso ético aplicando principios y valores en las situaciones de aprendizaje, que vivencia en el área técnica y en las normas de convivencia con los que le rodean.



Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
		<ul style="list-style-type: none">• Seleccionar estrategias para el desarrollo de la programación con ayuda de la tecnología, en el modelo de equidad social.



Nivel: Undécimo

Subárea: Emprendimiento e innovación aplicadas al desarrollo de aplicaciones de software

Tabla 12

Resultados de aprendizaje por unidad de estudio y tiempo estimado

Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
Oportunidades de negocio	40	<ul style="list-style-type: none">• Explicar las características esenciales e importancia del emprendimiento haciendo uso productivo de las tecnologías.• Examinar el mercado y su entorno, aplicando herramientas de recolección de información para la identificación de oportunidades de negocio, según las nuevas tendencias.• Utilizar técnicas creativas que permitan la generación de ideas de negocio innovadoras, brindando soluciones a las necesidades detectadas en los clientes potenciales.• Proponer soluciones creativas e innovadoras a necesidades y oportunidades del mercado.• Valorar el impacto social, económico y ambiental que generan las propuestas de proyectos de negocios sostenibles.



Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
Modelos de negocios	32	<ul style="list-style-type: none">• Construir modelos de negocios a partir de ideas innovadoras con propuestas de valor diferenciadoras, utilizando las herramientas y metodologías vigentes.• Validar el modelo de negocio, mediante el diseño de productos mínimos viable aplicando metodologías vigentes.• Desarrollar el plan de puesta en marcha del modelo de negocio y lanzamiento del producto.• Aplicar estrategias de negociación en el proceso de validación de propuestas de negocios.• Validar propuestas de negocios tomando en consideración el compromiso con la sociedad local y global.
Creación de la empresa	68	<ul style="list-style-type: none">• Describir los tipos de empresas con los cuales se puede desarrollar un negocio.• Estructurar el negocio con el enfoque orientado al cliente a través del plan de negocio.• Realizar labores en las áreas funcionales que conforman la empresa de práctica propuesta aplicando los principios de la administración y lo establecido en el plan de negocios.



Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
		<ul style="list-style-type: none">• Aplicar los principios de servicio con enfoque orientado al cliente en la puesta en marcha del plan de negocio.• Elegir las mejores estrategias para búsqueda de información a través del uso de las tecnologías de forma individual o colaborativa.
Plan de vida	20	<ul style="list-style-type: none">• Estimar el nivel alcanzado en la gestión del emprendimiento según las metas y objetivos propuestos en el plan de negocio, para la obtención de la certificación empresarial.• Evaluar las oportunidades que ofrece la sociedad para el desarrollo y consolidación del emprendimiento.• Emplear el aprendizaje permanente como herramienta en el desarrollo de competencias para el fortalecimiento de su desempeño en el área de formación técnica, personal y el de su plan de vida.• Planificar su vida, considerando sus competencias, recursos y el entorno, contribuyendo al desarrollo de la cultura emprendedora.



Subárea: Desarrollo de componentes de software

Tabla 13

Resultados de aprendizaje por unidad de estudio y tiempo estimado

Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
Desarrollo ágil de software	72	<ul style="list-style-type: none">• Explicar la importancia del desarrollo ágil de software para proyectos de desarrollo de aplicaciones.• Determinar el uso de herramientas para las metodologías ágiles en proyectos para el desarrollo de aplicaciones de software.• Emplear el uso de GIT y GitHub para el seguimiento, versionamiento para proyectos de desarrollo de aplicaciones de software.• Analizar la importancia de la creatividad en la carrera técnica de desarrollo de aplicaciones de software.• Discutir acerca de la importancia de reconocer los problemas socio ambientales globales en la actualidad.• Explicar la importancia del desarrollo ágil de software para proyectos de desarrollo de aplicaciones software.



Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
Calidad de software	72	<ul style="list-style-type: none">• Explicar la importancia de la calidad de software en el desarrollo de aplicaciones de software.• Analizar el impacto de los defectos y el aseguramiento de la calidad en el software para desarrollo de aplicaciones.• Explicar la importancia de las pruebas de software para el desarrollo de aplicaciones.• Ilustrar la importancia de la atención al detalle en la carrera técnica y para la vida.• Explicar el impacto de la ciudadanía planetaria en el mundo global.
Programación orientada a objetos (POO)	104	<ul style="list-style-type: none">• Utilizar la programación orientada a objetos en la resolución básica de problemas de programación.• Aplicar el uso de métodos y clases en la creación de tareas y aplicaciones informáticas.• Aplicar expresiones booleanas, condicionales y ciclos en la resolución de problemas básicos de tareas y aplicaciones informáticas.• Utilizar clases en la creación de tareas y aplicaciones informáticas para la programación orientada a objetos.



Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
		<ul style="list-style-type: none">• Utilizar arreglos unidimensionales, manejo de excepciones y depuración en la resolución de problemas básicos de tareas y aplicaciones informáticas.• Explicar la importancia de la inteligencia emocional en la vida diaria y profesional.• Analizar el impacto del cambio climático en la actualidad y la necesidad de conservar la biodiversidad.
Estructura de datos	72	<ul style="list-style-type: none">• Examinar los fundamentos de las estructuras de datos incluyendo su clasificación, eficiencia, rendimiento y escalabilidad.• Utilizar estructuras tales como: arreglos unidimensionales, multidimensionales y listas enlazadas.• Diseñar aplicaciones de software que incorporen pilas y colas.• Elaborar aplicaciones de software aplicando funciones de recursividad.• Diseñar aplicaciones de software utilizando estructuras de grafos, algoritmos de recorrido y búsqueda.



Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
		<ul style="list-style-type: none">• Diseñar aplicaciones de software utilizando estructuras de datos y buenas prácticas, de acuerdo con los requerimientos.• Explicar el impacto que posee la gestión del tiempo en el ámbito personal como profesional de las personas.• Argumentar la importancia de la biodiversidad, el uso responsable y sostenible de los recursos naturales en la actualidad.



Subárea: Bases de datos para proyectos de software

Tabla 14

Resultados de aprendizaje por unidad de estudio y tiempo estimado

Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
Introducción a los datos	32	<ul style="list-style-type: none">• Explicar las características de los datos, usos, tipos y aspectos básicos del análisis de los datos.• Describir características, elementos y conceptos fundamentales asociados con las bases de datos.• Demostrar conductas que reflejen compromiso ético, aplicando principios de gestión de datos responsable.• Ejercer acciones cotidianas que contribuyen a su bienestar y el de los demás.
Diagramas, estructuras y modelos de bases de datos	80	<ul style="list-style-type: none">• Emplear los elementos que forman parte del modelado de datos en las bases de datos• Contrastar los elementos, el diseño y características, implementados en el diseño de bases de datos.• Implementar las formas normales utilizadas en el diseño de bases de datos.



Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
		<ul style="list-style-type: none">• Demostrar conductas que reflejen compromiso ético, en la gestión de datos.• Argumentar sobre el concepto, importancia y buenas prácticas de desarrollo sostenible en procesos productivos relacionados con el sector de TI.
Bases de datos relacionales	104	<ul style="list-style-type: none">• Explicar los lenguajes relacionales utilizados en la creación de bases de datos relacionales.• Crear bases de datos utilizando el lenguaje relacional SQL.• Demostrar conductas que reflejen compromiso ético, aplicando principios de la gestión de datos.• Utilizar las tecnologías digitales como recurso para dinamizar el aprendizaje dentro de su vida profesional y cotidiana.
Bases de datos no relacionales NoSQL	104	<ul style="list-style-type: none">• Examinar los conceptos básicos sobre bases de datos NoSQL.• Emplear los tipos de almacenamiento utilizados en bases de datos NoSQL, para la resolución de problemas.• Aplicar conceptos, comandos y operadores esenciales en la implementación de bases de datos NoSQL.



Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
		<ul style="list-style-type: none">• Implementar acciones orientadas a la resolución de problemas en situaciones propias del área técnica y de la vida cotidiana.• Utilizar las tecnologías digitales como insumo para el aprendizaje en el desarrollo de su campo de formación técnica.



Nivel: Duodécimo

Subárea: Desarrollo de componentes de software

Tabla 15

Resultados de aprendizaje por unidad de estudio y tiempo estimado

Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
Programación orientada a objetos avanzada	108	<ul style="list-style-type: none">• Aplicar estructuras avanzadas de programación orientada a objetos en entornos concurrentes, asincrónicos y distribuidos.• Diseñar sistemas orientados a objetos aplicando arquitecturas modernas y principios de diseño evolutivo.• Implementar patrones de diseño avanzados en entornos de alta variabilidad, aplicando principios de extensión flexible y bajo acoplamiento para el diseño de componentes adaptativos.• Integrar herramientas de calidad y pruebas automatizadas orientadas a objetos en el ciclo de vida del software que garanticen robustez, legibilidad y mantenibilidad del código.• Desarrollar soluciones orientadas a objetos que integren el uso de inteligencia artificial (IA) y objetos inteligentes.



Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
		<ul style="list-style-type: none">• Emplear alternativas para el desarrollo de la empatía que promuevan la sana convivencia.• Planificar alternativas de solución, tanto individuales como colectivas, concientizando a otros respecto a los cambios que deben hacerse en los hábitos de consumo promovidos por la sociedad.
Aplicaciones móviles (APPS)	120	<ul style="list-style-type: none">• Analizar requerimientos funcionales y no funcionales para el desarrollo de aplicaciones móviles, considerando necesidades reales del usuario y principios de diseño.• Diseñar interfaces de usuario móviles, funcionales y accesibles, considerando los principios del diseño visual.• Desarrollar aplicaciones móviles funcionales, seguras y eficientes, integrando componentes comunes.• Aplicar pruebas funcionales y de usabilidad que optimicen el desarrollo de aplicaciones móviles, garantizando su rendimiento, seguridad y usabilidad.



Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
		<ul style="list-style-type: none">• Publicar aplicaciones móviles aplicando la normativa legal y ética vigente; y considerando la retroalimentación de los usuarios para la mejora continua.• Proponer soluciones creativas e innovadoras a necesidades y problemas cotidianos del contexto.• Examinar el concepto, características e importancia de la sostenibilidad en el desarrollo de aplicaciones móviles.
Comunicación entre componentes de software	72	<ul style="list-style-type: none">• Analizar los tipos de componentes de software y sus responsabilidades dentro de los sistemas distribuidos o modulares.• Aplicar mecanismos de comunicación síncronos y asíncronos entre componentes, evaluando su idoneidad, según los requerimientos funcionales y no funcionales del sistema.• Diseñar interfaces de comunicación entre componentes, utilizando contratos definidos que garanticen interoperabilidad, mantenibilidad y escalabilidad de aplicaciones de software.• Implementar mecanismos de integración entre componentes de software heterogéneos.



Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
		<ul style="list-style-type: none">• Validar la comunicación entre componentes mediante pruebas de integración y monitoreo.• Emplear formas de comunicación asertiva en la convivencia con las personas.• Argumentar sobre cómo la discriminación genera desigualdad.



Subárea: Experiencia de usuarios y diseño de interacciones para componentes

Tabla 16

Resultados de aprendizaje por unidad de estudio y tiempo estimado

Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
Experiencia de usuario (UX)	104	<ul style="list-style-type: none">• Determinar los fundamentos de la experiencia de usuario (UX) y su impacto en el desarrollo de productos digitales funcionales y accesibles.• Interpretar las necesidades, comportamientos y expectativas de los usuarios reales, mediante métodos innovadores de observación y recolección de datos.• Diseñar soluciones UX innovadoras, utilizando herramientas de ideación, prototipado y diseño iterativo en la resolución de problemas.• Verificar la experiencia de usuario, mediante la implementación de pruebas controladas e instrumentos de retroalimentación.• Comunicar propuestas de experiencia de usuario (UX) de manera creativa y visual, adaptando el mensaje a diversos públicos.• Aplicar estrategias de negociación que propicien acuerdos exitosos.• Argumentar sobre cómo la discriminación genera desigualdad.



Unidad de estudio	Tiempo estimado	Resultados de aprendizaje
Diseño de interacciones (IxD)	96	<ul style="list-style-type: none">• Aplicar los principios fundamentales del diseño de interacciones para la creación de experiencias digitales, coherentes, intuitivas y centradas en las personas.• Diseñar interacciones digitales que promuevan la equidad social, la accesibilidad y la participación activa de distintos perfiles de usuarios.• Elaborar el diseño y desarrollo de prototipos interactivos aplicando herramientas colaborativas.• Desarrollar prototipos funcionales de interacciones digitales utilizando herramientas de diseño.• Verificar las interacciones digitales propuestas, considerando criterios de usabilidad, accesibilidad, satisfacción del usuario y valor social del producto digital.• Aplicar los principios de discernimiento y responsabilidad en el manejo y protección de los datos.• Interpretar los Objetivos para el Desarrollo sostenible según lo establecido por la Asamblea General de las Naciones Unidas en la agenda 2030.



MINISTERIO DE
EDUCACIÓN PÚBLICA

GOBIERNO
DE COSTA RICA

Dirección de Educación
Técnica y Capacidades
Emprendedoras

Subárea Emprendimiento e innovación aplicadas al desarrollo de aplicaciones de software



Subárea: Emprendimiento e innovación aplicadas al desarrollo de aplicaciones de software



Desarrollo de
Aplicaciones
de Software



Nivel

Undécimo



Juega para aprender



"Motívate a la salud digital en tu aprendizaje"





Descripción de la Subárea Emprendimiento e innovación aplicadas al desarrollo de aplicaciones de software

La Dirección de Educación Técnica y Capacidades Emprendedoras a través del desarrollo de programas y proyectos educativos, la capacitación, y actualización fomenta como uno de sus pilares el desarrollo de competencias en el ámbito del emprendimiento y la empresarialidad; brindando a la persona estudiante, oportunidades para la formación ética, académica y profesional además de las herramientas que le posibilite ser forjador de su proyecto de vida, en beneficio propio y el de la sociedad.

La subárea Emprendimiento e innovación tiene como propósito fomentar que la persona estudiante logre la siguiente competencia: Desarrollar capacidades en los ámbitos del emprendimiento y la empresarialidad mediante la identificación de oportunidades de negocios, la aplicación de metodologías para la construcción de modelos de negocios; la creación de empresas de práctica y la creación de su proyecto de vida tomando en consideración sus competencias, recursos, el entorno y su compromiso local y social.

Con la incorporación de la subárea en el plan de estudios de las carreras técnicas de la Educación Técnica Profesional (ETP), se contribuye al desarrollo de una cultura emprendedora; a la luz de las recomendaciones propuestas por la Organización Internacional del Trabajo (OIT), la Organización para Cooperación y el Desarrollo Económico (OCDE), la política educativa del MEP “La persona: centro del proceso educativo y sujeto transformador de la sociedad”, así como los objetivos de la agenda 2030; los cuales se enfocan en que la empresarialidad y emprendimiento sean procesos constantes en los sistemas educativos que proveen



emprendedores al mercado laboral. La subárea tiene una duración de 160 horas y posee cuatro unidades de estudio.

Tabla 17

Distribución de unidades de estudio de la subárea Emprendimiento e innovación aplicadas al desarrollo de aplicaciones de software

Unidades de estudio	N° semanas	N° horas anuales
1 Oportunidades de negocios	10	40
2 Modelos de negocios	8	32
3 Creación de la empresa	17	68
4 Plan de vida	5	20
Total	40	160



Tabla 18

Información administrativa

Carrera técnica: Desarrollo de Aplicaciones de Software	Campo detallado: 0619 Tecnologías de la información y la comunicación (TIC)
Subárea: Emprendimiento e innovación aplicada al desarrollo de software	Nivel: Undécimo
Unidad de estudio: Oportunidades de negocios	Tiempo estimado: 40 horas
Competencias para el desarrollo humano: Innovación y creatividad	Eje política educativa: Educación para el desarrollo sostenible

Tabla 19

Planificación curricular de la unidad de estudio



Resultados de Aprendizaje	Saberes Esenciales	Indicador de logro ⁸
1. Explicar las características esenciales e importancia del emprendimiento haciendo un uso productivo de las tecnologías.	<p>Emprendimiento</p> <ul style="list-style-type: none">• Definición, características e importancia del fomento del espíritu emprendedor• Características de la cultura emprendedora• Habilidades y responsabilidades de un emprendedor• Importancia de ser emprendedor en su proyecto de vida• Elementos para tomar en cuenta al emprender un proyecto<ul style="list-style-type: none">• Justificación del proyecto• Estudio del mercado• Trámites administrativos y legales• Fuentes de financiamiento• Análisis integral• Uso productivo de las tecnologías en los negocios	<ul style="list-style-type: none">• Identifica habilidades y responsabilidades de la persona emprendedora.• Discrimina los elementos a tomar en cuenta al emprender un proyecto.• Explica el uso productivo de las tecnologías en la generación de ideas de negocios.

⁸ Indicadores para la macroevaluación.



Resultados de Aprendizaje	Saberes Esenciales	Indicador de logro ⁸
2. Examinar el mercado y su entorno, aplicando herramientas de recolección de información para la identificación de oportunidades de negocio, según las nuevas tendencias.	<p>Mercado</p> <ul style="list-style-type: none">• Concepto• Funcionamiento del mercado y tendencias innovadoras• Análisis del entorno• Oportunidades de negocios• Necesidades sociales• Problemáticas• Herramientas para detectar necesidades• Detección del mercado y clientes potenciales• El cliente como elemento clave	<ul style="list-style-type: none">• Caracteriza el funcionamiento del mercado y su dinámica.• Identifica las oportunidades del mercado según las nuevas tendencias.• Utiliza herramientas para la recolección de información que permita la detección de oportunidades de negocio.• Interpreta los resultados obtenidos en función del mercado y los clientes potenciales.
3. Utilizar técnicas creativas que permitan la generación de ideas de negocio innovadoras, brindando soluciones a las necesidades detectadas en los clientes potenciales.	<p>Generación de ideas empresariales</p> <ul style="list-style-type: none">• Concepto• Fuentes• Propósito• Necesidad de una idea• Respuesta a las necesidades del mercado• Cambios en la moda y los requisitos	<ul style="list-style-type: none">• Determina fuentes de generación de ideas empresariales.• Selecciona ideas empresariales usando distintas técnicas.• Aplica técnicas creativas que brinden soluciones a las necesidades



Resultados de Aprendizaje	Saberes Esenciales	Indicador de logro ⁸
	<ul style="list-style-type: none">• Mantenerse a la cabeza de la competencia• Tecnología <p>Técnicas para generar ideas empresariales</p> <ul style="list-style-type: none">• Características, utilidad y beneficios• Herramientas que apoyan el proceso de selección del mejor producto• Diseño de una idea de negocio innovadora	detectadas en los clientes potenciales.
4. Proponer soluciones creativas e innovadoras a necesidades y oportunidades del mercado.	<p>Creatividad e Innovación</p> <ul style="list-style-type: none">• Concepto• Importancia• El proceso de la creatividad y la habilidad de pensar creativamente• Innovación y su proceso• Tipos de innovación y cómo diferenciarlos	<ul style="list-style-type: none">• Identifica la importancia de la creatividad e innovación en los aspectos cotidianos de su quehacer.• Fomenta en el entorno una actitud creativa e innovadora en el desarrollo de emprendimientos.• Formula soluciones para las necesidades y oportunidades del mercado o mejora las existentes.



Resultados de Aprendizaje	Saberes Esenciales	Indicador de logro ⁸
5. Valorar el impacto social, económico y ambiental que generan las propuestas de proyectos de negocios sostenibles.	<p>Desarrollo sostenible</p> <ul style="list-style-type: none">• Concepto• Importancia• Elementos<ul style="list-style-type: none">• Social• Económico• Ambiental• Emprendimientos sostenibles	<ul style="list-style-type: none">• Describe los elementos del desarrollo sostenible y su importancia.• Discrimina el impacto al ambiente y a la salud producto del desarrollo de nuevos negocios.• Propone acciones creativas que mitiguen los daños al ambiente como parte del desarrollo de emprendimientos sostenibles.



Tabla 20

Información administrativa

Carrera técnica: Desarrollo de Aplicaciones de Software	Campo detallado: 0619 Tecnologías de la información y la comunicación (TIC)
Subárea: Emprendimiento e innovación aplicadas al desarrollo de aplicaciones de software	Nivel: Undécimo
Unidad de estudio: Modelo de negocios	Tiempo estimado: 32 horas
Competencias para el desarrollo humano: Capacidad de negociación	Eje política educativa: Fortalecimiento de una ciudadanía planetaria con identidad

Tabla 21

Planificación curricular de la unidad de estudio

Resultados de Aprendizaje	Saberes Esenciales	Indicador de logro
1. Construir modelos de negocios a partir de ideas innovadoras con propuestas de valor diferenciadoras, utilizando las herramientas y metodologías vigentes.	Modelos de negocios <ul style="list-style-type: none">• Concepto• Aspectos para considerar<ul style="list-style-type: none">• Clientes• Canales• Relación con los clientes	<ul style="list-style-type: none">• Distingue los aspectos que se consideran en la construcción de un modelo de negocio.• Compara las herramientas y metodologías vigentes en la construcción de modelos de negocios.



Resultados de Aprendizaje	Saberes Esenciales	Indicador de logro
	<ul style="list-style-type: none">• Actividades importantes• Recursos• Aliados• Estructura económica y financiera• Tipos de herramientas vigentes y su aplicabilidad<ul style="list-style-type: none">• Pensamiento de diseño (Design Thinking)<ul style="list-style-type: none">• Características.• Otras herramientas vigentes	<ul style="list-style-type: none">• Utiliza herramientas y metodologías vigentes en la construcción de modelos de negocios.• Diseña ideas de negocio con mayor oportunidad de éxito a partir de la aplicación de herramientas y metodologías vigentes.
2. Validar el modelo de negocio, mediante el diseño de un producto mínimo viable aplicando metodologías vigentes.	Producto mínimo viable (PMV) <ul style="list-style-type: none">• Concepto• Pasos de la metodología por ejemplo Lean Startup• Diseño del producto mínimo viable aplicando los pasos de las metodologías vigentes• Validación del modelo de negocio	<ul style="list-style-type: none">• Identifica concepto de producto mínimo viable.• Explica los pasos para la construcción del producto mínimo viable según las metodologías vigentes.• Diseña el producto mínimo viable aplicando los pasos de las metodologías vigentes.
3. Desarrollar el plan de puesta en marcha del modelo de negocio y lanzamiento del producto.	Plan de implementación <ul style="list-style-type: none">• Inversión inicial• Gestión de las finanzas	<ul style="list-style-type: none">• Identifica los aspectos que deben considerarse en la puesta en marcha del modelo de negocios.



Resultados de Aprendizaje	Saberes Esenciales	Indicador de logro
	<ul style="list-style-type: none">• Identificación de fuentes de financiamiento• Aspectos de formalización• Diseño de marca• Plan de mercadeo y ventas• Impactos: social, ambiental y la salud integral	<ul style="list-style-type: none">• Distingue las características de los aspectos que deben considerarse para la implementación del plan de puesta en marcha del modelo de negocio.• Construye el plan de puesta en marcha del modelo de negocios, tomando en cuenta las estrategias de mitigación de impacto.
4. Aplicar estrategias de negociación en el proceso de validación de propuestas de negocios.	Capacidad de negociación <ul style="list-style-type: none">• Concepto• Estrategias para la negociación• Acuerdos para la validación de propuestas de negocios	<ul style="list-style-type: none">• Explica la importancia del desarrollo de habilidades de negociación durante el proceso de validación de propuestas de negocios.• Selecciona estrategias de negociación que propicien acuerdos exitosos durante el proceso de validación de propuestas de negocios.• Negocia la ejecución de propuestas viables de emprendimiento.
5. Validar propuestas de negocios tomando en consideración el	Derechos económicos, sociales, culturales y valores éticos universales	<ul style="list-style-type: none">• Expone propuestas de negocios considerando los derechos económicos,



Resultados de Aprendizaje	Saberes Esenciales	Indicador de logro
compromiso con la sociedad local y global.	<ul style="list-style-type: none">• Trabajo en condiciones justas y favorables• Protección social, a un nivel de vida adecuado y al disfrute del más alto nivel posible de salud física y mental• Educación, libertad cultural y el progreso científico <p>Valores éticos universales</p> <ul style="list-style-type: none">• Respeto• Equidad• Justicia• Honestidad <p>Economía social solidaria</p> <ul style="list-style-type: none">• Concepto• Características• Tipos de formas jurídicas asociativas<ul style="list-style-type: none">• Asociaciones Solidaristas<ul style="list-style-type: none">• Modelo• Beneficios• Requisitos para la conformación• Legislación vigente	<p>sociales, culturales y valores éticos universales de la economía social solidaria.</p> <ul style="list-style-type: none">• Organiza propuestas de negocios considerando los derechos económicos, sociales, culturales y valores éticos universales de la economía social solidaria.• Propone soluciones a problemas reales de la comunidad considerando los tipos de formas jurídicas asociativas de la economía social solidaria.



Resultados de Aprendizaje	Saberes Esenciales	Indicador de logro
	<ul style="list-style-type: none">• Cooperativas<ul style="list-style-type: none">• Modelo• Beneficios• Requisitos para la conformación• Legislación vigente	



Tabla 22

Información administrativa

Carrera técnica: Desarrollo de Aplicaciones de Software	Campo detallado: 0619 Tecnologías de la información y la comunicación (TIC)
Subárea: Emprendimiento e innovación aplicadas al desarrollo de aplicaciones de software	Nivel: Undécimo
Unidad de estudio: Creación de la empresa	Tiempo estimado: 68 horas
Competencias para el desarrollo humano: Orientación de servicio al cliente	Eje política educativa: La ciudadanía planetaria con identidad nacional

Tabla 23

Planificación curricular de la unidad de estudio

Resultados de Aprendizaje	Saberes Esenciales	Indicador de logro
1. Describir los tipos de empresas con los cuales se puede desarrollar negocios.	Tipos de empresas <ul style="list-style-type: none"> • Concepto, características, ventajas y desventajas <ul style="list-style-type: none"> • Según el ámbito de actividad • Según el destino de sus beneficios 	<ul style="list-style-type: none"> • Compara los tipos de empresas que interactúan en el sistema financiero y económico nacional. • Selecciona el tipo de empresa para el desarrollo de su modelo de negocio.



Resultados de Aprendizaje	Saberes Esenciales	Indicador de logro
	<ul style="list-style-type: none">• Según la forma jurídica• Según origen o procedencia de capital• Según el tamaño• Según su actividad desde el punto de vista de la materia que utiliza	
2. Estructurar el negocio con el enfoque orientado al cliente a través del plan de negocio.	<p>Plan de negocios</p> <ul style="list-style-type: none">• Objetivos• Metas• Modelo de negocios• Estudios: mercado, mercadeo, técnico, económico y financiero <p>Estructuración del negocio, según el modelo empresarial</p> <ul style="list-style-type: none">• Constitución legal.• Modalidades de contratación según la legislación costarricense• Permisos de funcionamiento y/o patentes• Permisos de salud• Inscripción en Hacienda y Caja Costarricense de Seguro Social como patrono• Catálogo de productos	<ul style="list-style-type: none">• Identifica los elementos que conforman el plan de negocios.• Diseña el plan de negocios, considerando todos sus elementos.• Elabora la estructura organizativa, procesos y procedimientos de la empresa, basándose en el plan de negocios y utilizando el enfoque orientado al cliente,



Resultados de Aprendizaje	Saberes Esenciales	Indicador de logro
3. Realizar labores en las áreas funcionales que conforman la empresa de práctica propuesta aplicando los principios de la administración y lo establecido en el plan de negocios.	<ul style="list-style-type: none">• Estructura organizativa de la empresa utilizando cadena de valor orientada al cliente.• Unidades y departamentos de la empresa• Procesos y procedimientos del negocio• Asociatividad, encadenamientos y clúster• Principios de la administración• Uso de la tecnología como aliado estratégico para la operación de la empresa• Roles de trabajo por áreas funcionales• Puesta en operación del negocio<ul style="list-style-type: none">• Transacciones comerciales• Centro de Operaciones• Registro de las empresas• Transacciones bancarias• Compra y venta de bienes y servicios entre empresas• Compras del Estado• Uso eficiente de los datos para la toma de decisiones• Pago de impuestos• Cargas sociales	<ul style="list-style-type: none">• Identifica las áreas funcionales y labores que se ejecutan para la puesta en marcha del negocio.• Utiliza la tecnología en las transacciones y otras actividades propias de la operación del negocio, incrementando la productividad de la empresa.• Ejecuta experiencias educativas mediante la simulación de una empresa de práctica.



Resultados de Aprendizaje	Saberes Esenciales	Indicador de logro
4. Aplicar los principios de servicio con un enfoque orientado al cliente en la puesta en marcha del plan de negocio.	<ul style="list-style-type: none">• Pólizas y seguros• Asesoría empresarial <p>Enfoque orientado al cliente</p> <ul style="list-style-type: none">• Concepto• Cliente• Servicio al cliente• Importancia• Diferencia entre el servicio y la atención al cliente• Triángulo del servicio <p>Estrategias de servicio al cliente</p> <ul style="list-style-type: none">• Acuerdos de niveles de servicio• La evaluación del servicio• Manejo de quejas, reclamos y sugerencias• Retención y fidelización de clientes• Sistemas de gestión de las relaciones con los clientes• Valores que mejoran el servicio al cliente	<ul style="list-style-type: none">• Explica la diferencia entre atención y servicio al cliente.• Emplea estrategias de servicio al cliente en la puesta en marcha del plan de negocio.• Desarrolla su plan de negocio, considerando el cliente como el eje principal sobre el cual gira su emprendimiento.
5. Elegir las mejores estrategias para búsqueda de información a través del uso de las tecnologías de forma individual o colaborativa.	<p>Herramientas para la productividad</p> <ul style="list-style-type: none">• Redes sociales• Blog• Wikis• Software específico• Herramientas ofimáticas	<ul style="list-style-type: none">• Valora implicaciones económicas, socioculturales y éticas del uso de las tecnologías en la creación de la empresa.



Resultados de Aprendizaje	Saberes Esenciales	Indicador de logro
	<ul style="list-style-type: none">Otras herramientas que faciliten la mediación pedagógica	<ul style="list-style-type: none">Aplica herramientas tecnológicas vigentes en el mercado para la operación de su empresa de práctica.



Tabla 27

Información administrativa

Carrera técnica: Desarrollo de Aplicaciones de Software	Campo detallado: 0619 Tecnologías de la información y la comunicación (TIC)
Subárea: Emprendimiento e innovación aplicadas al desarrollo de aplicaciones de software	Nivel: Undécimo
Unidad de estudio: Plan de vida	Tiempo estimado: 20 horas
Competencias para el desarrollo humano: Autoaprendizaje	Eje política educativa: Fortalecimiento de una ciudadanía planetaria con identidad

Tabla 28

Planificación curricular de la unidad de estudio

Resultados de Aprendizaje	Saberes Esenciales	Indicador de logro
1. Estimar el nivel alcanzado en la gestión del emprendimiento según las metas y objetivos propuestos en el plan de	Evaluación de la empresa a través de indicadores	<ul style="list-style-type: none">Determina el nivel de logro según los indicadores propuestos para la certificación.



Resultados de Aprendizaje	Saberes Esenciales	Indicador de logro
negocio, para la obtención de la certificación empresarial.	<ul style="list-style-type: none">• Sistematización de resultados• Valoración de los logros alcanzados• Resumen ejecutivo de lecciones aprendidas• Conclusiones• Recomendaciones <p>Certificación de empresa</p> <ul style="list-style-type: none">• Procedimiento• Revisión de los alcances del plan de negocios según indicadores	<ul style="list-style-type: none">• Describe los resultados de la empresa a través de la revisión de indicadores de certificación.• Sistematiza los resultados obtenidos durante el periodo de funcionamiento de la empresa, en función de la certificación de empresa.• Aplica lecciones aprendidas en su desarrollo personal y profesional, adaptándose a un entorno cambiante.
2. Evaluar las oportunidades que ofrece la sociedad para el desarrollo y consolidación del emprendimiento.	<p>Instituciones de apoyo al emprendimiento nacional</p> <ul style="list-style-type: none">• Incubadoras y aceleradoras de Empresas• Ministerio de Economía, Industria y Comercio• Sistema de Banca para el Desarrollo	<ul style="list-style-type: none">• Examina las áreas de acción y los requerimientos que establecen las instituciones de apoyo para el desarrollo y consolidación del emprendimiento.• Identifica los procesos requeridos para la formalización del emprendimiento en las instituciones de apoyo.



Resultados de Aprendizaje	Saberes Esenciales	Indicador de logro
3. Emplear el aprendizaje permanente como herramienta en el desarrollo de competencias para el fortalecimiento de su desempeño en el área de formación técnica, personal y el de su plan de vida.	<ul style="list-style-type: none">• Sistema Bancario Nacional público y privada• INFOCOOP• Otros operadores financieros• Instituciones de apoyo	<ul style="list-style-type: none">• Diseña la propuesta de formalización considerando los requerimientos establecidos por la institución de apoyo seleccionada.
	<p>Aprendizaje permanente</p> <ul style="list-style-type: none">• Autoaprendizaje<ul style="list-style-type: none">• Concepto de aprendizaje• ¿Qué significa aprender a aprender?• Utilidad del autoaprendizaje.• Motivación para aplicar el autoaprendizaje• Adaptabilidad a nuevas situaciones• Importancia del autoaprendizaje en el área de formación técnica.	<ul style="list-style-type: none">• Identifica las competencias específicas y para el desarrollo humano alcanzadas a través del proceso educativo y su relación con el entorno.• Propone ideas innovadoras propias de su área de formación técnica, aplicando sus conocimientos, habilidades y destrezas como parte del proceso de gestión de su plan de vida.
	<p>Competencias</p> <ul style="list-style-type: none">• Específicas• Para el desarrollo humano	<ul style="list-style-type: none">• Enriquece su proyecto de vida aprovechando las oportunidades de aprendizaje disponibles, los obstáculos y las competencias desarrolladas.



Resultados de Aprendizaje

Saberes Esenciales

Indicador de logro

4. Planificar su vida, considerando sus competencias, recursos y el entorno, contribuyendo al desarrollo de una cultura emprendedora.

Plan de vida

- Concepto
- Aspectos para considerar en la elaboración de un plan de vida a corto, mediano y a largo plazo
 - Sociales
 - Económicos
 - Personales

- Toma conciencia de sus competencias y limitaciones y lo pone en práctica de acuerdo con su contexto.
- Desarrolla estrategias individuales y colectivas que propicien el logro de las metas propuestas.



MINISTERIO DE
EDUCACIÓN PÚBLICA

GOBIERNO
DE COSTA RICA

Dirección de Educación
Técnica y Capacidades
Emprendedoras

Subárea Desarrollo de componentes de software



Subárea: Desarrollo de componentes de software

»» Desarrollo de
Aplicaciones
de Software



Nivel

Undécimo

Juega para aprender



"Motívate a la salud digital en tu aprendizaje"





Descripción de la Subárea Desarrollo de componentes de software

La subárea de Desarrollo de Componentes de Software busca fortalecer las competencias técnicas, éticas y colaborativas del estudiantado para la creación de soluciones informáticas innovadoras, sostenibles y de calidad, mediante la aplicación de metodologías ágiles, principios de programación avanzada y estructuras eficientes de datos, en entornos reales o simulados.

Esta subárea abarca 160 horas anuales y está compuesta por cinco unidades de estudio: Desarrollo ágil de software, Calidad de software, Programación orientada a objetos (POO), Estructura de datos, y Programación orientada a objetos avanzada (si aplica en continuidad), que en su conjunto permiten abordar el ciclo completo de diseño, desarrollo y mejora de componentes de software.

La unidad de estudio Desarrollo ágil de software, desarrolla capacidades en el estudiantado orientadas a la comprensión y aplicación de los valores y principios de las metodologías ágiles, promoviendo la ética profesional, la sana convivencia y la colaboración efectiva con el cliente. Se desarrollan proyectos empleando marcos de trabajo ágiles, priorización de requerimientos, integración de buenas prácticas técnicas y uso de bases de datos en ciclos de desarrollo continuo.

La unidad de estudio Calidad de software fomenta la cultura de la mejora continua mediante la aplicación de estándares internacionales, herramientas de aseguramiento de calidad, y estrategias de evaluación que permiten garantizar productos confiables, funcionales y centrados en la satisfacción del usuario. Se valora la



toma de decisiones informada, la transparencia en los procesos y la comunicación efectiva con los diferentes actores del proyecto.

La Programación Orientada a Objetos (POO), facilita el desarrollo de soluciones modulares, escalables y reutilizables mediante el uso creativo de clases, objetos, herencia, polimorfismo, principios SOLID y patrones de diseño. Se refuerzan prácticas de documentación, manejo de errores y validación de soluciones, fortaleciendo el pensamiento estructurado y la innovación técnica.

La unidad de Estructura de datos profundiza en el diseño y aplicación de estructuras eficientes, tales como: listas, pilas, colas, grafos y algoritmos de recorrido, ordenamiento y búsqueda, seleccionando las más adecuadas de acuerdo con los requerimientos de rendimiento y funcionalidad del software. Se promueve la optimización del código, el trabajo en equipo y el uso ético y proactivo de buenas prácticas de programación.

Esta subárea permite al estudiantado enfrentar los desafíos del desarrollo de software moderno con pensamiento crítico, responsabilidad social y liderazgo técnico, orientado a la calidad, la mejora continua y la transformación digital sostenible.



Tabla 24

Distribución de unidades de estudio de la subárea Desarrollo de Componentes de Software

Unidades de estudio	Nº semanas	Nº horas anuales
1 Desarrollo ágil de software	9	72
2 Calidad de software	9	72
3 Programación orientada a objetos	13	104
4 Estructura de datos	9	72
Total	40	320



Tabla 25

Información administrativa

Carrera técnica: Desarrollo de Aplicaciones de Software	Campo detallado: 0619 Tecnologías de la información y la comunicación (TIC)
Subárea: Desarrollo de componentes de software	Nivel: Undécimo
Unidad de estudio: Desarrollo ágil de software	Tiempo estimado: 72 horas
Competencias para el desarrollo humano: Creatividad	Eje política educativa: Educación para el desarrollo sostenible

Tabla 26

Planificación curricular de la unidad de estudio

Resultados de aprendizaje	Saberes esenciales	Indicador de logro
1. Explicar la importancia del desarrollo ágil de software para proyectos de desarrollo de aplicaciones.	<ul style="list-style-type: none">• ¿Qué es la agilidad?• La agilidad y el costo del cambio• ¿Qué es un proceso ágil?• Principios de agilidad• La política del desarrollo ágil• Otros modelos ágiles de proceso	<ul style="list-style-type: none">• Reconoce qué es la agilidad en proyectos de desarrollo de aplicaciones de software.• Diferencia principios y metodologías ágiles para el desarrollo de aplicaciones de software.• Explica la importancia del desarrollo ágil de software para



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Desarrollo adaptativo de software (DAS)• Scrum• Método de desarrollo de sistemas dinámicos (MDSD)• Desarrollo impulsado por las características (DIC)• Desarrollo esbelto de software (DES)• Modelado ágil (MA)• El proceso unificado ágil (PUA)• Conjunto de herramientas para el proceso ágil• Metodologías ágiles para desarrollo de aplicaciones web avanzadas<ul style="list-style-type: none">• Método Kanban• Scrum y Scrumban• DA Disciplined Agile• Lean• Metodologías híbridas<ul style="list-style-type: none">• Comparación de metodologías ágiles	proyectos de desarrollo de aplicaciones de software.
2. Determinar el uso de herramientas para las metodologías ágiles en	<ul style="list-style-type: none">• ¿Qué es scrum?• Componentes de scrum• Equipo scrum y sus roles (Scrum Master, Product Owner, equipo	<ul style="list-style-type: none">• Reconoce qué es sprint, scrum, el equipo scrum y sus roles en proyectos de desarrollo de aplicaciones de software.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
proyectos para el desarrollo de aplicaciones de software.	<ul style="list-style-type: none">• de desarrollo)• ¿Qué es el backlog, las épicas y las historias de usuario?• ¿Cómo crear historias de usuario?• ¿Cómo estimar historias de usuario?• ¿Qué es un sprint?• ¿Qué es el sprint planning?• Daily - Seguimiento de proyecto• ¿Qué es el sprint review?• Retrospectiva del sprint• Herramientas utilizadas en metodologías ágiles para el seguimiento de proyectos<ul style="list-style-type: none">• Trello• Jira• Notion• Clickup• Slack (medio de comunicación en equipo ágil)• Miro• Smartsheet• Inicio de un proyecto ágil	<ul style="list-style-type: none">• Distingue las herramientas utilizadas en metodologías para proyectos de seguimiento en desarrollo de aplicaciones de software.• Explica el uso de herramientas para las metodologías ágiles en proyectos para desarrollo de aplicaciones de software.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Proceso de planificación ágil• Recopilación de requisitos e historias de usuarios <p>Establecimiento del alcance del proyecto mediante métodos ágiles.</p>	
3. Emplear el uso de GIT y GitHub para el seguimiento, versionamiento para proyectos de desarrollo de aplicaciones de software.	<ul style="list-style-type: none">• ¿Qué es GIT?• GIT y su ecosistema• Instalación de GIT• Editores de código, archivos binarios y texto plano• Introducción a la terminal y línea de comandos• Crear repositorio de git• Uso de la terminal para GIT• Manipulación de archivos y carpetas• Inicialización y configuración• Objetos de GIT• Blobs• Funciones hash• Árboles• Áreas de trabajo• Área staging• Directorio de trabajo• Cierre de objetos con GIT <p>• Operaciones esenciales en GIT</p>	<ul style="list-style-type: none">• Reconoce los alcances de GIT y GitHub en proyectos de desarrollo de aplicaciones de software.• Diferencia las operaciones esenciales en GIT y GitHub• Utiliza operaciones esenciales, ramas, tags, repositorios remotos para proyectos de desarrollo de aplicaciones de software.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Flujo de trabajo y commint• Comandos• Git add• Git commint• Unstaging• Archivo gitignore• Comandos y recursos colaborativos en git y github <ul style="list-style-type: none">• Ramas<ul style="list-style-type: none">• ¿Qué es una rama?• Instalación• Encabezados• Funciones• Gestión de ramas• Fast-forward• Resolución de conflictos• Rebasing• Checkout <ul style="list-style-type: none">• Github<ul style="list-style-type: none">• ¿Qué es github?• Conexiones remotas• Git push• Fetching• Git pull• Git clone• Pull request	



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Tags• ¿Qué es un tag?<ul style="list-style-type: none">• Versionamiento semántico• ¿Cómo funciona?• Etiquetas literales• Etiquetas anotadas• Gestión de etiquetas• Repositorios remotos en github<ul style="list-style-type: none">• ¿Cómo funcionan las llaves públicas y privadas?• Configurar llaves SSH en local• Cambios en github: de master a main• Push• Manejo de ramas en Github• Múltiples colaboradores en un repositorio de github• Pull request• Crear un fork, contribuyendo a un repositorio• Hacer deployment a un servidor• Readme.md	
4. Analizar la importancia de la creatividad en la carrera	<ul style="list-style-type: none">• ¿Qué es la creatividad?• Imaginación, conocimiento y actitud	<ul style="list-style-type: none">• Reconoce el concepto de la creatividad y su importancia



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
técnica de desarrollo de aplicaciones de software.	<ul style="list-style-type: none">• ¿Cómo fomentar la creatividad?• Estrategias para ser más creativos	<p>para el desarrollo de aplicaciones de software.</p> <ul style="list-style-type: none">• Diferencia estrategias para ser más creativo.• Distingue métodos para fomentar la creatividad.
5. Discutir acerca de la importancia de reconocer los problemas socio ambientales globales en la actualidad.	<ul style="list-style-type: none">• Problemáticas socio ambientales globales• Cambio climático<ul style="list-style-type: none">• Causas y consecuencias.• Medidas de mitigación y adaptación.• Pérdida de biodiversidad<ul style="list-style-type: none">• Importancia de la biodiversidad.• Estrategias de conservación.• Contaminación y gestión de residuos<ul style="list-style-type: none">• Tipos de contaminación.• Soluciones y prácticas sostenibles.	<ul style="list-style-type: none">• Reconoce el impacto del cambio climático y la pérdida de la biodiversidad en la actualidad.• Identifica la importancia de la gestión de residuos y los tipos de contaminación de la actualidad.• Analiza la importancia de reconocer los problemas socio ambientales globales en la actualidad.
6. Explicar la importancia del desarrollo ágil de software para proyectos de desarrollo de aplicaciones software.	<ul style="list-style-type: none">• ¿Qué es la agilidad?• La agilidad y el costo del cambio• ¿Qué es un proceso ágil?• Principios de agilidad• La política del desarrollo ágil	<ul style="list-style-type: none">• Reconoce qué es la agilidad en proyectos de desarrollo de software.• Diferencia principios y metodologías ágiles para el desarrollo de aplicaciones de software.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Otros modelos ágiles de proceso• Desarrollo adaptativo de software (DAS)• Scrum• Método de desarrollo de sistemas dinámicos (MDSD)• Desarrollo impulsado por las características (DIC)• Desarrollo esbelto de software (DES)• Modelado ágil (MA)• El proceso unificado ágil (PUA)• Conjunto de herramientas para el proceso ágil• Metodologías ágiles para desarrollo de aplicaciones web avanzadas<ul style="list-style-type: none">• Método Kanban• Scrum y Scrumban• DA Disciplined Agile• Lean• Metodologías híbridas<ul style="list-style-type: none">• Comparación de metodologías ágiles	<ul style="list-style-type: none">• Explica la importancia del desarrollo ágil de software para proyectos de aplicaciones.



Tabla 27

Información administrativa

Carrera técnica: Desarrollo de Aplicaciones de Software	Campo detallado: 0619 Tecnologías de la información y la comunicación (TIC)
Subárea: Desarrollo de componentes de software	Nivel: Undécimo
Unidad de estudio: Calidad de software	Tiempo estimado: 72 horas
Competencias para el desarrollo humano: Orientación de servicio al cliente	Eje política educativa: La ciudadanía planetaria con identidad nacional

Tabla 28

Planificación curricular de la unidad de estudio

Resultados de aprendizaje	Saberes esenciales	Indicador de logro
1. Explicar la importancia de la calidad de software en el desarrollo de aplicaciones de software.	<ul style="list-style-type: none">• ¿Qué es calidad?• Calidad del software• Dimensiones de la calidad de Garvin• Factores de la calidad de McCall• Factores de la calidad ISO 25000	<ul style="list-style-type: none">• Reconoce qué es la calidad de software en el desarrollo de aplicaciones.• Identifica las dimensiones y factores de la calidad para el desarrollo de aplicaciones.• Discrimina el dilema de la calidad del software.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Factores de calidad que se persiguen• Transición a un punto de vista cuantitativo• El dilema de la calidad del software• Software "suficientemente bueno"• El costo de la calidad• Riesgos• Negligencia y responsabilidad• Calidad y seguridad• El efecto de las acciones de la administración• Lograr la calidad del software• Métodos de la ingeniería de software• Técnicas de administración de proyectos• Control de calidad<ul style="list-style-type: none">• Aseguramiento de la calidad	<ul style="list-style-type: none">• Explica la importancia de la calidad de software en el desarrollo de aplicaciones.
2. Analizar el impacto de los defectos y el aseguramiento de la calidad en el software para desarrollo de aplicaciones.	<ul style="list-style-type: none">• Efecto de los defectos del software en el costo• Amplificación y eliminación del defecto	<ul style="list-style-type: none">• Identifica el efecto que producen los defectos en el software para el desarrollo de aplicaciones.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Métricas de revisión y su empleo<ul style="list-style-type: none">• Análisis de las métricas• Eficacia del costo de las revisiones• Revisiones: espectro de formalidad• Revisiones informales<ul style="list-style-type: none">• Revisiones técnicas formales• La reunión de revisión• Reporte y registro de la revisión• Lineamientos para la revisión• Revisiones orientadas al muestreo• Elementos de aseguramiento de la calidad del software (ACS)• Tareas, metas y métricas del ACS<ul style="list-style-type: none">• Tareas del ACS• Metas, atributos y métricas• Enfoques formales al ACS• Aseguramiento estadístico de la calidad del software<ul style="list-style-type: none">• Seis Sigma para la ingeniería de software• Confiabilidad del software• Mediciones de la confiabilidad y disponibilidad	<ul style="list-style-type: none">• Diferencia los elementos de aseguramiento de la calidad del software.• Diferencia las normas de calidad ISO asociadas a la calidad del software.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Seguridad del software<ul style="list-style-type: none">• Las normas de calidad ISO 9000• Las normas de calidad ISO/IEC 5055• Las normas de calidad ISO/IEC 12207• Las normas de calidad ISO/IEC 33000• El plan de ACS	
3. Explicar la importancia de las pruebas de software para el desarrollo de aplicaciones.	<ul style="list-style-type: none">• Pruebas de software<ul style="list-style-type: none">• Pruebas de rendimiento• Pruebas de seguridad• Pruebas de carga• Tipos de pruebas: unitarias, de integración, de sistema, de aceptación.• Automatización de pruebas (Selenium, JUnit, TestNG)• Pruebas de seguridad informática• Pruebas de penetración<ul style="list-style-type: none">• Aseguramiento de la Calidad en el ciclo de vida ágil.	<ul style="list-style-type: none">• Reconoce la importancia de las pruebas de software.• Identifica los tipos de pruebas de software.• Diferencia las pruebas de software.
4. Ilustrar la importancia de la atención al detalle en la carrera técnica y para la vida.	<ul style="list-style-type: none">• Atención al detalle<ul style="list-style-type: none">• ¿Qué es la atención al detalle?• Importancia• Beneficios	<ul style="list-style-type: none">• Reconoce la importancia de la atención al detalle.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Habilidades para atención al detalle.• Observación• Concentración• Manejo de distracciones• Organización	<ul style="list-style-type: none">• Identifica la importancia y los beneficios de la atención al detalle.• Diferencia las habilidades para la atención al detalle.
5. Explicar el impacto de la ciudadanía planetaria en el mundo global.	<ul style="list-style-type: none">• Ciudadanía planetaria• Derechos humanos• Derechos ambientales• Responsabilidades individuales y colectivas• Participación comunitaria• Cultura de Paz<ul style="list-style-type: none">• Convivencia y diversidad cultural	<ul style="list-style-type: none">• Describe los derechos y responsabilidades de la ciudadanía planetaria.• Explica el impacto de la ciudadanía planetaria en el mundo global.



Tabla 29

Carrera técnica: Desarrollo de Aplicaciones de Software	Campo detallado: 0619 Tecnologías de la información y la comunicación (TIC)
Subárea: Desarrollo de componentes de software	Nivel: Undécimo
Unidad de estudio: Programación Orientada a Objetos (POO)	Tiempo estimado: 104 horas
Competencias para el desarrollo humano: Inteligencia emocional	Eje política educativa: Educación para el desarrollo sostenible

Información administrativa

Tabla 30

Planificación curricular de la unidad de estudio



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
1. Utilizar la programación orientada a objetos en la resolución básica de problemas de programación.	<ul style="list-style-type: none">• Introducción a los conceptos de programación orientada a objetos• Diferenciar entre la programación de procedimiento y orientada a objetos• Describir una clase como un plano de un objeto• Describir una clase que se utiliza para crear instancias de un objeto• Modelar objetos como una combinación de:<ul style="list-style-type: none">• Propiedades (campos de datos)• Comportamientos (métodos)• Describir las ventajas de las variables<ul style="list-style-type: none">• Identificar cuatro tipos principales de variables: (boolean, int, double, String)• Declarar y asignar valores a variables• Asignar un nombre a las variables según las convenciones	<ul style="list-style-type: none">• Diferencia la programación procedimental y la orientada a objetos.• Distingue variables según el tipo requerido o solicitado.• Utiliza datos numéricos y textuales en la resolución de problemas básicos de programación.• Utiliza entradas por teclado para la resolución de problemas básicos de programación.•



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Datos numéricos<ul style="list-style-type: none">• Diferenciar los tipos de dato entero (byte, short, int, long)• Tipos de dato de coma flotante (float, double)• Manipular y hacer cálculos con datos numéricos• Utilizar los paréntesis y el orden de las operaciones• Datos textuales<ul style="list-style-type: none">• Tipo de dato char• Cadenas• Concatenar cadenas• Secuencias de escape• las sentencias print• Conversión entre tipos de datos• Automática• Convertir variables a otros tipos de dato• Cadenas como valores numéricos• Entrada del teclado<ul style="list-style-type: none">• Describir la entrada del usuario• Crear un JOptionPane para recopilar la entrada del usuario	



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• La entrada de la consola• La entrada de un archivo• Manejo de tokens y los delimitadores	
2. Aplicar el uso de métodos y clases en la creación de tareas y aplicaciones informáticas.	<ul style="list-style-type: none">• ¿Qué es un método?• Estructurar el código dentro de una clase• Instanciar el objeto• Describir las ventajas de los métodos• Utilizar el operador de punto (.) para acceder a los campos y métodos de un objeto• Proporcionar argumentos a un método• Devolver valores desde un método• Declaración import y paquetes• Acceder a una clase mediante su nombre totalmente cualificado• Describir la función de la sentencia import• Utilizar la sentencia import para acceder a una clase en un paquete	<ul style="list-style-type: none">• Reconoce qué es un método y las formas de instanciar que se posee en la programación orientada a objetos.• Describir los alcances que posee la declaración import y sus usos en creación de tareas y aplicaciones informáticas.• Utiliza diferentes clases en la creación de tareas y aplicaciones informáticas para resolver problemas específicos.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Describir la finalidad de un asterisco en una sentencia import• Identificar los paquetes que se importan automáticamente• Clase String<ul style="list-style-type: none">• Métodos de la clase String• Comparar dos objetos String de forma lexicográfica• Subcadena en un objeto String• Clase Random<ul style="list-style-type: none">• Describir la finalidad y los usos de números aleatorios en la programación Java• Identificar los métodos de la clase Random que obtienen números aleatorios• Obtener números aleatorios en un rango de números• Describir la finalidad del valor inicial de los números aleatorios• Clase Math<ul style="list-style-type: none">• Utilizar métodos de la clase Math	



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
3. Aplicar expresiones booleanas, condicionales y ciclos en la resolución de problemas básicos de tareas y aplicaciones informáticas.	<p>Utilizar campos de la clase Math</p> <ul style="list-style-type: none">• Expresiones booleanas y construcciones if/else• Declarar, inicializar y utilizar variables booleanas• Comparar las expresiones booleanas mediante operadores relacionales• Sentencia if<ul style="list-style-type: none">• Construcciones if/else• Comparar cadenas• Ejecución condicional• Operadores lógicos• Construcciones if encadenadas• Sentencia SWITCH<ul style="list-style-type: none">• Estructura de control de cambio• Construcciones IF/ELSE con estructuras de control de cambio• Palabra clave BREAK• Ciclos FOR	<ul style="list-style-type: none">• Reconoce el uso de las expresiones booleanas en la programación orientada a objetos.• Diferencia los tipos de ciclos presentes en la programación orientada a objetos.• Emplea expresiones booleanas, condicionales y ciclos en la resolución de problemas básicos de tareas y aplicaciones informáticas.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Componentes del ciclo for estándar• Crear y utilizar un ciclo for• Técnicas de depuración• Producen los ciclos infinitos• Ciclos WHILE y DO-WHILE• Ciclos WHILE• Ciclos DO-WHILE• Sentencias BREAK y CONTINUE <p>Documentación de Ciclo</p>	
4. Utilizar clases en la creación de tareas y aplicaciones informáticas para la programación orientada a objetos.	<ul style="list-style-type: none">• Creando de una clase• Crear una clase principal• Cree en el IDE• Utilizar condicionales en los métodos• Instanciación de objetos• Consecuencias en la memoria de la instanciación de objetos• Referencias de objetos• La memoria de pila y de montón• Cadenas son objetos especiales• Constructores	<ul style="list-style-type: none">• Reconoce las características de las clases y los métodos.• Diferencia los tipos de constructores de clases. <p>Emplea clases en la creación de tareas y aplicaciones informáticas para la programación orientada a objetos.</p>



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Valores por defecto• Referencia null• Constructor por defecto• Constructor que acepta argumentos• Campos con un constructor• Referencia de objeto• Sobrecarga de métodos• Varios constructores en una clase• Sobrecarga de un método• Interacción de objeto y encapsulación• Modificador privado para definir variables de clase• Los métodos getter• Variables y métodos estáticos• Método estático• Variables estáticas	
5. Utilizar arreglos unidimensionales, manejo de excepciones y depuración en la resolución de problemas básicos de tareas y aplicaciones informáticas.	<ul style="list-style-type: none">• Arreglos unidimensionales• Crear e inicializar arreglos unidimensionales• Modificar un elemento de arreglo• Recorrer un arreglo unidimensional mediante un ciclo	<ul style="list-style-type: none">• Reconoces los arreglos unidimensionales.• Diferencia el manejo de excepciones y la depuración.• Resuelve problemas básicos de tareas y aplicaciones informáticas utilizando arreglos



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Manipular un arreglo con sus métodos• Recorrer un arreglo con iteradores y ciclos for-each• Clases wrapper y el empaquetado automático• Manejo de excepciones<ul style="list-style-type: none">• Finalidad del manejo de excepciones• Excepciones con una construcción try/catch• Excepciones comunes• Conceptos y técnicas de depuración<ul style="list-style-type: none">• Probar y depurar un programa• Los tres tipos de errores• Técnicas de depuración• Sentencias print• Uso del depurador de su IDE• Consejos y técnicas de depuración	unidimensionales, manejo de excepciones y depuración.
6. Explicar la importancia de la inteligencia emocional en la vida diaria y profesional.	<ul style="list-style-type: none">• ¿Qué es la inteligencia emocional?• La importancia en la vida diaria y profesional	<ul style="list-style-type: none">• Reconoce qué es y la importancia de la inteligencia emocional.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Componentes de la inteligencia emocional:<ul style="list-style-type: none">• Autoconciencia• Autorregulación• Motivación• Empatía• Habilidades sociales.	<ul style="list-style-type: none">• Identifica componentes de la inteligencia emocional.• Explica la importancia de la inteligencia emocional en la vida diaria y profesional.•
7. Analizar el impacto del cambio climático en la actualidad y la necesidad de conservar la biodiversidad.	<ul style="list-style-type: none">• Cambio Climático<ul style="list-style-type: none">• Causas del cambio climático• Impactos en el medio ambiente y la sociedad.• Conservación de la Biodiversidad<ul style="list-style-type: none">• ¿Por qué es crucial conservar la biodiversidad?• Amenazas a la biodiversidad y las posibles soluciones.	<ul style="list-style-type: none">• Reconoce el impacto del cambio climático en el mundo y la necesidad de conservar la biodiversidad.• Identifica las causas del cambio climático y amenazas a la biodiversidad.• Explica el impacto del cambio climático en la actualidad y la necesidad de conservar la biodiversidad.



Tabla 35

Información administrativa

Carrera técnica: Desarrollo de Aplicaciones de Software	Campo detallado: 0619 Tecnologías de la información y la comunicación (TIC)
Subárea: Desarrollo de componentes de software	Nivel: Undécimo
Unidad de estudio: Estructura de datos	Tiempo estimado: 72 horas
Competencias para el desarrollo humano: Gestión del tiempo	Eje política educativa: Educación para el Desarrollo Sostenible

Tabla 36

Planificación curricular de la unidad de estudio

Resultados de aprendizaje	Saberes esenciales	Indicador de logro
1. Examinar los fundamentos de las estructuras de datos incluyendo su clasificación, eficiencia, rendimiento y escalabilidad.	<ul style="list-style-type: none"> Fundamentos de estructura de datos <ul style="list-style-type: none"> Conceptos Diferencia entre estructura lógica y física Software y su importancia 	<ul style="list-style-type: none"> Reconoce el concepto de estructura de datos. Diferencia estructuras de datos según su tipo, comportamiento y aplicación.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Eficiencia• Rendimiento• Escalabilidad• Estructuras<ul style="list-style-type: none">• Lineales• No lineales• Estáticas• Dinámicas	<ul style="list-style-type: none">• Compara la eficiencia, rendimiento y escalabilidad de estructuras lógicas y físicas.• Identifica las estructuras de datos.
2. Utilizar estructuras tales como: arreglos unidimensionales, multidimensionales y listas enlazadas.	<ul style="list-style-type: none">• Listas lineales y estructuras<ul style="list-style-type: none">• Arreglos<ul style="list-style-type: none">• Unidimensionales• Multidimensionales• Listas enlazadas<ul style="list-style-type: none">• Simples• Dobles• Circulares• Comparación de listas y arreglos<ul style="list-style-type: none">• Inserción• Eliminación• Uso de memoria• Dinámica de crecimiento	<ul style="list-style-type: none">• Diferencia las listas lineales y arreglos.• Compara la inserción, eliminación y dinámica de crecimiento de las estructuras aplicadas.• Selecciona la estructura según el rendimiento, la dinámica de crecimiento y los requerimientos.
3. Diseñar aplicaciones de software que incorporen pilas y colas.	<ul style="list-style-type: none">• Pilas y colas<ul style="list-style-type: none">• Funcionamiento<ul style="list-style-type: none">• LIFO• FIFO	<ul style="list-style-type: none">• Explica el funcionamiento LIFO y FIFO, y los tipos de pilas y colas.• Elabora pilas y colas, según los requerimientos.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Implementación con arreglos y listas• Tipos de colas<ul style="list-style-type: none">• Simples• Circulares• Prioridad• Casos de uso con pilas<ul style="list-style-type: none">• Retroceso• Validación de paréntesis• Reversa de textos• Casos de uso para colas<ul style="list-style-type: none">• Sistemas de turnos• Buffers de impresión• Colas de procesos	
4. Elaborar aplicaciones de software aplicando funciones de recursividad.	<ul style="list-style-type: none">• Recursividad aplicada a estructuras de datos<ul style="list-style-type: none">• Recursividad• Condiciones base• Comparación con iteración• Uso de recursividad<ul style="list-style-type: none">• Cálculo factorial• Cálculo Fibonacci• Recorrido de árboles• Búsqueda• Conteo de estructuras enlazadas• Árboles	<ul style="list-style-type: none">• Explica la recursividad aplicada a estructuras de datos.• Implementa funciones recursivas en cálculos y recorridos de estructuras como árboles y listas.• Aplica funciones de recursividad en el desarrollo de aplicaciones de software.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Términos clave<ul style="list-style-type: none">• Raíz• Nodo• Hijos• Hojas• Altura• Árbol binario• Árbol de búsqueda• Recorridos<ul style="list-style-type: none">• Preorden• Inorden• Postorden• Por niveles	
5. Diseñar aplicaciones de software utilizando estructuras de grafos, algoritmos de recorrido y búsqueda.	<ul style="list-style-type: none">• Grafos<ul style="list-style-type: none">• Dirigido y no dirigido• Ponderado y no ponderado• Representación<ul style="list-style-type: none">• Matriz adyacente• Lista de adyacencia• Algoritmos<ul style="list-style-type: none">• BFS recorrido en anchura• DFS recorrido de profundidad• Algoritmos de búsqueda y ordenamiento<ul style="list-style-type: none">• Búsqueda<ul style="list-style-type: none">• Lineal	<ul style="list-style-type: none">• Utiliza grafos dirigidos o no dirigidos, ponderados o no, mediante matrices o listas de adyacencia.• Implementa algoritmos de recorrido, búsqueda y ordenamiento, según la naturaleza del problema.• Selecciona algoritmos para la solución de problemas complejos en software.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Binaria• Ordenamiento<ul style="list-style-type: none">• Burbuja• Inserción• Selección• Otros y comparación entre ellos	
6. Diseñar aplicaciones de software utilizando estructuras de datos y buenas prácticas, de acuerdo con los requerimientos.	<ul style="list-style-type: none">• Integración de estructuras en proyectos<ul style="list-style-type: none">• Selección de estructuras óptimas• Integración de objetos• POO• Modularidad• Reutilización de componentes• Control de errores• Validación de datos• Eficiencia con ética y trabajo colaborativo<ul style="list-style-type: none">• Buenas prácticas de codificación con estructuras de datos• Impacto del rendimiento algorítmico con experiencia de usuario• Ética en el desarrollo<ul style="list-style-type: none">• Transparencia	<ul style="list-style-type: none">• Implementa estructuras de datos, según los requerimientos del proyecto.• Aplica buenas prácticas de codificación, validación de datos y control de errores.• Utiliza el control de versiones, la modularidad y la reutilización del código en el desarrollo de proyectos colaborativos.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Mantenimiento• Calidad del código• Trabajo colaborativo con control de versiones	
7. Explicar el impacto que posee la gestión del tiempo en el ámbito personal como profesional de las personas.	<ul style="list-style-type: none">• ¿Qué es la gestión del tiempo?• ¿Por qué es importante?• Beneficios<ul style="list-style-type: none">• Mejora de la productividad• educación del estrés• Logro de objetivos personales y profesionales• Metodología SMART para la definición de objetivos<ul style="list-style-type: none">• Específicos• Medibles• Alcanzables• Relevantes• Con tiempo definido	<ul style="list-style-type: none">• Reconoce la importancia de la gestión del tiempo.• Identifica los beneficios de la gestión del tiempo.• Discute sobre el impacto que posee la gestión del tiempo tanto en el ámbito personal como profesional.
8. Argumentar la importancia de la biodiversidad, el uso responsable y sostenible de los recursos naturales en la actualidad.	<ul style="list-style-type: none">• Biodiversidad<ul style="list-style-type: none">• ¿por qué es importante conservar la biodiversidad?• Principales amenazas a la biodiversidad• Posibles soluciones• Recursos naturales	<ul style="list-style-type: none">• Reconoce porqué es importante conservar la biodiversidad.• Identifica las amenazas y soluciones para la biodiversidad.• Distingue los tipos de energías renovables.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Gestión de los recursos naturales de manera sostenible.• Tipos de energías renovables• Importancia	<ul style="list-style-type: none">• Explica la importancia del uso responsable y sostenible de los recursos naturales en la actualidad.



MINISTERIO DE
EDUCACIÓN PÚBLICA

GOBIERNO
DE COSTA RICA

Dirección de Educación
Técnica y Capacidades
Emprendedoras

Subárea Bases de datos para proyectos de software



Subárea: Bases de datos para proyectos de software



Desarrollo de
Aplicaciones
de Software



Nivel

Undécimo



Juega para aprender



"Motívate a la salud digital en tu aprendizaje"





Descripción de la Subárea Bases de datos para proyectos de software

La Subárea de Bases de Datos para proyectos de software tiene como propósito el desarrollo competencias para el diseño, implementación y administración de sistemas de bases de datos, que respalden eficazmente las soluciones de software, promoviendo el análisis ético y estratégico de la información como soporte clave para la toma de decisiones en entornos reales o simulados.

Esta subárea contiene 320 horas anuales y contempla el desarrollo de cuatro unidades de estudio: Introducción a los datos, Diagramas, estructuras y modelos de bases de datos, Bases de datos relacionales y Bases de datos no relacionales (NoSQL). La mediación pedagógica de las unidades de estudio desarrolla en el estudiantado capacidades sobre el ciclo completo del manejo de datos, desde la recolección y modelado conceptual, hasta la implementación física y la gestión eficiente.

En la unidad de estudio Introducción a los datos, aborda los fundamentos teóricos y prácticos sobre la naturaleza, representación y clasificación de los datos, así como su valor en la toma de decisiones informadas. Se promueve el análisis ético, el tratamiento de datos y el uso de herramientas para su visualización e interpretación.

La unidad de estudio Diagramas, estructuras y modelos de bases de datos, fortalece capacidades para el modelado de datos de forma conceptual, lógica y física, aplicando principios de normalización, integridad referencial y diseño orientado a objetos mediante diagramas entidad-relación, relacionales y UML. Se



promueve la documentación técnica que garantice trazabilidad y comprensión de las estructuras de información.

La unidad de Bases de datos relacionales profundiza en el modelo relacional, desarrollando habilidades para diseñar esquemas normalizados, implementar bases de datos con lenguaje SQL, y administrar consultas complejas con enfoque en la integridad, eficiencia y seguridad de la información, utilizando herramientas especializadas de gestión.

Finalmente, Bases de datos no relacionales (NoSQL) introduce al estudiantado en el uso de tecnologías emergentes para el manejo de grandes volúmenes de datos no estructurados o semiestructurados. Se analiza el uso de bases como MongoDB y similares, destacando su aplicabilidad en entornos que requieren escalabilidad, personalización y atención ágil al cliente, integrando una cultura de servicio con sensibilidad, eficiencia y ética profesional.

Esta subárea propicia una comprensión integral y actualizada del papel de las bases de datos en el desarrollo de software, formando personas técnicas capaces de gestionar información de manera responsable, estratégica y centrada en la mejora continua de los servicios tecnológicos.



Tabla 31

Distribución de unidades de estudio de la subárea Bases de datos para proyectos de software

Unidades de estudio	N° semanas	N° horas anuales
1 Introducción a los datos	4	32
2 Diagramas, estructuras y modelos de bases de datos	10	80
3 Bases de datos relacionales	13	104
4 Bases de datos no relacionales NoSQL	13	104
Total	40	320



Tabla 32

Información administrativa

Carrera técnica: Desarrollo de Aplicaciones de Software	Campo detallado: 0619 Tecnologías de la información y la comunicación (TIC)
Subárea: Bases de datos para proyectos de software	Nivel: Undécimo
Unidad de estudio: Introducción a los datos	Tiempo estimado: 32 horas
Competencias para el desarrollo humano: Compromiso ético	Eje política educativa: Fortalecimiento de una ciudadanía planetaria con identidad

Tabla 33

Planificación curricular de la unidad de estudio

Resultados de aprendizaje	Saberes esenciales	Indicador de logro
1. Explicar las características de los datos, usos, tipos y aspectos básicos del análisis de los datos.	<ul style="list-style-type: none">Datos:<ul style="list-style-type: none">Valor de los datos.Datos y datos masivos.Datos abiertos y privados.Datos estructurados y no estructurados.	<ul style="list-style-type: none">Identifica los tipos de datos y el análisis de los datos.Reconoce los tipos de datos mediante la manipulación y análisis de la información.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Datos almacenados y en movimiento.• Administración de datos masivos.• Evolución hacia los datos masivos.• Tecnologías de administración básica de datos.	<ul style="list-style-type: none">• Reconoce el análisis de los datos en la resolución de problemas.
2. Describir características, elementos y conceptos fundamentales asociados con las bases de datos.	<ul style="list-style-type: none">• Aspectos básicos del análisis de datos:<ul style="list-style-type: none">• Definición• Uso de datos masivos.• Tipos de análisis de datos.• Ciclo de vida del análisis de datos.• Fuente y preparación de los datos.• Adquisición de datos y preparación.• Conceptos:<ul style="list-style-type: none">• Datos• Registros• Archivo• Campo.• Fuentes de datos	<ul style="list-style-type: none">• Identifica los conceptos de datos, registros, archivos, campos, las fuentes y tipos de datos.• Ejemplifica los diferentes atributos de los datos.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Tipos de datos• Atributos• Valor de los datos• Sistemas de manejo de datos.• Bases de datos:<ul style="list-style-type: none">• Concepto.• Características.• Usos y aplicaciones.• Aportes al trabajo cotidiano.	<ul style="list-style-type: none">• Distingue los sistemas para el manejo de datos y las bases de datos.
3. Demostrar conductas que reflejen compromiso ético, aplicando principios de gestión de datos responsable.	<ul style="list-style-type: none">• Retos intrínsecos a los datos• Discriminación pasada• Desbalance de clase y validez	<ul style="list-style-type: none">• Identifica los retos intrínsecos a los datos.• Explica la discriminación pasada.
4. Ejercer acciones cotidianas que contribuyen a su bienestar y el de los demás.	<ul style="list-style-type: none">• Promoción de acciones para el cuidado personal• Medidas a tomar para una buena nutrición en su entorno.• Práctica de hábitos de vida saludables, mejorando la interacción social y ambiental	<ul style="list-style-type: none">• Explica el desbalance de clases y validez.• Identifica acciones para el mejoramiento de cuidado personal y nutrición.• Reconoce la importancia de hábitos de vida saludables.• Implementa hábitos de vida saludable que propicien mejores condiciones desde el punto de vista social, económico y ambiental.



Tabla 35

Información administrativa

Carrera técnica: Desarrollo de Aplicaciones de Software	Campo detallado: 0619 Tecnologías de la información y la comunicación (TIC)
Subárea: Bases de datos para proyectos de software	Nivel: Undécimo
Unidad de estudio: Diagramas, estructuras y modelos de bases de datos	Tiempo estimado: 80 horas
Competencias para el desarrollo humano: Compromiso ético	Eje política educativa: Educación para el desarrollo sostenible

Tabla 36

Planificación curricular de la unidad de estudio

Resultados de aprendizaje	Saberes esenciales	Indicador de logro
1. Emplear los elementos que forman parte del modelado de datos en las bases de datos	<ul style="list-style-type: none">• Modelado de datos• Esquemas• Instancias	<ul style="list-style-type: none">• Identifica las categorías de modelos de datos, tipos de relaciones e independencia de los datos.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Categorías de los modelos de datos<ul style="list-style-type: none">• Alto nivel o conceptuales• Bajo nivel o físicos• Representativo o implementación• Arquitectura de tres esquemas<ul style="list-style-type: none">• Nivel interno• Nivel conceptual• Nivel de vista o externo• Independencia de datos<ul style="list-style-type: none">• Lógica• Física• Estructura de datos relaciones<ul style="list-style-type: none">• Relaciones• Propiedades• Características• Tipos de relaciones<ul style="list-style-type: none">• Relaciones base• Vistas• Atributos• Tuplas	<ul style="list-style-type: none">• Reconoce esquemas, instancias, atributos, claves, tuplas, relaciones y sus características como elementos del modelado de datos.• Aplica las reglas de integridad en el modelado de datos.• Explica la arquitectura de tres esquemas para el modelado de datos.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Claves	
	<ul style="list-style-type: none">• Reglas de integridad<ul style="list-style-type: none">• Nulos• Integridad de identidades• Integridad referencial	
2. Contrastar los elementos, el diseño y características, implementados en el diseño de bases de datos.	<ul style="list-style-type: none">• Reglas de negocio• Elementos del diseño de bases de datos<ul style="list-style-type: none">• Recolección y análisis de requisitos• Sistema de gestión de base de datos• Diseño de aplicación• Prototipado• Implementación• Pruebas• Mantenimiento• Diseño de bases de datos<ul style="list-style-type: none">• Diseño conceptual<ul style="list-style-type: none">• Entidades• Relaciones• Atributos• Dominios• Jerarquías	<ul style="list-style-type: none">• Identifica los elementos que conforman los diseños de bases de datos conceptual, lógico y físico.• Reconoce los elementos involucrados en el diseño de bases de datos.• Realiza diagramas de clase UML para el diseño de bases de datos.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Diseño lógico<ul style="list-style-type: none">• Entidades fuertes• Entidades débiles• Relaciones binarias• Jerarquías• Normalización• Restricciones de integridad• Diseño físico<ul style="list-style-type: none">• Esquema lógico• Representación física• Mecanismos de seguridad	
3. Implementar las formas normales utilizadas en el diseño de bases de datos.	<ul style="list-style-type: none">• Diagramas de clase UML para diseños de bases de datos• Dependencias funcionales<ul style="list-style-type: none">• Definición• Reglas de inferencia• Equivalencia de conjuntos• Conjuntos mínimos• Normalización<ul style="list-style-type: none">• Formas normales basadas en claves principales• Normalización de relaciones<ul style="list-style-type: none">• Primera forma normal	<ul style="list-style-type: none">• Reconoce las dependencias funcionales y las formas normales basadas en las claves principales.• Describe las formas normales utilizadas en el diseño de bases de datos.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Segunda forma normal• Tercera forma normal	<ul style="list-style-type: none">• Aplica las formas normales utilizadas en el diseño de bases de datos.
4. Demostrar conductas que reflejen compromiso ético, en la gestión de datos.	<ul style="list-style-type: none">• Forma normal de Boyce-Codd• La ética de los datos: ¿qué es?• ¿Por qué es importante la ética de los datos?• La ética de los datos y la gobernanza de datos	<ul style="list-style-type: none">• Reconoce qué es la ética en la gestión de datos.• Describe la importancia de la ética en los datos.
5. Argumentar sobre el concepto, importancia y buenas prácticas de desarrollo sostenible en procesos productivos relacionados con el sector de TI.	<ul style="list-style-type: none">• Desarrollo sostenible:<ul style="list-style-type: none">• Concepto y elementos• Objetivos para el Desarrollo Sostenible (ODS) según la UNESCO• Impacto ambiental y social.• Causas y consecuencias.• Labores técnicas sostenibles.	<ul style="list-style-type: none">• Explica la ética y gobernanza de los datos.• Reconoce el concepto de Desarrollo sostenible y sus elementos.• Distingue los 17 ODS según la UNESCO.• Ejemplifica efectos al ambiente y la salud producto de procesos productivos del sector de tecnologías de información (TI).



Tabla 37

Información administrativa

Carrera técnica: Desarrollo de Aplicaciones de Software	Campo detallado: 0619 Tecnologías de la información y la comunicación (TIC)
Subárea: Bases de datos para proyectos de software	Nivel: Undécimo
Unidad de estudio: Bases de datos relacionales	Tiempo estimado: 104 horas
Competencias para el desarrollo humano: Compromiso ético	Eje política educativa: La ciudadanía digital con equidad social

Tabla 38

Planificación curricular de la unidad de estudio

Resultados de aprendizaje	Saberes esenciales	Indicador de logro
1. Explicar los lenguajes relacionales utilizados en la creación de bases de datos relacionales.	<ul style="list-style-type: none">• Lenguajes relacionales<ul style="list-style-type: none">• Manejo de datos• Álgebra relacional• Cálculo relacional• Orientado a tuplas• Orientando a dominios	<ul style="list-style-type: none">• Identifica los lenguajes relacionales y el manejo de los datos.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">Otros lenguajes relacionales	<ul style="list-style-type: none">Reconoce tipos de lenguajes relacionales para bases de datos
2. Crear bases de datos utilizando el lenguaje relacional SQL.	<ul style="list-style-type: none">Lenguaje SQL<ul style="list-style-type: none">Esquemas y catálogosCreación de tablasTipos de datosInserción de datosConsulta de datosActualización y eliminación de datosEstructuras de la sentencia SelectExpresiones Select, From, WhereNulosTipos de datos	<ul style="list-style-type: none">Aplica álgebra y cálculo relacional en bases de datos relacionalesIdentifica estructuras de sentencia, expresiones y funciones del lenguaje relacional SQL.Reconoce los tipos de datos, operadores y consultas del lenguaje relacional SQL.Distingue las sentencias Insert, Delete, Update del lenguaje relacional SQL.Aplica funciones y consultas del lenguaje relacional SQL en bases de datos relacionales.Aplica las cláusulas Group By y Having en bases de datos relacionales.



Resultados de aprendizaje

Saberes esenciales

Indicador de logro

- Operadores y funciones
 - Operadores lógicos
 - Operadores de comparación
 - Operadores matemáticos
 - Operadores y funciones de cadena de caracteres
 - Operadores y funciones de fechas
 - Función Case
 - Funciones COALESCE y NULLIF
 - Funciones Exists y Unique
 - Funciones agregadas
 - Consultas en SQL
 - Orden de resultados
 - Comparación con valores NULL
 - Consultas anidadas, tuplas y comparaciones
 - Consultas anidadas correlacionas
 - Tablas concatenadas en SQL
 - Agrupamiento
- Aplica las sentencias Insert, Delete, Update en bases de datos relacionales.
 - Emplea transacciones, restricciones, triggers y excepciones en bases de datos relacionales.
 - Desarrolla bases de datos usando el lenguaje relacional SQL.



Resultados de aprendizaje

Saberes esenciales

Indicador de logro

- | | | |
|--|--|---|
| | <ul style="list-style-type: none">• Cláusulas Group By y Having• Sentencias Insert, Delete, Update de SQL• Restricciones, triggers y excepciones en SQL• Transacciones en SQL | |
| 3. Demostrar conductas que reflejen compromiso ético, aplicando principios de la gestión de datos. | <ul style="list-style-type: none">• Formularios para SQL (Forms)• ¿Qué es un marco ético para el uso de los datos?• ¿Qué proyectos e iniciativas exige la ética de los datos?• ¿Cómo puede ayudar el almacenamiento a mejorar las mejores prácticas de la ética de los datos? | <ul style="list-style-type: none">• Identifica el marco ético para el uso de datos.• Reconoce proyectos o iniciativas que exigen el uso de la ética de los datos.• Explica cómo puede ayudar el almacenamiento de datos a |



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
4. Utilizar las tecnologías digitales como recurso para dinamizar el aprendizaje dentro de su vida profesional y cotidiana.	<ul style="list-style-type: none">• Tecnologías digitales:<ul style="list-style-type: none">• Uso• Importancia en el proceso de aprendizaje.• Impacto económico y social.	<p>mejorar prácticas en el uso de datos.</p> <ul style="list-style-type: none">• Identifica las tecnologías digitales como recurso para procesos de aprendizaje.• Reconoce la importancia de procesos de aprendizaje y su impacto económico y social



Tabla 34

Información administrativa

Carrera técnica: Desarrollo de Aplicaciones de Software	Campo detallado: 0619 Tecnologías de la información y la comunicación (TIC)
Subárea: Bases de datos para proyectos de software	Nivel: Undécimo
Unidad de estudio: Bases de datos no relacionales NoSQL	Tiempo estimado: 104 horas
Competencias para el desarrollo humano: Orientación de servicio al cliente	Eje política educativa: La ciudadanía digital con equidad social

Tabla 35

Planificación curricular de la unidad de estudio

Resultados de aprendizaje	Saberes esenciales	Indicador de logro
1. Examinar los conceptos básicos sobre bases de datos NoSQL.	<ul style="list-style-type: none">• Qué es NoSQL• Bases de datos NoSQL	<ul style="list-style-type: none">• Identifica los principales motores de bases de datos NoSQL



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Principales motores de bases de datos NoSQL• Qué puede hacer<ul style="list-style-type: none">• Categorías• Características• NoSQL frente a sistemas de gestión de bases de datos relacionales (RDBMS)• Qué es JSON<ul style="list-style-type: none">• Sintaxis• Crear objetos JSON• Qué es BSON<ul style="list-style-type: none">• Tipos de BSON• Conversores JSON a BSON• Bases de datos colecciones y documentos en NoSQL• Bases de datos orientadas a grafos<ul style="list-style-type: none">• Conceptos básicos de grafos	<ul style="list-style-type: none">• Reconoce qué son bases de datos, categorías, características, colecciones y documentos en NoSQL.• Diferencia entre NoSQL y sistemas de gestión de bases de datos relacionales• Crea objetos JSON y conversores de JSON a BSON.• Resuelve problemas específicos utilizando objetos JSON y conversores de JSON a BSON.• Utiliza bases de datos orientadas a grafos.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Operaciones básicas sobre grafos• Respuesta en volumen de los algoritmos sobre grafos. Escalabilidad• Limitaciones de las bases de datos relacionales	
2. Emplear los tipos de almacenamiento utilizados en bases de datos NoSQL, para la resolución de problemas.	<ul style="list-style-type: none">• Bases de datos orientadas a grafo• Almacenamiento en base de datos NoSQL• Almacenamiento basado en Clave Valor<ul style="list-style-type: none">• Ordenación• Conexiones• Comandos que operan sobre toda clase de valores• Comandos que operan con cadenas• Comandos que operan en hashes• Comandos de operación en listas	<ul style="list-style-type: none">• Reconoce los tipos de almacenamiento presentes en bases de datos NoSQL.• Distingue los comandos que operan en el almacenamiento basado en clave valor.• Emplea el almacenamiento basado en clave valor, utilizando software especializado para bases de datos NoSQL.• Utiliza el almacenamiento basado en columnas mediante



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Comandos que operan sobre conjuntos y conjuntos ordenados• Transacciones• Almacenamiento basado en columnas• Modelado del esquema• Arquitectura• MapReduce• Almacenamiento basado en documentos• Conexiones a la base datos• Autenticación• Obtener la lista de colecciones• Obtener una colección• El uso de un cursor para obtener todos los documentos• Obtención de un documento con una consulta• Obtención de un conjunto de documentos con una consulta• Creación de un índice• Obtener una lista de índices en una colección• Funciones administrativas• Obtener una lista de bases de datos	<p>software especializado para bases de datos NoSQL</p> <ul style="list-style-type: none">• Utiliza el almacenamiento basado en documentos haciendo uso de software especializado para bases de datos NoSQL.• Resuelve problemas específicos utilizando tipos de almacenamiento presentes en bases de datos NoSQL.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
3. Aplicar conceptos, comandos y operadores esenciales en la implementación de bases de datos NoSQL.	<ul style="list-style-type: none">• Proceso de instalación de base de datos NoSQL<ul style="list-style-type: none">• Modificar variables de entorno• Crear directorio de almacenamiento• Iniciar y detenerte el motor de base de datos• Tipos de datos• Instalación de editor de texto• Modelado de datos con base de datos NoSQL• Crear base de datos• Crear colecciones• Crear documentos• Crear, leer, actualizar y borrar (CRUD)• Insert(), insertOne(), insertMany(), remove()• Insert para array de Objetos• Cursor• Iterando sobre los documentos<ul style="list-style-type: none">• Batch Size• Métodos de uso para cursor• Consultas (Query)• Creando colecciones	<ul style="list-style-type: none">• Describe el proceso de instalación de bases de datos NoSQL.• Identifica los tipos y modelado de datos en bases de datos NoSQL.• Crea bases de datos, colecciones, documentos en bases de datos NoSQL, utilizando software especializado.• Emplea CRUD (crear, leer, actualizar y borrar), en la solución de problemas específicos.• Aplica Insert(), insertOne(), insertMany(), remove(), Insert para array de Objetos, en la solución de problemas específicos.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Búsqueda fin()• Búsqueda simple• Búsquedas avanzadas	<ul style="list-style-type: none">• Utiliza los métodos de uso del curso, en la solución de problemas específicos.
	<ul style="list-style-type: none">• Operadores<ul style="list-style-type: none">• Relacionales• Elementos• Cadenas (strings)• Lógicos• Sobre arrays	<ul style="list-style-type: none">• Efectúa consultas (query), operadores en la solución de problemas específicos.
	<ul style="list-style-type: none">• Operadores de actualización<ul style="list-style-type: none">• Operadores de modificación• Primeros documentos• Documento no existente• Documento con valor actual• Opción múltiple• Operadores para fechas• Operadores para campos• Operador posicional dentro de arreglos• Operador posicional en objetos dentro de arreglos• Operadores para borrador documentos, colecciones y bases de datos	<ul style="list-style-type: none">• Aplica operadores en la solución de problemas específicos.• Emplea operadores de actualización en la solución de problemas específicos.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
4. Implementar acciones orientadas a la resolución de problemas en situaciones propias del área técnica y de la vida cotidiana.	<ul style="list-style-type: none">• Solución de problemas:<ul style="list-style-type: none">• Concepto.• Actitud hacia los problemas.• Generación de soluciones alternativas• Procesos para la solución de problemas.	<ul style="list-style-type: none">• Identifica situaciones que pueden entenderse como problema en el ámbito de su área de formación técnica.• Interpreta procesos para la solución de problemas.• Genera oportunidades y alternativas que brinden solución a los problemas identificados en el ámbito de su área de formación técnica.
5. Utilizar las tecnologías digitales como insumo para el aprendizaje en el desarrollo de su campo de formación técnica.	<ul style="list-style-type: none">• Tecnologías digitales<ul style="list-style-type: none">• Utilización• Relevancia en el proceso de aprendizaje• Impacto del uso en la actualidad	<ul style="list-style-type: none">• Identifica el uso de las tecnologías digitales en las bases de datos.• Explica la relevancia de la utilización de las bases de datos.• Valora el impacto de uso de las tecnologías digitales en la actualidad.



MINISTERIO DE
EDUCACIÓN PÚBLICA

GOBIERNO
DE COSTA RICA

Dirección de Educación
Técnica y Capacidades
Emprendedoras

Subject Area English Oriented to Software Applications Development





Description

The Higher Education Board has approved a subject area for acquiring language skills in English for Specific Purposes. This significant addition to the curriculum of Technical Vocational Education and Training (TVET) is designed to enhance our students' employability and shape their future career prospects, thereby contributing to the overall improvement of the country's competitiveness.

The development of language skills in English is essential for Costa Rican youth to successfully integrate into society, take advantage of new opportunities, and enhance their employability.

The subject area **English Oriented to Software Applications Development** in **Eleventh grade** offers a truly innovative curricular approach. It combines the development of communicative skills with student-centered pedagogy, a technical orientation that integrates collaborative learning, the development of critical thinking, instruction based on a conversation about a problem or product in the classroom, and project-based learning. This groundbreaking approach is set to revolutionize the way we teach and learn English in the context of technical education.

For the first time, English for Specific Purposes (ESP) is incorporated into this program. In this program, the four linguistic competencies are worked on using the six levels of the Common European Framework of Reference (CEFR), with essential knowledge that belongs to the Cloud Computing field and related topics such as employability and entrepreneurship.



At the end of the Tenth grade, the student will become an English Independent User (B1.1) according to the Common European Framework of Reference (CEFR).

The subject area contains four scenarios, each with several themes, detailed in the Curricular Grid and the Curriculum Scope and Sequence, detailed later in this section.

The organization outlined in this Curriculum is closer to real-life language use, grounded in interaction in which meaning is co-constructed. The goals are presented under four modes of communication: reception, production, interaction, and mediation. (CEFF, 2019 p.30.)

Language, embracing language learning, comprises the action performed by people who develop a range of general and communicative language competencies as individuals and social agents. They draw on the competencies at their disposal in various contexts under various conditions and under different constraints to engage in language activities involving language processes to produce and receive texts concerning themes in specific domains, activating those strategies that seem most appropriate for accomplishing the tasks. Monitoring these actions by the participants leads to reinforcing and modifying their competencies.

The CEFR has two axes: a horizontal axis for describing different activities and aspects of competence and a vertical axis representing progress in proficiency. To facilitate organization, the CEFR presents six expected reference levels. Firstly, they can be grouped into three broad categories: Basic user (A1 and A2), Independent user (B1 and B2), and Proficient User (C1 and C2). Secondly, the six reference levels are often segmented.



Illustration 1

Expected Reference Levels in the Professional Technical Education Curriculum

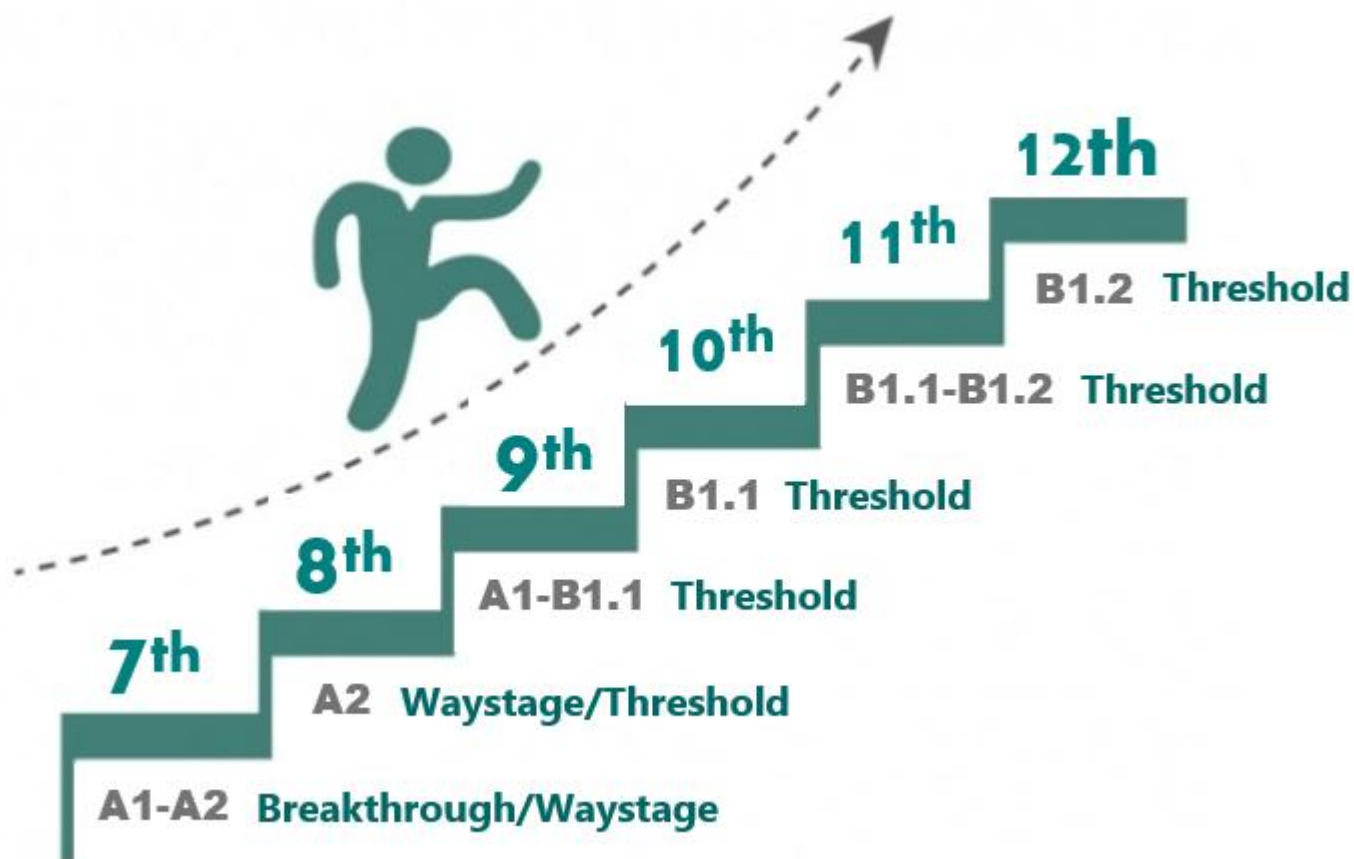




Table 1

Range of hours required to achieve category.

Category	Range of hours required to achieve the category
A1	Approximately 90-100
A2	Approximately 180-200
B1	Approximately 350- 400
B2	Approximately 500-600
C1	Approximately 700-800
C2	Approximately 1000 –1200

Source: Prepared by the authors based on data supplied by CEFR, 2014.



Rationale

The education system is based on the Constitution of Costa Rica (1949), which states that “the State is obliged to provide adequate education conforming to the needs and requirements of students, to allow them the greatest development of their abilities, and determining education as a fundamental right” (Article 77 and 78).

In Costa Rica, education is viewed as a human and constitutional right. The education system seeks knowledge, abilities, skills, values, and attitudes to foster students' comprehensive development and active participation in civil society and the country's economic life.

As part of its constitutional mandate, the High Education Board (CSE) has approved several significant provisions, regulations, and policies to guide Costa Rican education. The curricular policy document "Educating for a New Citizenship" and the educational policy document "The person: center of the educational process and transforming subject of society" are essential.

In compliance with the provisions of the regulations and policies approved by the High Education Council, The Bureau of Technical Education and Entrepreneurship (DETCE) has implemented several educational reforms aimed at providing tools to promote the incorporation of people into employability, the creation of their enterprise, or pursuing higher education studies.



Seeking ongoing improvement and promoting upward social mobility of the Costa Rican population, technical vocational education (ETP) in Costa Rica continues to evolve to generate qualified, technical human talent capable of making informed decisions, taking responsibility for their actions, and influencing current and future communities. All this must be coupled with environmental integrity, economic viability, social justice with cultural diversity respect, and environmental ethics to contribute to the country's competitiveness.

The educational and curricular policies approved by the CSE establish the educational model framework for the ETP curriculum, which is focused on competency-based education. This constitutes the foundation and the frame of reference to follow for the achievement of the proposed goals and objectives of the subsystem.

The study programs are based on the philosophical pillars established in the Educational Policy: The person, the center of the educational process, and the transforming subject of society.

Paradigm of Complexity

It claims that the human being is self-organized and self-referential, aware of himself and his environment, and whose existence makes sense within a social-family natural ecosystem and as part of society. Regarding knowledge acquisition, this paradigm considers that students develop a bio-natural ecosystem (which refers to the biological nature of knowledge in terms of brain forms and learning modes) and a social ecosystem that conditions knowledge acquisition. Human beings are characterized by autonomy and individuality, establishing relationships with the environment by having skills for learning, inventiveness, creativity, the ability to integrate information from the natural and social world, and the capacity to make decisions.



In the educational field, the paradigm of complexity allows for broadening the training horizon. It considers that human action, due to its characteristics, is uncertain and full of unpredictable events that require students to develop their inventiveness and propose new strategies to address a reality that changes daily.

Humanism

It is aimed at personal growth and, therefore, appreciates students' experience, including their emotional aspects. Every person is considered responsible for their own life and self-realization. Consequently, education focuses on the individual so that they evaluate and guide their own experience through the meaning acquired by their learning process.

Every person is unique and different, with initiative, personal needs to grow, with the potential to develop activities and solve problems creatively.

Social Constructivism

It proposes the maximum and multifaceted development of the student's capacities and interests according to learning in the social context, considering their prior experiences and the mental structures of the person participating in the processes of knowledge construction. It is both a part and a product of human activity in the social and cultural context where the person develops.

Rationalism



It is based on reason and objective truths as the principles for building valid knowledge; it has been essential in conceptualizing Costa Rican educational policies (CSE; MEP, 2016, pp. 8-10).

Principles and axes that permeate education policy

Study programs aim to develop specific skills and competencies for human growth based on the philosophical pillars of educational policy and articulated with the axes permeating different situations in the educational field. These axes are part of the actions implemented in this curriculum across all the themes to be developed.

Education for Sustainable Development

This axis turns education into a tool to empower people to make informed decisions and take responsibility for their actions and their impact on present and future communities. This contributes to developing societies with environmental integrity, economic viability, and social justice for present and future generations.

Global Citizenship with National Identity

This axis aims to strengthen awareness of the connection and immediate interaction between people and the environment worldwide and the influence of local actions on the global sphere and vice versa. In addition, it implies regaining our historical memory to be aware of who we are, where we come from, and where we want to go.



Digital Citizenship with Social Equity

This axis seeks the development of several practices aimed at reducing the social and digital gap through the use and exploitation of digital technologies (CSE; MEP, 2016, p 10-12).

From the perspective of a competence-focused education, the four scopes promoted by Curriculum Transformation are integrated: Educating for a new citizenship (2015):

- Ways of thinking: It refers to the cognitive development of each person, which implies those skills related to the generation of knowledge, problem-solving, creativity, and innovation.
- Ways of living in the world: It entails sociocultural development, the interrelationships woven within global citizenship with multicultural roots, and the construction of life projects.
- Ways of relating to others: This is associated with developing bridges built through communication and collaboration.
- Tools to integrate into the world: These refer to the adoption of digital technologies and other integration forms and the attention that must be paid to information management (MEP, 2015, p 33-37).

Due to technological, social, economic, and environmental changes, it is necessary to develop specific and generic competencies for human development, which would allow students to join the workforce successfully or to start their entrepreneurial initiative in their technical careers. These competencies will help to continue learning throughout life, for innovation and creativity in individual and teamwork, critical thinking, problem-solving with social responsibility, environmental awareness, and ethical commitment.



In this sense, the term "localized" communities is considered, which implies that individuals or groups are capable of "thinking globally and acting locally." Therefore, it incorporates the need to learn to live together and recognize the collective power of citizen action. **English Oriented to Software Applications Development** curriculum presents the goals under four modes of communication: reception, production, interaction, and mediation, using the standard reference levels established by the Common European Framework of Reference for languages.



Common European Framework of Reference for Languages

The Common European Framework of Reference for Languages: learning, teaching, assessment (CEFR) is a guideline used to describe the achievements of foreign language learners. This guideline contains standards for grading an individual's language proficiency. The Council of Europe established it between 1989 and 1996 as part of the "Language Learning for European Citizenship" project. The main objective of this guideline is to provide a teaching, learning, and assessing method that applies to all languages in Europe.

The CEFR has three principal dimensions: language activities, the domains in which the language activities occur, and the competencies on which we draw when we engage in them.

Language Activities

The CEFR distinguishes between four kinds of language activities:

- Reception (listening and reading),
- Production (spoken and written),
- Interaction (spoken and written),
- Mediation (translating and interpreting).



Domains

General and communicative competencies are developed by producing or receiving texts in various contexts under various conditions and constraints. These contexts correspond to multiple sectors of social life that the CEFR refers to as domains. Four broad domains are then distinguished: educational, occupational, public, and personal.

Competences

The Common European Framework of Reference for Languages: learning, teaching, assessment presents a comprehensive descriptive scheme of language proficiency and a set of standard reference levels (A1, A2, B1, B2, C1, C2) defined in illustrative descriptor scales, plus options for curriculum design promoting plurilingual and intercultural education. One of the main principles of the CEFR is promoting the positive formulation of educational aims and outcomes at all levels.



General Mediation Strategies and Pedagogical Approach

The Action Oriented Approach

This curriculum adopts the action-oriented approach to make language learning/teaching more efficient. It emphasizes what learners know and do to communicate successfully by completing tasks (not exclusively language-related) in each set of circumstances, in a specific environment, and within a particular field of action. It uses general and specific competencies in meaningful contexts and real-life scenarios to use the language.

There is a progressive shift from complementing and improving the missing aspects of the Communicative Approach to the Action-Oriented Approach; increasing communication among people from various countries increases the need for foreign language learning and the methods, approaches, and techniques.

The action-oriented approach, which does not ignore the social and cultural nature of the language or its communicative nature, deals with a new social dimension. It calls the learners “social actors” (CEFR., 2000, p. 9), creating a common point in acquiring skills and learning knowledge. “Actor means a person performing and animating some duties. Since foreign language is learned through some duties and actions as well, it handles the learners as (social) people who should perform tasks” (Delibaş, 2013, p. 1). Learners/users are responsible for learning in this approach, where the social dimension is first mentioned in language teaching. “This social



dimension is to prepare the learners not only to live together but also to work with strangers in their own country or a foreign country with different cultures and spoken languages.

The need to use the language that emerged while fulfilling the tasks makes the learning process effective and the learner active. Puren expresses the importance of actions in communication by saying, "This is an action that determines communication"(2006, p. 38). Bourguignon supported this opinion by adding, "There is no point in establishing communication on its own. But it becomes meaningful when it mediates actions" (2006, p. 69).

The action-oriented approach considers the learner a social agent. Learning occurs in a social learning environment, and linguistic, pragmatic, and communicative skills are developed. Creating a social language environment where the learners can communicate with each other in the middle of the pluricultural and plurilingual environment depends on teachers' skills and knowledge. The tasks in or out of the classroom must be parallel to the needs of the learners, and the teachers must make learners feel these needs. If considered, language learning is divided into knowledge and skills.

The action-oriented approach is the name of these two processes from constructive learning, in which the learner is autonomous and directs his process. Knowledge is constructed during the process, and skills are acquired commonly and internationally.

Krashen explains this feature of language acquisition by saying, "Language acquisition is a subconscious process; language acquirers are not usually aware of the fact that they are acquiring language but are only aware of the fact that they are using the language for communication (2009, p. 10). He also makes clear the



difference between learning and using a language. In this acquisition and learning process, "language is not only a means of communication but a tool of social action at the same time" (Alrabadi, 2012, p. 1).

Bourguignon also emphasizes the same characteristic: "In an action-oriented approach, communication is at the action service" (2006, p. 64). It shouldn't be forgotten that "the action came before the language in the process of the evolution of humanity, and it constitutes the first stage of the interaction between the people; first, the action is revealed, then the language develops" (Moreno; Dökme; as cited in Sayinsoy, 2003, p. 116). This phrase shows the learner and the teacher how necessary the action is.

They summarize the components of the action-oriented approach. The social agent who learns in a learning environment uses various knowledge, skills, and abilities when performing tasks. Every place where language learning is considered a social process occurs is the social learning environment; therefore, this social environment can be a classroom, home, or shopping center. The learner is an autonomous language user in this social environment but a collaborator as a social agent. It shouldn't be forgotten that this approach is based on the tasks. Essential tools to create meaningful experiences are authentic materials, comprehensible input, as much as possible, and IT access. Functions, vocabulary, grammar, and phonology are taught to facilitate communication. This approach also considers cognitive and emotional resources.



Task-Based Language Teaching (TBLT)

What is a Task? It is the purposeful actions performed by one or more individuals strategically using their specific competencies to achieve a given result. When the description of the text (oral and written) is scrutinized, it reveals that language learners face tasks in everyday life within domains and scenarios. To fulfill these tasks, the learner will need several bits of knowledge, skills, and abilities. The learner is not speaking or writing to another person but speaking or writing in a real-life context for a social purpose.

The task stimulates the learners' commitment to the learning process. It may differ according to the balance determined by the goal and the combination of dimensions (general and communicative competencies). There are different types of task orientations to complexity (from simple to complex), length (from shortest to longest), and social implication (from individual actions to collective actions).

Task-based language teaching aims to provide opportunities for learners to experiment with and explore both spoken and written language through learning activities designed to engage learners in the authentic, practical, and functional use of language for meaningful purposes. Learners are encouraged to activate and use whatever language they already have in completing a task. Using functions will also give a clear and purposeful context for the teaching and learning of grammar and other language features as well as skills. The role of task-based language learning is to stimulate a natural desire in learners to improve their language competence by challenging them to complete meaningful tasks.

Task-based language teaching has strengthened the following principles and practices:



- A needs-based approach to content selection.
- An emphasis on learning to communicate through interaction in the target language.
- The introduction of authentic texts into the learning situation.
- The provision of opportunities for learners to focus on language and the learning process itself.
- Enhancement of the learner's personal experiences significantly contributes to classroom learning.
- The linking of classroom language learning with language use outside the classroom.

Seven Principles for Task-Based Language Teaching

Principle 1: Scaffolding. Lessons and materials should provide supporting frameworks within which the learning takes place. At the beginning of the learning process, learners should not be expected to produce language that has not been introduced explicitly or implicitly. An essential role for an educator is to provide a supporting framework within which the learning can take place. The learners will encounter holistic 'chunks' of language often beyond their current processing capacity. The 'art' of TBLT is knowing when to remove the scaffolding. If the scaffolding is removed prematurely, the learning process will 'collapse.' If maintained too long, the learners will not develop the independence required for autonomous language use.



Principle 2: Task dependency. Within a lesson, one task should grow out of and build upon the ones that have gone before. Within the task-dependency framework, several other principles are in operation. One of these is the receptive-to-productive principle. At the beginning of the instructional cycle, learners spend more time engaged in receptive (listening and reading) tasks than productive (speaking and writing) tasks. Later in the cycle, the proportion changes, and learners spend more time in productive work. The reproductive-to-creative-language principle is also used in developing chains of tasks.

Principle 3: Recycling. Recycling language maximizes learning opportunities and activates the 'organic' learning principle. This recycling allows learners to encounter target language items in various linguistic and experiential environments. As such, they will see how a particular item functions with other closely related items in the linguistic 'jigsaw puzzle.' They will also see how it functions in different content areas.

Principle 4: Active learning. Learners learn best by actively using the language they are learning. A fundamental principle behind this concept is that learners learn best through doing – through actively constructing their knowledge rather than having it transmitted to them by the teacher. When applied to language teaching, this suggests that most class time should be devoted to opportunities for learners to use the language. These opportunities could be many and varied, from practicing memorized dialogues to completing a table or chart based on some listening input. The critical point is that the learner, not the teacher, is doing the work. This does not suggest that there is no place for teacher input, explanation, and so on but that such teacher-focused work should not dominate class time.



Principle 5: Integration. Learners should be taught in ways that clearly explain the relationships between linguistic form, communicative function, and semantic meaning. The challenge for pedagogy is to 'reintegrate' formal and functional aspects of language, and what is needed is a pedagogy that explicitly explains to learners the systematic relationships between form, function, and meaning.

Principle 6: Reproduction to creation. Learners should be encouraged to move from reproductive to creative language use. In reproductive tasks, learners reproduce language models provided by the teacher, the textbook, or the tape. These tasks are designed to give learners mastery of form, meaning, and function and provide a basis for creative tasks. In creative tasks, learners recombine familiar elements in novel ways. This principle can be deployed not only with students who are at intermediate levels and above but also with beginners if the instructional process is carefully sequenced.

Principle 7: Reflection. Learners should be given opportunities to reflect on what they have learned and how well they perform. Becoming a reflective learner is part of learner training, where the focus shifts from language content to learning processes.



This Curriculum is based on real-world communicative needs, oriented toward real-life tasks, and constructed around purposefully selected notions and functions. This promotes a proficiency perspective guided by “Can Do” descriptors.

In this approach in which knowledge and skill are blended, the learner can no longer be called the constructor of knowledge but the one who can combine new information with existing knowledge and carry acquired knowledge to future learning processes. Teachers are the facilitators and guides that guide the learning process, form the need, and take an active role with the learners in the learning process. Their task is to facilitate the acquisition of natural or near-real learning environments for acquiring language skills.

English for Specific Purposes (ESP)

English for Specific Purposes (ESP) refers to the teaching and learning of the English language that is tailored to meet the specific needs of learners in a particular technical career. Unlike general English language instruction, which aims to develop overall language proficiency, ESP focuses on developing the language skills, competencies, and knowledge necessary for effective communication in specific contexts to equip learners to succeed within their chosen field or profession. ESP courses use authentic materials, such as texts, documents, and multimedia resources, that reflect the language and communication demands of the learners' target field or career.

Breen suggests that when we place communication at the center of the curriculum, the goal of that curriculum (individuals who can communicate in the target language) and the means (classroom procedures that



develop this capability) begin to merge. Learners learn to speak by communicating. The ends and the means become the same.

ESP is a significant activity worldwide. It is an enterprise involving education, training, and practice that draws upon three significant realms of knowledge: language, pedagogy, and the students' / participants' specialist areas of interest.

ESP teachers generally have various simultaneous roles as researchers, course designers, material writers, testers, evaluators, and classroom teachers. These teachers need some knowledge of or at least access to information on any field of study that students are professionally involved with, such as business, tourism, agriculture, mechanics, computer science, drawing, accounting, and electronics (Robinson, p.1).



The Methodology Used in the Classroom

The Bureau of Technical Education and Entrepreneurship recommends **English Oriented to Software Applications Development** in **Eleventh** grade to implement a student-centered pedagogy that integrates collaborative learning, development of critical thinking skills, and conversation-based instruction around a problem or product in the classroom. The purpose of implementing this curriculum is to improve the level of instruction and, as a result, improve Costa Rican students' English communication skills through a student-centered pedagogy aligned with a technical orientation.

Aristotle said you must know *what* you are teaching, but you also need to know why and how. It isn't enough to know "the learnings" you are teaching. Some elements must be integrated into your classroom for your students to learn, such as their strengths, what they have already learned, and what matters to them.

Teaching **English Oriented to Software Applications Development** prioritizes communicative competence involving oral comprehension and oral and written communication so that they become independent users of English and can reach the B1.2 level based on the descriptors of the CEFR. Each level has scenarios and themes:

- Each theme presents an Essential Question which introduces the lesson.
 - a) They are open-ended and resist a simple or single correct answer.



- b) They are deliberately thought-provoking, counterintuitive, and controversial.
 - c) They require students to draw upon content knowledge and personal experience.
 - d) They can be revisited throughout the unit to engage students in evolving dialogue and debate.
 - e) They lead to other essential questions posed by students.
- The teacher shares the Essential Competence and the New Citizenship Axis at the beginning of each theme to connect students with the core ideas that have lasting value beyond the classroom.
 - Essential Competence is presented to the students; they need to follow human development competencies, which are already established to articulate the three learnings: learn to know, do, and be and live in a community.
 - The New Citizenship Axis might be *Sustainable Development Education, Digital Citizenship with Social Equity, and Strengthening Planetary Citizenship with Identity*.
 - Teachers select the goals from each theme. Depending on the lesson's pedagogical purpose, they can combine oral or written comprehension with oral and written production.
 - Teachers start the lesson with a warm-up activity related to the theme's name. Then, they share the learning goals/expected outcomes with the learners for that day or week.
 - Lessons follow a task-based approach combined with the action-oriented approach.



- Grammar is developed by combining inductive and deductive instruction within a meaningful context.
- The teacher follows integrated sequence procedures established to develop different linguistic competencies.



Curricular Design Template Elements

The elements considered in the curricular design are shown and defined in the following table:

Table 2

Curricular Elements of English Oriented to...

Element	Definition
CEFR	A tool promotes positive formulation of educational aims and outcomes at all levels.
Scenario	A real-life context is referenced for an entire unit, providing the authenticity of situations, tasks, activities, and texts.
Time	Number of hours devoted to the theme.
Essential Question	A question to develop and deepen students' understanding of essential ideas and processes so that they can transfer their learning within and outside school. It stimulates learner thinking and inquiry.
Theme	The focus of attention for communicative acts and tasks refers to the real-life scenario. (context rather than content)
Essential Competence	These are defined as competencies not specific to an occupation, which are needed for the comprehensive development of any person, professional, or



Element	Definition
	citizen. They are acquired during the development of the pedagogical mediation process, the performance of the discipline, and throughout life.
New Citizenship Axis	Sustainable Development Education Digital Citizenship with Social Equity Strengthening of Planetary Citizenship with Identity
Goals	Can do performance descriptors based on CEFR.
Oral and Written Comprehension Listening and Reading	What a learner can understand or can do when listening and reading.
Oral and Written Production Spoken production, Spoken Interaction and Writing	What a learner can produce in an oral and written way.
Performance Indicator	Describe observable behaviors and give information about the student's performance acquired during the learning process. It shows the achievement of knowledge, skills, abilities, and attitudes. It also contains two essential elements: Verb-Action and Condition.



Element	Definition
Pedagogical Task	They are communicative or non-communicative activities that demand knowledge, skills, and abilities and occur in the classroom.
Learnings	Learners need to know this to communicate effectively within a domain, scenario, and theme.
Functions	The use of spoken discourse and written texts in communication for a particular purpose (e.g., asking and giving information, describing)
Grammar	The grammatical components that will be covered in each theme.
Vocabulary	Words learners need to know to communicate effectively within a domain, scenario, and theme related to the field.
Phonology	The part of the lesson that addresses the learner's ability to hear, identify, and manipulate sounds.

Source: Prepared by the authors based on data supplied by CEFR, 2014.



Curriculum Template

Subject Area: English Oriented to Software Applications Development

Grade: Tenth

CEFR:

Scenario 1:

Theme 1:

Time: hours

Essential Question:

Essential Competences:

New Citizenship Axis:

Table 3

Curriculum Pedagogical Design

Goals	Performance Indicator
The learner can...	The student...



Table 4

Oral and Written Comprehension

Goals		Performance Indicator
The learner can...		The student...
Listening:		
Reading:		

Table 5

Oral and Written Production

Goals		Performance Indicator
The learner can...	The student...	
Spoken Interaction:		
Spoken Production:		
Writing:		



Annual Learning Plan

The annual plan is prepared based on the current study program, and the schedule presents the study program's development in months and weeks throughout the school year. It represents the time distribution of the scenarios, the themes to be developed, and their respective goals according to the study program.

The number of weeks and hours that will be devoted to developing each scenario must be indicated. This includes the names of the themes that make up each scenario and their goals.

In addition, it must respect the logical sequence that the study program provides for approaching the educational process. The information for preparing the annual plan must be taken from the curriculum, specifically the curricular structure, curricular grid, and scope and sequence.

This plan must be submitted to the School Principal in a printed or digital format, as established by the administration, at the beginning of the school year.



Table 6

Illustration 2

Annual Learning Plan

Annual Learning Plan

Technical High School:																																																
Subárea Area:	English Oriented to ...																								Level:																							
Teacher:																									Year:																							
Scenarios Theme and Goals	Feb				Mar				Apr				May				Jun				Jul				Aug				Sep				Oct				Nov				Dic				Hours			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4								
Scenario																																																
Theme																																																
Goals																																																



Pedagogical Practice Plan

This plan must be prepared monthly. It is for daily use at school and must be submitted to the principal, either printed or digital, as the school administration deems appropriate so that it can be verified that its development is consistent with the annual plan prepared at the beginning of the school year.

Definition of the Pedagogical Practice Plan template.

Its format includes the development of two aspects: administrative and technical qualities. The included administrative information is related to the name of the school, the name of the teacher, CEFR, grade,

In addition, it indicates the subject area, the scenario, the theme, and the estimated time for the teaching process. These aspects must follow the contents of the annual plan and, therefore, with the curricular structure, the curricular grid, and the scope and sequence of the study program.

The essential question, competence, and the educational policy axis are developed throughout the entire theme, and these elements are part of the development of the technical part of the pedagogical practice plan.

When planning, the teacher first writes the Essential Competence suggested in the study program and the associated tasks proposed by the teacher, second the New Citizenship Axis given in the program, and the tasks proposed by the teacher to accomplish it. Then, the teacher writes the Goals for Oral and Written



Comprehension: Listening and Reading, and finally, the goals for Oral and Written Production: Spoken Interaction, Spoken Production, and Writing. All of them are found in the study program.

The table Task Building Process shows how language learning should be directed towards enabling learners to act in real-life situations, express themselves, and accomplish tasks of different natures.

It has two columns: Task Mediation Activities and Performance Indicators.

The first column is a six-step pedagogical sequence for introducing tasks, a linked sequence of enabling exercises and activities to prepare learners to carry out different tasks, and the corresponding indicators. See the set out below.

Task-Building Process

Pre task

Schemata building. The first step is to develop several schema-building exercises that will introduce the topic, set the context for the task, and introduce some of the key vocabulary and expressions that the students will need to complete the task.



Example:

1. *Create opportunities for schemata-building to introduce the meaning of unknown vocabulary, structures, and functions for a concrete action according to the field of study.*

Task Rehearsal

Controlled practice. The next step is to provide students with controlled practice using the target language vocabulary, structures, and functions. In this way, early in the instructional cycle, they would get to see, hear, and practice the target language for the theme of work. This type of controlled practice extends the scaffolding learning that was initiated in the previous step. Learners are introduced to the language within a communicative context. In the final part of the step, they are also beginning to develop communicative flexibility. Involve learners in intensive listening practice. The listening texts could involve several native speakers. This step would expose them to an authentic or simulated conversation.

Examples:

2. *Expose learners to authentic materials related to the real world of communication in the field of study.*



Focus on linguistic elements.

The students now get to take part in a sequence of exercises in which the focus is on one or more linguistic elements. The task-based procedure being presented here occurs relatively late in the instructional sequence. Before analyzing aspects of the linguistic system, they have seen, heard, and spoken the target language within a communicative context. Hopefully, this will make it easier for the learner to know the relationship between communicative meaning and linguistic form than when linguistic elements are isolated and presented out of context, as in more traditional approaches.

Example:

3. *Focus on linguistic elements such as functions, discourse markers, grammar, and vocabulary required to review the essential question related to the field of study.*
4. *Give learners-controlled practice using the target language, vocabulary, structures, and functions.*

Post Task

Provide freer practice. The student should be encouraged to extemporize, using whatever language they have at their disposal to complete the task. Those who innovate will produce what is known as 'pushed output' (Swain 1995) because the learners will be 'pushed' by the task to the edge of their current linguistic competence. In this process, they will create their meanings and, at times, language, but over time, it will



Approximate more closely to native speaker norms as learners 'grow' into the language. (See Rutherford 1987 and Nunan 1999 for an account of language acquisition as an 'organic' process.)

Example:

5. *Engage learners in meaningful, productive tasks based on the context.*

Assessment

The final step in the instruction to assess is the pedagogical sequence itself. Having worked through the sequence, students find it highly motivating to arrive at step 6 and see that they can create a project successfully.

Example:

6. *Project: integration of activities. It must be done in class.*

In the second column, Performance Indicators are measurable variables used to assess the progress or success of students in reaching specific goals. These indicators provide tangible evidence of knowledge, performance, or product, allowing the teacher to evaluate the effectiveness of efforts, make informed decisions, and track progress over time. Teachers can use some macro indicators given in the study program, and they are responsible for generating the achievement indicators based on the proposed task mediation activities so the students can demonstrate they have accomplished the expected competencies for each theme.



Performance indicators established by the teacher in the Pedagogical Practice Plan must be consistent with the information in the assessment instruments developed to evaluate performance. The evidence resulting from this process must be filed in the student's evidence portfolio.

Finally, the teacher writes the required pedagogical resources to develop the task mediation process: the classroom, English laboratory, devices, and material needed for each theme.

Pedagogical Recommendations

- Teacher makes sure that all learners understand task instructions.
- Teachers should ensure learners know how to use strategies through teacher scaffolding and modeling, peer collaboration, and individual practice.
- Learners have at their disposition valuable words, phrases, and idioms that they need to perform the task. The task could be an audio recording with instructions and the pronunciation of the required words and phrases.
- The task could involve integrating listening, speaking, reading, and writing and is given to students individually, in pairs, or in teams.
- The learners complete the task together using all their resources. They rehearse their presentation, revise their written report, present their spoken reports, or publish their written reports.



- The teacher monitors the learners' performance and encourages them when necessary.
- The learners consciously assess their language performances (using rubrics, checklists, and other technically designed instruments provided and explained to them in advance). Teachers assess performance, provide feedback through assistance, bring back helpful words and phrases to learners' attention, and offer additional pedagogical resources to learners who need more practice.
- At the end of each period, the learners develop and present Integrated Mini-Projects to demonstrate mastery of the scenario goals.
- The Essential Competencies and The New Citizenship Axis correspond to the educational policy to articulate the three learnings: learning to know, do, be, and live in the community. The Integrated Mini-Project allows students to integrate these three learnings in a single task.
- Teach and plan English lessons to engage learners socially and cognitively, following the steps mentioned above.



Pedagogical Practice Plan

Institution:

Teacher:

Subject Area:

Grade:

CEFR:

Scenario:

Themes:

Time:

Essential Question:

Essential Competences:

New Citizenship Axis:



Linguistic competences

Oral and Written Comprehension Goals:

Listening:

Reading:

Oral and Written Production Goals:

Spoken Interaction:

Spoken Production:

Writing



Table 6

Task Building Process

Task Mediation Activities :	Performance Indicators
Pre Task: <i>Schemata-building</i> 1. Create opportunities for schemata-building to introduce the meaning of unknown vocabulary, structures, and functions <i>for a concrete action according to the field of study</i>	
Task Rehearsal: <i>Controlled practice</i> 2. Expose learners to authentic materials to deal with the real world of communication-related to the field of study. 3. Focus on linguistic elements such as functions, discourse markers, grammar, and vocabulary.	
4. Give learners-controlled practice using the target language, vocabulary, structures, and functions.	



Task Mediation Activities :	Performance Indicators
Post Task: 5. Engage learners in meaningful, productive tasks based on the context	
Assessment: 6. Project: integration of activities. It has to be done in class.	

Resources:

Classroom:

English laboratory:

Devices:

Materials:



Evaluation of the Learning Process

Talking about linguistic competence evaluation means incorporating new assessment strategies. In this regard, it emphasizes the importance of implementing a learning-oriented evaluation focused on student participation, aimed at situations that are authentic and increasingly closer to real life. Therefore, competence is contextual; it reflects the relationship between people's skills and the activities they perform in a particular situation in the real world (adapted from López, 2014).

Linguistic competence evaluation is a continuous, dynamic, holistic approach aimed at analyzing the performance levels achieved by the student. In this sense, evaluation fulfills a self-regulation function that empowers students to actively monitor their learning progress and take responsibility for their development.

From this perspective, competence predicts performance; it is directly linked to the student's practical processes and not so much to data accumulation. The evaluation identifies and records the acquisition of the linguistic competencies to be developed through the methods and the evidence generated by the student to evaluate the evolution of the domain. Teachers make judgments based on the processes and evidence of their students through the observation and analysis of the evolution of the domain of each level.

Evaluation must be aligned with the curriculum. There must be a balance among goals, mediation strategies to be developed throughout the educational process, and a system for evaluating knowledge, performance, and expected products according to established performance indicators.



Evaluation offers strategies that allow in-depth knowledge of the students' results and awareness of their expectations. Through linguistic competence evaluation, students provide teachers, parents, classmates, and the community with "evidence" of their performance through new tools and evaluation methods. These tools are based on a constructivist perspective, and their dynamics focus on processes.

Upon selecting the pedagogical mediation strategies, the evaluation instruments are defined. They include the achievement indicators and performance criteria by which the learning situation will be evaluated since they allow the teacher to judge what each student has achieved.

The Learning Evaluation Regulations, approved through an executive decree, govern the Costa Rican evaluation and establish the evaluation components of each modality of the educational system. The grade of each subject, for each period, is obtained from the sum of the percentages corresponding to the grades obtained by the student in each component. Below is a description of the evaluation components currently established by the Learning Evaluation Regulations (REA) for the experimental workshops and sub-areas developed in Technical Vocational Education, in both daytime and evening modalities and in a two-year program. REA defines the percentage value of the components as appropriate.

- **Daily work.** It consists of the educational activities carried out by students with the guidance and orientation of the teacher according to the pedagogical practice plan and the curriculum.



Technically prepared instruments must record information related to the student's performance to evaluate it. This information is collected over the period and lessons as part of the teaching-learning process, not as a product; it must reflect the student's gradual learning progress.

The daily work includes preparing the evidence portfolio in the technical specialties of the Curriculum of Adult Education and Technical Diversified Education.

- **Homework.** It consists of short tasks assigned to students to reinforce their expected learning according to the information collected during daily work. Students can review or reinforce the expected learning through these assignments. Therefore, these assignments must be carried out exclusively by the students so that they can strengthen their knowledge. Homework should not be assigned during school hours or vacation periods, Easter and mid-year, nor scheduled during testing periods at the school.
- **Tests.** These are measuring instruments intended for students to demonstrate the acquisition of cognitive, psychomotor, or linguistic skills. They can be written, performance, or oral tests. The expected learnings and indicators are selected according to the current study program of the corresponding level to construct these instruments.

Quizzes must be formative, except when those are applied to students with educational needs.

- **Project.** This is a learning construction process, guided and oriented by the teacher. It is based on the identification of the student's contexts of interest. It is related to the learning and linguistic competencies goals, acquired learning, values, attitudes, and practices proposed in each thematic unit of the study



program. The purpose is for students to apply what they have learned by reflexively completing a systematic set of actions of interest in a specific context of their sociocultural environment.

It can be completed individually or in groups. For project evaluation, students must receive indicators and criteria according to the stages defined for such project and consider both the process and the product, as well as evidence of self-evaluation and co-evaluation.

- **Attendance.** Attendance is the student's presence at lessons and all other school activities to which the student is conducted. Absences and tardies may be excused or unexcused (MEP, 2018, Art. 25-30).

Currently, there is a range of strategies and tools that the teacher can use as part of the evaluation process of some of the mentioned components, as is in the case of daily work: concept map, portfolio of evidence, timeline, mental map, cognitive maps, video forum, projects, collage, complete sessions, oral presentations, among many others. The teacher must prepare technically formulated evaluation instruments that show indicators and allow visualizing the student's achievement level in compliance with current regulations and the ministerial guidelines issued.

Written and performance tests are crucial instruments for evaluating student performance. They must be prepared according to the technical guidelines established by the MEP Learning Assessment Department.

In addition to having a percentage assigned in the component of the daily work evaluation, the portfolio of evidence is a valuable evaluation tool because the proof of the student's learning process in the development



**MINISTERIO DE
EDUCACIÓN PÚBLICA**

**GOBIERNO
DE COSTA RICA**

**Dirección de Educación
Técnica y Capacidades
Emprendedoras**

of linguistic competencies must be observed in it, according to the guidelines established by the Directorate of Technical Education and Entrepreneurial Skills.



Curricular Structure

Level: Eleventh

Table 8

Themes per scenarios

Scenarios	Weekly Hours	Yearly Hours
Entrepreneurship	4	48
Data-Base	4	60
Software Development	4	52
Total		160



CURRICULAR GRID: ENGLISH-ORIENTED TO SOFTWARE APPLICATIONS DEVELOPMENT

Level: Tenth

Table 9

Themes per scenarios

Scenarios	Theme 1	Theme 2	Theme 3	Theme 4
1. Digital Transformation	Digital Literacy (16 Hours)	Internet of Things (IoT) (16 Hours)	Artificial Intelligence (16 Hours)	
2. Fundamentals of Connectivity and Cybersecurity	Connectivity (16 Hours)	Hardware (16Hours)	Cybersecurity Essentials (16Hours)	
3. Programming, Virtualization and Networking	Flow Chart (16 Hours)	Programming (16 Hours)	Virtualization (16 Hours)	Introduction to Networking (16 Hours)



Level: Eleventh

Table 10

Themes per scenarios

Scenarios	Theme 1	Theme 2	Theme 3
Entrepreneurship	Business Opportunities and Models (24 Hours)	Creation of a Company (24 Hours)	
Data-Base	Data-Base Design (20 Hours)	Relational Data-Based (20 Hours)	Non-Relational Data Based (20 Hours)
Software Development	Agile and Quality Development (20 Hours)	Object-Oriented Programming (32 Hours)	



Level: Twelfth Grade

Table 11

Themes per scenarios

Scenarios	Theme 1	Theme 2
Programming	Class Design and Exception Models (20 Hours)	Mobile Applications (32 Hours)
Network and Virtualization	User Experience (28 Hours)	Interaction Design (20 Hours)



Curriculum Scope and Sequence

Grade: Eleventh

Scenario 1: Entrepreneurship

Theme 1: Business Opportunities and Models

Goals:

Essential competence: Propose proactively the needs and opportunities of the market.

New citizenship axis: Offer a variety of solutions to current situations in their day-to-day living by applying technology.

Listening:

Understand the main points of clear standard speech about the market and its environment in order to identify business opportunities, according to new trends.

Reading: Understand the important information in simple articles in newspapers or magazines about the generation of innovative business ideas, providing solutions to the needs detected in potential customers.

Spoken interaction: Enter unprepared into a conversation of possible solutions to market needs and opportunities, models, express personal opinions and exchange information.



Spoken production:

- Reasonably fluently sustain a straightforward description of how to develop the plan for an ongoing proposal for the business model and product launch.
- Produce familiar sounds and prosodic patterns.

Writing:

- Work out how to communicate the main points he/she wants to get across in a business.
- Summarize, report, and give his/her opinion about accumulated factual information when applying negotiation strategies in the process of validating business proposals with some confidence.

Theme 2: Creation of a Company for Living

Goals:

Essential competence: Employ empowerment as a tool in the development of skills for strengthening his/her performance in the technical field, personal training, and for his/her life plan.

New citizenship axis: Estimate the level of empowerment achieved in entrepreneurship management according to the goals and objectives proposed in the plan deal.

Listening:



Follow a lecture or talk about applying the service principles with a customer-oriented approach in the implementation of the business plan, provided the presentation is straightforward and clearly structured.

Reading: Find and understand relevant information in official documents to choose the best strategies for information search through the use of technologies individually or collaboratively.

Spoken interaction: Follow clearly articulated speech directed at him/her in a conversation, about the description of the types of companies with which a business can be developed, though will sometimes have to ask for repetition of particular words and phrases.

Spoken production:

- Give a straightforward description of job performance in the functional areas that make up the proposed practice company applying the provisions of the business plan.
- Produce sounds and prosodic patterns.

Writing: Write straightforward connected texts about structuring the business with a customer-oriented approach based on the business plan.

Scenario 2: Data-Base

Theme 1: Data-Base Design

Goals:



Essential competence: Establish innovative strategies and mechanisms to respond with efficiency to the constant changes in modern working environments.

New citizenship axis: Engage in dynamic digital environments that facilitate the achievement of common social changes with fairness and invention.

Listening: Understand simple technical information, such as characteristics, elements, and fundamental concepts associated with databases.

Reading:

- Search the internet, or other reliable sources of information, for specific everyday or work-related material about database uses, applications, and contributions to daily work.
- Read newspapers/magazine accounts of films, and books, written for a wider audience and understand the main points regarding the purpose, types, and uses of Artificial Intelligence.

Spoken interaction: Provide reasons and explanations, to a specific audience about elements of database design, to solve a situation, using simple language.

Spoken production:

- Make a short instructional or informational text easier to understand by presenting it as a list of separate points about the elements involved in database design.



- Produce sounds and prosodic patterns.

Writing: Write a basic description of procedures to achieve AI databases that should care for business.

Theme 2: Relational Data-Based

Goals:

Essential competence: Show willingness to work collaboratively to achieve common goals.

New citizenship axis: Determines how the Industrial Internet of Things becomes a supreme competitive advantage for Artificial Intelligence.

Listening:

Understand the use of information content of the majority of recorded audio material about relational databases for solving specific challenges delivered in clear standard speech.

Reading: Read newspapers / magazine accounts of films, and books, written for a wider audience and understand the main points regarding the examples and uses of relational databases.

Spoken interaction: Follow what is said about the use of relational databases with the help of technological devices, though he/she may occasionally have to ask for repetition or clarification if the other people's talk is rapid or extended.



Spoken production:

- Give simple reasons to justify a viewpoint of the applications of relational databases.
- Produce unfamiliar sounds and prosodic patterns.

Writing: Write a brief standard report conveying factual information, stating the benefits and disadvantages of relational databases.

Theme 3: Non-Relational Data Based

Goals:

Essential competence: Implement decision-making with empowerment for different processes that require the technical skills of the database.

New citizenship axis: Develop skills as app developers for a healthy coexistence in the world respecting human rights.

Listening: Follow a straightforward presentation or demonstration about the use of non-relational databases.

Reading: Find and understand relevant information about the use of non-relational databases.

Spoken interaction: Start up a conversation and help it to keep going by asking people relatively spontaneous questions about non-relational databases as widely recognized for building hi-tech applications.



Spoken production: Explain the main points and crucial aspects to take into account when using a non-relational database.

Writing: Present a topic in a short report or poster, using photographs and short blocks of text about non-relational databases.

Scenario 3: Software Development

Theme 1: Agile and Quality Development

Goals:

Essential competence: Develop the ability to apply problem-solving techniques in Agile software development by identifying quality issues, analyzing root causes, and implementing continuous improvement strategies to enhance product quality and team performance.

New citizenship axis: Understand and apply sustainable development principles in Agile software development by promoting resource-efficient practices, fostering collaboration, and ensuring the delivery of high-quality software that supports long-term social, economic, and environmental sustainability.

Listening:

- Understand the main ideas and key points in discussions and presentations on Agile practices and quality assurance in software development when clear, standard language is used.



- Follow detailed explanations and instructions related to Agile workflows and quality improvement processes, identifying important information even in moderately complex contexts.

Reading:

- Identify and understand key information in texts related to Agile methodologies and quality assurance, such as guides, reports, and process documentation.
- Interpret detailed descriptions of Agile practices and quality improvement strategies, recognizing relevant concepts and instructions within moderately complex written materials.

Spoken interaction:

- Participate in discussions about Agile practices and quality improvement, expressing opinions, asking for clarification, and responding to others clearly and coherently.
- Engage in team conversations on Agile processes and quality issues, using appropriate vocabulary to collaborate, share ideas, and solve problems in familiar work-related situations.

Spoken production:

- Explain Agile concepts and quality management practices in software development clearly and in an organized way, using appropriate terminology.
- Prepare and deliver short presentations or reports on Agile processes and quality improvement strategies, highlighting key points and providing relevant examples.



- Produce sounds and prosodic patterns.

Writing: Write clear and structured texts, such as reports or summaries, on Agile practices and quality assurance processes, using appropriate vocabulary and linking ideas coherently.

Theme 2: Object-Oriented Programming

Goals:

Essential competence: Implement preventive techniques aimed at maintaining self-control.

New citizenship axis: Demonstrate actions that promote sustainable development.

Listening: Follow much of everyday conversation and discussion about object-oriented programming (OOP) provided it takes place in standard speech and is clearly articulated in a familiar accent.

Reading: Scan longer texts to gather information from different parts of the text, or from different manuals to fulfill a specific task related to object-oriented programming languages.

Spoken interaction: Take part in routine formal discussions that are conducted in clearly articulated speech in the standard form of the language, and which involve the exchange of factual information, receiving instructions, or the discussion about object-oriented programming and problem-solving.

Spoken production:



- Deliver short, rehearsed announcements despite possibly very foreign stress and intonation, are nevertheless clearly intelligible when talking about object-oriented programming and problem-solving.
- Produce sounds and prosodic patterns.

Writing: Write a very brief report in a standard conventionalized format about the applications of Object-Oriented Programming, articulating one's efforts with others.

Curricular Design

Subject Area: English Oriented to Software Applications Development

Grade: Eleventh

CEFR : B1.2

Scenario 1: Entrepreneurship

Theme 1: Business Opportunities and Models

Time: 24 hours

Essential Question

How to bring great business opportunities together?



Essential Competences: Proactive attitude

New Citizenship Axis: Digital Citizenship with Social Equity

Table 7

Curriculum Pedagogical Design

Goals	Performance Indicator
The learners can...	The student...
Propose proactively the needs and opportunities of the market.	-Identifies the importance of a proactive attitude in the day-to-day aspects of their work.
Offer a variety of solutions to current situations in their day-to-day living by applying technology.	-Participates in technology contests or fairs to demonstrate the application of new technological devices developed to contribute to our daily lives.

TABLE 8

Oral and Written Comprehension

Goals	Performance Indicator
The learners can...	The student...



Listening:

Understand the main points of clear standard speech about the market and its environment in order to identify business opportunities, according to new trends.

Recognizes market opportunities according to new trends in order to characterize market performance, market dynamics, and potential customers.

Reading: Understand the important information in simple and clear articles in newspapers or magazines about the generation of innovative business ideas, providing solutions to the needs detected in potential customers.

Discriminates sources for generating business ideas according to the information read and selects business ideas using different techniques.



Table 9

Oral and Written Production

Goals The learners can...	The student...	Performance Indicator
Spoken Interaction: Exploit a wide range of simple language to deal with situations likely to arise whilst explaining the characteristics and importance of entrepreneurship. Enter unprepared into the conversation of possible solutions to market needs and opportunities, express personal opinions, and exchange information.	<ul style="list-style-type: none">• Provides a basic description of the skills and responsibilities of the entrepreneur and discriminates the elements for undertaking a project.• Suggests possible solutions for market needs and opportunities or enhance existing ones with a proactive attitude for enterprise development.	



Spoken Production:

Reasonably fluently sustain a straightforward description of how to develop the plan for an ongoing proposal for the business model and product launch.

Produce familiar sounds and prosodic patterns.

- Talks about the aspects that should be considered in the start-up of the business model to build the business plan implementation taking into account the impact mitigation strategies.
- Employs a range of phonological features in the target language by manipulating prosodic features of spoken discourse(e.g. stress, intonation, rhythm) to support the message intended to convey.

Writing:

Work out how to communicate the main points he/she wants to get across of a business model based on an

- Designs business ideas with greater chance of success from the application of current tools and methodologies.
- Writes negotiation strategies that foster successful agreements during the business proposal validation process and negotiate the execution of viable entrepreneurship proposals.



innovative idea using
current tools and
methodologies.

Summarize, report, and
give his/her opinion
about accumulated
factual information
when applying
negotiation strategies in
the process of validating
business proposals with
some confidence.



Table 10

Learnings of the curriculum pedagogical design

Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Functions</p> <p>Defining terms related to the generation of innovative business ideas.</p> <p>Describing the characteristics and importance of entrepreneurship.</p> <p>Distinguishing types of business model</p> <p>Discourse Markers</p> <p>Sequential future time</p>	<p>Modals must/can't deduction</p> <p>Modals might, may, will, probably</p> <p>Modals for Deduction: Must / Can't (Present Deduction)</p> <p>These modals are used to express certainty about a situation in the present based on evidence or logical reasoning.</p> <p>Must → used when we are sure something is true.</p> <p>"He must be at work because his car is parked outside."</p>	<p>SWOT Analysis:</p> <p>Analysis of the capacities, resources, strategies, competitive advantages, strengths and weaknesses of the current potential competitors of a company, which is carried out to make decisions.</p> <p>Customer Analysis:</p> <p>Analysis of the needs,</p>	<p>Pronouncing Final Consonant Clusters</p> <p>a. Consonant Clusters at the beginning of the words.</p> <p>b. Consonant Clusters in the middle of words.</p> <p>c. Consonant Clusters at the end of the words.</p>



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Initiating and closing conversation</p> <p>These markers help to indicate the order in which events will happen in the future.</p> <p>First</p> <p>"First, we will finish the current project before moving on to the new one."</p> <p>Then</p>	<p>Can't → used when we are sure something is not true.</p> <p>"She can't be the manager; she's too young."</p> <p>Modals for Possibility: Might, May, Could (Present Deduction)</p> <p>These modals express less certainty. They indicate that something is possible, but not certain.</p> <p>Might / May / Could → used when we think something is possible but not sure.</p> <p>"They might be at the gym."</p> <p>"It may rain later."</p>	<p>tastes, preferences, desires, consumption habits, purchasing behaviors, customs, attitudes and other characteristics of the consumers that make up a target market.</p> <p>Advertisement: Message that is sent to the public through advertising means in order to publicize, inform, persuade their purchase, consumption or use, or to remind a</p>	<p>PROSODIC FEATURES:</p> <p>A. STRESS</p> <p>1. Stress within the word.</p> <p>a. Words stressed on the first syllable.</p> <p>b. Words stressed on the second syllable.</p> <p>c. Words stressed on the third syllable.</p> <p>d. Stress in nouns/verbs</p>



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
"I will send the email, then I'll start preparing for the meeting." Next "Next, we will review the results from last quarter." After that "After that, we can discuss the next steps for the marketing campaign." Finally	"He could be the new teacher." Note: Might, may, and could have similar meanings for possibility, with might being the most commonly used in spoken English. Modal with Will and Probably (Expressing Future Deduction) At the B1.1 level, will and probably are often used together to predict or express logical conclusions about the future. Will (certainty) "She will definitely come to the meeting." Will probably (likely prediction)	product or service. Customer Service: Attention that a company provides to its customers. When it comes to giving good customer service, it is usually referred to being nice to them, courteous or helpful. Database: Set of data related to a certain aspect of a company that is systematically stored for later use. Benchmarking: A management	(Homographs)



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>"Finally, we will schedule the follow-up meeting to review progress."</p> <p>Idioms and common colloquial expressions</p> <p>Here are 8 examples of idioms and common colloquial expressions that can be used within the topic of Business Opportunities and Models:</p> <p>"Think outside the box"</p>	<p>"They will probably move to a bigger office next year."</p> <p>Examples in Context:</p> <p>"It must be cold outside; everyone is wearing a coat."</p> <p>"He can't be the new employee. I met him last year."</p> <p>"She might be late because of the traffic."</p> <p>"We may go to the meeting if we finish on time."</p> <p>"It will probably take an hour to get there."</p> <p>References:</p>	<p>technique or tool that consists of taking as a model or reference the best aspects or practices of other companies, whether they are direct competitors or belonging to another sector (and, in some cases, other areas of the company), and adapt them to the company by adding improvements and creativity.</p> <p>Electronic Newsletter:</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Meaning: To think creatively and unconventionally, especially in business.</p> <p>"In order to succeed, we need to think outside the box and come up with innovative business models."</p> <p>"Strike while the iron is hot"</p> <p>Meaning: To take advantage of a business opportunity at the right moment.</p>	<p>Recommended sources where you can find information about modals (must, can't, might, may, will, probably):</p> <p>Cambridge University Press. (n.d.). Modals of deduction and speculation. Cambridge Dictionary. Retrieved from https://dictionary.cambridge.org</p> <p>British Council. (n.d.). Modals of deduction and possibility. Learn English – British Council. Retrieved from https://learnenglish.britishcouncil.org</p> <p>Carter, R., & McCarthy, M. (2016). English Grammar Today: An A–Z of</p>	<p>Publication regularly distributed by email to people who have previously subscribed.</p> <p>Quality: Set of properties and characteristics of a product or service to meet consumer expectations for example: the design, the presentation, the aesthetics, the conservation, the durability, the customer service and the after-sales service.</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>"We should invest in that new technology now—strike while the iron is hot!"</p> <p>"A win-win situation" Meaning: A situation where all parties benefit, often used when discussing partnerships or deals.</p> <p>"This partnership is a win-win situation for both companies."</p> <p>"Put all your eggs in one basket"</p>	<p>Spoken and Written Grammar. Cambridge University Press. Retrieved from https://www.cambridge.org/grammar/today</p> <p>University of Victoria English Language Centre. (n.d.). Modals of certainty, possibility, and impossibility. University of Victoria. Retrieved from https://continuingstudies.uvic.ca</p> <p>BBC Learning English. (n.d.). English at B1 level: Modals of deduction and possibility. BBC Learning English. Retrieved from</p>	<p>Total Quality: Philosophy, culture or management style where all members of a company seek to improve quality.</p> <p>Distribution Channel: Channel or medium through which the products of a company are distributed to where they will be offered or sold to consumers. A distribution channel can be direct (products are sold in a</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Meaning: To rely on a single investment or business venture, which can be risky.</p> <p>"It's not wise to put all your eggs in one basket. Diversifying your investments is key."</p> <p>"Hit the ground running"</p> <p>Meaning: To start something with a lot of energy and enthusiasm, often at the beginning of a business venture.</p>	<p>https://www.bbc.co.uk/learningenglish</p>	<p>store or own premises), or indirect (use intermediaries, to wholesalers or retailers).</p> <p>Publicity Channel:</p> <p>Examples of an advertising channel are television, radio, newspapers, the Internet, the mail, signs, billboards, posters.</p> <p>Product Lifecycle: Set of stages through which a product passes from its launch</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>"We need to hit the ground running with this new product launch."</p> <p>"Get the ball rolling" Meaning: To begin a project or start making progress on something.</p> <p>"Let's get the ball rolling on this new business model and start planning the first steps."</p> <p>"In the driver's seat" Meaning: To be in control of a situation, often used in the context of a business</p>		<p>to its exit from the market. The stages of the product life cycle are: the introduction, the expansion or growth stage, the maturity stage or stagnation and the stage of decline.</p> <p>Sales closure: Stage of the sales process in which, after having presented the product to the potential customer, the seller induces him to decide on the</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
leader or decision-maker. "With the new investment, we're now in the driver's seat when it comes to expansion." "A stepping stone" Meaning: A stage in the process of achieving something bigger, often used in relation to career or business growth. "This new venture will be a stepping stone to		purchase. Demand: Total volume in physical or monetary terms of one or more products, which is demanded by a market for a certain period of time. There is a market demand, potential or unsatisfied. Email marketing: Type of marketing that exclusively uses electronic mail to promote a product or service, to maintain	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
greater business opportunities."		<p>contact with a consumer, to create interest for a product or service, or seek customer loyalty.</p> <p>Strategy: Action that is carried out in order to achieve certain objectives.</p> <p>Customer Expectations: Expectations that consumers have before buying or acquiring a product or service, and that they obtain due to</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>the product or service advertising, based on previous experiences or comments from other consumers.</p> <p>Loyalty: Act and effect of making a customer loyal to a brand, product or service.</p> <p>Focus group: a demographically diverse group of people assembled to participate in a guided discussion about a particular</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>product before it is launched.</p> <p>Industry: Group of companies that produce similar products. Examples: the textile, food, the automotive industry.</p> <p>Product Line: Group of products or related to each other that a company owns.</p> <p>Trademark: Name, term, sign, symbol, design, or a combination of these that is assigned to a</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>product, service, company, or business to distinguish it from other products, services, companies, or businesses that exist in the market.</p> <p>Market: Place where buyers and sellers meet to carry out transactions of goods and services.</p> <p>Advertising: is what a company says about its own product, giving out information for promotional</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>purposes.</p> <p>Customer Satisfaction: when the consumers have met or exceeded the expectations about a product they have bought or a service they have received.</p> <p>Competitive Advantage: a superior performance that a company could have in some aspect over other companies from the same sector or market, for example:</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		the brand, the customer service, the production process, the technology, the personnel, the infrastructure, the location, the distribution.	

Curricular Design

Subject Area: English Oriented to Software Applications Development

Grade: Eleventh

CEFR : B1.1

Scenario 1: Entrepreneurship



Theme 2: Creation of a Company for a Living

Time: 24 hours

Essential Question: How does empowering employees benefit a company?

Essential Competences: Empowerment

New Citizenship Axis: Strengthening of Planetary Citizenship with Identify

Table 16

Curriculum Pedagogical Design

Goals	Performance Indicator
The learners can...	The student...
Employ empowerment as a tool in the development of skills for strengthening his/her performance in the technical field, personal training, and for his/her life plan.	Explains life projects with empowerment by creating a company based on the available learning opportunities, obstacles, and developed skills.
Estimate the level of empowerment achieved in entrepreneurship management according to the goals and objectives proposed in the plan deal.	Describes lessons learned in their personal and professional development, adapting to a changing environment.



TABLE 17

Oral and Written Comprehension

Goals	Performance Indicator
The learners can...	The student...
Listening: Follow a lecture or talk about applying the service principles with a customer-oriented approach in the implementation of the business plan, providing the presentation straightforward and structured.	Distinguish the difference between attention and customer service to develop your business plan considering the customer as the main axis on which revolves your enterprise.
Reading: Find and understand relevant information in official documents to choose	<ul style="list-style-type: none">• Distinguish the types of companies with which a business can be developed.• Applies current technological tools in the market for the operation of your practice enterprise.



the best strategies for
information search
through the use of
technologies individually
or collaboratively.

Table 18

Oral and Written Production

Goals		Performance Indicator
The learners can...	The student...	
Spoken Interaction: Follow articulated speech directed at him/her in a conversation about the description of the types of companies with which a business can be developed, though will sometimes have to ask	<ul style="list-style-type: none">• Compares the types of companies that interact in the national financial and economic system.• Select the type of company for the development of its business model.	



for repetition of particular words and phrases.	
Spoken Production: Give a straightforward description of job performance in the functional areas that make up the proposed practice company applying the provisions of the business plan. Produce familiar sounds and prosodic patterns.	<ul style="list-style-type: none">• Identifies the functional areas and tasks that are executed for the start-up of the business using technology in transactions to increase productivity in the company.• Employs a range of phonological features in the target language by manipulating prosodic features of spoken discourse (e.g., stress, intonation, rhythm) to support the message intended to convey.
Writing: Write straightforward connected texts about structuring the business with a customer-	Develops the company's organizational structure, processes, and procedures based on the business plan and using the customer-oriented approach.



oriented approach
based on the business
plan.

Table 19

Learnings of the curriculum pedagogical design

Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Functions</p> <p>Expressing the importance of the implementation of a business plan.</p> <p>Describing the types of companies.</p>	<ul style="list-style-type: none"> Modals – should have/might have/etc Modals: must/have to <p>Modals: Should Have / Might Have / Could Have (Past Deduction or Regret)</p>	<p>Debit: remove an amount of money from a customer's bank account.</p> <p>Amortization: Repayment or payment of a debt.</p> <p>Financial Analysis: Analysis of the projection of sales, costs and profits</p>	<p>Pronouncing Final Consonant Clusters</p> <p>a. Consonant Clusters at the beginning of the words.</p> <p>b. Consonant Clusters in the middle of words.</p>



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Describing challenges related to the implementation of a business plan.</p> <p>Selecting the most appropriate functional areas and tasks to start up a business plan.</p> <p>Describing the necessary company's organizational structure, processes, and procedures to create a company.</p>	<p>These modals are used to talk about possibilities, deductions, or regrets about past actions.</p> <p>Should have + past participle → to express advice or a regret about the past (something that didn't happen but would have been a good idea).</p> <p>"You should have studied more for the exam." (You didn't study enough, and now you regret it.)</p>	<p>of a new product to determine if these factors meet the objectives of the company.</p> <p>Personal Guarantee: An individual's legal promise to repay credit issued to a business for which they serve as an executive or partner, it means that if the business becomes unable to repay the debt, the individual assumes personal responsibility for the balance.</p> <p>Beneficiary: Recipient of</p>	<p>c. Consonant Clusters at the end of the words.</p> <p>PROSODIC FEATURES:</p> <p>A. STRESS</p> <p>1. Stress within the word.</p> <p>a. Words stressed on the first syllable.</p> <p>b. Words stressed on the second syllable.</p> <p>c. Words stressed on the third syllable.</p> <p>d. Stress in nouns/verbs</p>



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Discourse Markers</p> <p>Sequential future time</p> <p>Initiating and closing conversation</p> <p>These markers help to indicate the order in which events will happen in the future.</p> <p>First</p> <p>"First, we will finish the current project before moving on to the new one."</p> <p>Then</p>	<p>Might have / Could have + past participle → to express possibility in the past (something that was possible but we're not certain if it happened).</p> <p>"She might have missed the train." (It's possible she missed it.)</p> <p>"He could have forgotten the appointment." (It's possible he forgot.)</p> <p>Must have + past participle → to express</p>	<p>the funds of an operation.</p> <p>Business Angel: A natural person willing to invest in ventures in the early stages of their development.</p> <p>Credit Rating: Rating, applied to an individual or company, which indicates the credit risk that this individual or company represents.</p> <p>Social Capital: Number of financial resources contributed by the partners to a company.</p>	<p>(Homographs)</p>



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>"I will send the email, then I'll start preparing for the meeting."</p> <p>Next</p> <p>"Next, we will review the results from last quarter."</p> <p>After that</p> <p>"After that, we can discuss the next steps for the marketing campaign."</p> <p>Finally</p> <p>"Finally, we will schedule the follow-up meeting to review progress."</p>	<p>certainty about a past action or situation.</p> <p>"They must have left early because the house is empty."</p> <p>Can't have + past participle → to express certainty that something did not happen in the past.</p> <p>"He can't have finished the project already; it's too soon."</p> <p>Modals: Must / Have to (Present Obligation)</p>	<p>Commission: Amount, normally a percentage, charged by the intermediary for carrying out any financial transaction.</p> <p>Opening Commission: Commission charged at the time of formalizing a financial transaction.</p> <p>Closing and early cancellation commission: Commissions that are paid to cover accounting and documentation activities at the end of the</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Idioms and common colloquial expressions</p> <p>Get the ball rolling – Meaning: To start something, especially a business or project. "Let's get the ball rolling on our new startup idea."</p> <p>Burn the midnight oil – Meaning: Work late into the night. "We'll have to burn the midnight oil to meet the launch deadline."</p>	<p>These modals express obligation or necessity in the present.</p> <p>Must → used for a strong personal obligation or a rule. "You must wear a seatbelt while driving." "I must finish this report before the deadline."</p> <p>Have to → used for external rules, laws, or general obligations.</p>	<p>payment of a given loan, either at the agreed maturity or in advance.</p> <p>Subrogation Commission: Commission charged when substituting the ownership of a right or obligation for another person under the same conditions as the financial transaction.</p> <p>Availability Commission: Charge, normally quarterly, made on the capital not disposed of in a line of credit.</p> <p>Cash advance fee: is a</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Think outside the box – Meaning: Be creative and come up with innovative ideas. "To stand out from competitors, you need to think outside the box."</p> <p>Cut corners – Meaning: Do something the easiest or cheapest way, often sacrificing quality.</p>	<p>"We have to be at school by 8 a.m." (It's a rule.)</p> <p>"You have to show your ID to enter the building."</p> <p>Note: Must and Have to are often interchangeable, but must feel stronger and more formal.</p> <p>Examples in Context:</p> <p>"You should have called me if you were going to be late."</p>	<p>charge by the bank for using a credit card to obtain cash.</p> <p>Currency Conversion Fee: is a charge levied by the credit or debit card payment processor or ATM network to convert one currency to another as part of a financial transaction.</p> <p>Late fee: Amount charged for late payment of a fee for any type of credit.</p> <p>Maintenance fee: A fee for administrative services</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>"We can't cut corners when building our brand reputation."</p> <p>Raise the bar – Meaning: Set higher standards. "Our goal is to raise the bar in sustainable business practices."</p> <p>Bring to the table – Meaning: Offer or contribute something of value.</p>	<p>"He might have left his phone at home." "They must have forgotten about the meeting." "I can't have made that mistake; I double-checked my work." "You must submit the application by Friday." "We have to follow the safety guidelines at work."</p> <p>References: Recommended sources</p>	<p>provided by an entity such as a bank or a credit card issuer. Conditions: The particulars by which a financial contract is governed. Renewable Credit: A credit that allows you to repay and re-borrow variable amounts of money. Checking Account: The most basic type of bank account. Deposits typically do not earn interest, or earn minimal</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>"What unique skills do you bring to the table for this startup?"</p> <p>Go the extra mile – Meaning: Put in extra effort to achieve success.</p> <p>"If we want loyal customers, we need to go the extra mile."</p> <p>Learn the ropes – Meaning: Become familiar with how something is done,</p>	<p>Cambridge University Press. (n.d.). Modals: Must, have to, should, might. Cambridge Grammar. Retrieved from https://dictionary.cambridge.org</p> <p>British Council. (n.d.). Modals for advice, obligation, and possibility. Learn English – British Council. Retrieved from https://learnenglish.britishcouncil.org</p>	<p>interest. Funds can be withdrawn by check, debit card, or electronic transfer.</p> <p>Saving Account: An account in which the money deposited accrues interest.</p> <p>Check: Written payment order with money from an account.</p> <p>Deposit: Amount of money paid to credit institutions for safekeeping and to obtain interest.</p> <p>Discovered: Situation that</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
especially in a business context. "Before launching, you should learn the ropes of the industry."	Carter, R., & McCarthy, M. (2016). English Grammar Today: Modals for deduction and advice. Cambridge University Press. Retrieved from https://www.cambridge.org/grammartoday Grammarly. (n.d.). How to use modal verbs for advice and necessity. Grammarly Handbook. Retrieved from https://www.grammarly.com	occurs when a bank checking account has a debit balance for its holder. Available: Free balance in a checking or credit account. Withdrawal: Take money out from an account at an ATM or a bank branch. Value Date: It is when funds are posted to an account and available for immediate use. Real Guarantee: Movable and immovable	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
	Purdue Online Writing Lab (OWL). (n.d.). Modal verbs for necessity and speculation. Purdue OWL. Retrieved from https://owl.purdue.edu	property, intended to reduce non-payment risk in financial operations. Credit History: Documentation of the financial history of an individual or company that includes income, debts or pending financial commitments and judicial decisions for debts. Non-payment: Omission, intentional or not, of the amortization of one or more installments of a loan or a credit.	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>Interest: Amount charged for a loan, usually as a percentage of the total amount.</p> <p>Nominal interest of Banks and savings: It is the annual interest rate stipulated by a bank or savings bank for a financial or credit product. It is the basic price at which they lend the money, or pay it, depending on whether we request it or deposit it.</p> <p>Preferential Interest: It is the one that financial</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>institutions apply to their best clients.</p> <p>Financial Intermediaries: Banks, finance companies, insurance companies and other institutions that help finance transactions or insure against risks related to the purchase and sale of goods.</p> <p>Liquidation: Closing of a transaction and delivery to the investor of the obtained amount as profitability.</p> <p>Liquidity: liquid assets;</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>cash.</p> <p>Credit limit: Total amount that a customer can spend with a credit card.</p> <p>Payer: Person who delivers or sends the funds.</p> <p>Promissory note: a signed document containing a written promise to pay a stated sum to a specified person or the bearer at a specified date or on demand.</p> <p>Patrimony: Assets: Set of assets and rights that belong to a natural or</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>legal person.</p> <p>Grace Period: Period of time during which the payment of interest or principal on a pending loan is waived.</p> <p>Goodwill: Profit obtained by whoever sells a title, security, or well above the price paid for it.</p> <p>Principal: Original amount of a loan, not including interest. The principal amount can be reduced by amortizing an amount higher than the amortizable interest</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>on a particular date.</p> <p>Bankruptcy: A situation that occurs when a company cannot meet its obligations, as a consequence of bearing a liability greater than its assets.</p> <p>Profitability: Relationship, usually in percentage, between the return provided by the operation and what has been invested in it.</p> <p>Reserves: Part of the profit of a Company that is not intended to pay</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>dividends or taxes and that is left as the company's own resources to increase its solvency.</p> <p>Credit Insurance: Insurance that pays the outstanding debit balance in case of financial difficulties</p> <p>Initial Rate: Special interest rate applied for a specified time when opening a credit or savings account.</p> <p>Floating Rate: An interest rate that rises or falls</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>based on the base interest rate set by a central or national bank.</p> <p>Monetary Transaction: Marketing transaction in which goods or services are exchanged for money.</p> <p>Transactions: Business between two parties that involves at least two things of value, agreed conditions</p>	

Curricular Design

Subject Area: English Oriented to Software Applications Development



Grade: Tenth

CEFR : B1.1

Scenario 2: Data-Base

Theme 1: Data-Base Design

Time: 20 hours

Essential Question: How does connectivity help us improve our working, learning, and living environments?

Essential Competences: Innovation

New Citizenship Axis: Digital Citizenship with Social Equity

Table 20

Curriculum Pedagogical Design

Goals	Performance Indicator
The learners can...	The student...
Establish innovative strategies and mechanisms to respond with efficiency to the constant changes in modern working environments.	Makes an effective selection of procedures and mechanisms to satisfy the modern demands of a Global Community.



Goals	Performance Indicator
The learners can...	The student...
Engage in dynamic digital environments that facilitate the achievement of common social changes with fairness and invention.	Interacts with other citizens to obtain a determined goal using modern digital tools with responsibility and innovation.

TABLE 21

Oral and Written Comprehension

Goals	Performance Indicator
The learners can...	The student...
Listening: Understand simple technical information, such as characteristics, elements, and fundamental concepts associated with databases.	<ul style="list-style-type: none">• Recognizes specific terminology and its meaning associated with databases.• Distinguishes relevant information such as characteristics, elements, and fundamental concepts associated with databases.• Determines the most suitable and efficient examples of databases.
Reading: Search the internet, or other reliable	<ul style="list-style-type: none">• Extracts relevant details about database uses, applications, and contributions to daily work.



sources of information, for specific everyday or work-related material about databases uses, applications, and contributions to daily work.	<ul style="list-style-type: none">• Identifies key subject vocabulary in the text related to database uses and applications.• Describes the implementation of database attributes and contributions in your daily life or work environment.
---	--

Table 22

Oral and Written Production

Goals		Performance Indicator
The learners can...	The student...	
Spoken Interaction: Provide reasons and explanations, to a specific audience about elements of database design, to solve a	<ul style="list-style-type: none">• Uses clear straightforward technical vocabulary to explain elements of database design.• Describes the elements of conceptual, logical, and physical database design.• Formulates strategies to create a logical database.	



situation, using simple language.	
Spoken Production: Make a short instructional or informational text easier to understand by presenting it as a list of separate points about the elements involved in database design. Produce familiar sounds and prosodic patterns.	<ul style="list-style-type: none">• Distinguishes the characteristics of UML class diagrams for database design.• Describes database design using UML class diagram.• Integrates multiple sources of information presented in diverse formats and media (e.g., visually, quantitatively, orally) to make informed decisions and solve problems, evaluating the credibility and accuracy of each source about the methodology for applying UML to database design.• Employs a range of phonological features in the target language by manipulating prosodic features of spoken discourse (e.g. stress, intonation, rhythm) to support the message intended to convey.
Writing: Write a basic description of procedures to achieve AI databases that should care for business.	<ul style="list-style-type: none">• Traces a logical set of procedures that demonstrate that database and artificial intelligence can benefit from each other.• Writes a short summary considering grammar, discourse markers, and vocabulary studied in this unit related to the use of databases in artificial intelligence.



Table 23

Learnings of the curriculum pedagogical design

Functions and Discourse Markers	Grammar	Vocabulary	Phonology
Functions Describing database design in artificial intelligence. Describing characteristics, elements and fundamental concepts associated with databases.	•Past simple/regular /irregular Past Simple Tense Overview The Past Simple tense is used to talk about completed actions in the past. These actions happened at a specific time in the past. Time Expressions Used with the Past Simple: Yesterday, last week, last year, two days ago, in 2020, etc. "I visited Paris last summer."	Data: it is the representation of a variable, which can be qualitative or quantitative, that indicates a specific value assigned to things. It is represented by a series of symbols, numbers, or letters. File: the base files are data files that are used to save	2. Stress within the sentence. a. Words generally Stressed in Sentence: Content Words (Nouns, verbs, adjectives, adverbs, and question words).



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
Developing an argument about elements of database design for solving AI situations.	"They moved to a new house two years ago." Forming the Past Simple Regular Verbs → Add -ed to the base form of the verb. Spelling Rules for Regular Verbs: Verbs ending in -e → Add only -d (live → lived). Verbs ending in a consonant + y → Change y to i and add -ed (study → studied). Verbs with one syllable ending in a vowel + consonant → Double the final consonant and add -ed (stop → stopped).	structured contents of the data base in a single file using graphs and separated spaces. Registers: they refer to items presented in the data base. Each register corresponds to each line in a given chart. Field: in computer science a data base is divided into different columns. The columns are divided into registers. Then	b. Words generally unstressed in sentences: Function Words (articles, prepositions, pronouns, conjunctions, helping verbs). c. Stress in adjective/noun combination. Example: he sawed a blackboard.
Expressing opinions about the use of database design in AI.			
Taking initiative in interaction to describe.			
Discourse Markers			



Functions and Discourse Markers	Grammar	Vocabulary	Phonology																											
<p>Phrases used to interrupt and change topics</p> <p>Managing interaction (interrupting, changing topic)</p> <p>Examples of discourse markers used for interrupting and changing topics in conversation. These expressions help speakers manage interactions smoothly:</p>	<table><tr><th>Base Form</th><th>Past Simple</th><th>Example</th></tr><tr><td>walk</td><td>walked</td><td>"I walked to school."</td></tr><tr><td>play</td><td>played</td><td>"She played tennis yesterday."</td></tr><tr><td>study</td><td>studied</td><td>"We studied for the exam."</td></tr></table> <p>2.Irregular Verbs → These verbs do not follow a regular pattern, and their past forms must be memorized.</p> <table><tr><th>Base Form</th><th>Past Simple</th><th>Example</th></tr><tr><td>go</td><td>went</td><td>"We went to the beach."</td></tr><tr><td>have</td><td>had</td><td>"He had a meeting yesterday."</td></tr><tr><td>eat</td><td>ate</td><td>"She ate lunch at 12."</td></tr><tr><td>buy</td><td>bought</td><td>"I bought a new phone."</td></tr></table> <p>Negative Form</p> <p>Did not (didn't) + base form of the verb</p> <p>"I didn't go to the party."</p> <p>"She didn't study for the test."</p> <p>Questions</p>	Base Form	Past Simple	Example	walk	walked	"I walked to school."	play	played	"She played tennis yesterday."	study	studied	"We studied for the exam."	Base Form	Past Simple	Example	go	went	"We went to the beach."	have	had	"He had a meeting yesterday."	eat	ate	"She ate lunch at 12."	buy	bought	"I bought a new phone."	<p>each column is called a field.</p> <p>Data analysis: it is the science in charge of examining a group of data with the purpose of obtaining conclusions about the information to make decisions, or simply broaden the knowledge about a subject.</p> <p>Database: they are formed by a group of classified data according to the</p>	<p>d. Stress in compound nouns</p> <p>Example: The teacher writes on the blackboard.</p> <p>B. RHYTHM</p> <p>a. Contractions / Full form</p> <p>Example: I'll / I will</p> <p>b. Blending and Word Reductions</p> <p>Examples:"How are you?" is often</p>
Base Form	Past Simple	Example																												
walk	walked	"I walked to school."																												
play	played	"She played tennis yesterday."																												
study	studied	"We studied for the exam."																												
Base Form	Past Simple	Example																												
go	went	"We went to the beach."																												
have	had	"He had a meeting yesterday."																												
eat	ate	"She ate lunch at 12."																												
buy	bought	"I bought a new phone."																												



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>1. Interrupting politely</p> <p>Discourse markers used to interrupt the flow of conversation and take the floor.</p> <p>Examples:</p> <p>“Sorry to interrupt, but...”</p> <p>Example: “Sorry to interrupt, but can we go back to the budget issue?”</p>	<p>Did + subject + base form of the verb</p> <p>"Did you watch the movie?"</p> <p>"Did they arrive on time?"</p> <p>Examples in Context</p> <p>"I visited my grandparents last weekend." (Regular verb)</p> <p>"He went to the gym yesterday." (Irregular verb)</p> <p>"We didn't watch the football match." (Negative form)</p> <p>"Did you call the doctor?" (Question form)</p>	<p>criteria and saved in a support (digital or not) with the aim of being consulted and accessed in a simple and easy way.</p> <p>Data sources: also called data files, they are simply a collection of registers that save data. This is why any text file can be used as a data source, either plain text or a real data base.</p> <p>Types of data:</p> <p>Numerical</p>	<p>pronounced</p> <p>“howaryou”</p> <p>c. Linking sounds: is the technique for smoothly moving from one word into the next during pronunciation.</p> <p>Double consonant: often pronounced as a single consonant.</p> <p>Example: pretty-little-pillow...</p>



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>"If I may add something here..." Example: "If I may add something here, I think we should explore another option." "Before you go on..." Example: "Before you go on, can I clarify something?"</p> <p>2. Changing the topic abruptly</p> <p>Discourse markers used to introduce a new topic or steer</p>		<p>Whole numbers: this type of data is formed by a numerical variable that has no decimal part. Real numbers: this type of data is formed by a variable that can have a decimal part. Text Character: this type of data is formed by a unit or symbol that can be a letter, number, capital letter or punctuation mark. Chain: this type of data is formed by a</p>	<p>Phrasing and Pausing:</p> <p>Phrase: a group of words that convey meaning.</p> <p>Pause: a brief moment of silence to emphasize meaning.</p>



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>the conversation in a different direction. Examples:</p> <p>“Speaking of...” Example: “Speaking of holidays, have you decided where you’re traveling this summer?” “That reminds me...” Example: “That reminds me, we need to order more supplies.”</p>		<p>group of characters assemble in consecutive order and it is represented by quotations.</p> <p>Logic</p> <p>Boolean data: this type of data can represent one of two possible values (usually denoted true and false) which is intended to characterize the two truth values of logic and Boolean algebra.</p> <p>Entity: an entity is an</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>"By the way..."</p> <p>Example: "By the way, have you heard the latest update from the team?"</p> <p>"On a different note..."</p> <p>Example: "On a different note, how is your new project going?"</p> <p>3. Redirecting the conversation back</p> <p>Discourse markers used to return to</p>		<p>object that exists in the real world. It can also be an abstract concept. It is distinguishable from all of the other objects. An entity contains a series of properties or features that characterize it.</p> <p>Attributes: attributes are the describing characteristics or properties that define all items pertaining to a certain category</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>the original topic after a digression. Examples:</p> <p>“Anyway...” Example: “Anyway, as I was saying, the deadline is next Friday.” “Where were we?” Example: “Where were we? Oh yes, the marketing strategy.” “To get back on track...”</p>		<p>applied to all cells of a column.</p> <p>Data Management Systems: data management systems (DBMS) are software that create and manage data bases. The DBMS provide users and programmers with a systematic way of administrating data. A DBMS also allows the final users to create, read, update, and</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Example: "To get back on track, let's focus on the action plan."</p> <p>Idioms and common colloquial expressions</p> <p>8 idioms and common colloquial expressions related to Database Design that could be used in professional discussions, training sessions, or casual conversation:</p>		<p>eliminate data from a data base.</p> <p>Data modeling: it is a way of structuring and organizing data to be used in an easy way by data bases.</p> <p>Conceptual design: it refers to the description of information content from a data base and not to the structures of storage that will be needed to manage it. Its goal is to design a database that is independent of</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>1. Garbage In, Garbage Out (GIGO)</p> <p>Meaning: Bad input will always result in bad output. Example: "If you don't clean up the data before importing it, remember— garbage in, garbage out!"</p> <p>2. Hit the nail on the head</p>		<p>database software and physical details. Logical Design: is the process of transforming (or mapping) a conceptual schema of the application domain into a schema for the data model underlying a particular DBMS. Physical Design: it optimizes performance while ensuring data integrity by avoiding unnecessary data redundancies. During</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Meaning: To be exactly right or accurate. Example: "Your explanation of the primary key hit the nail on the head."</p> <p>3. Back to the drawing board Meaning: Start again from the beginning due to a failure. Example: "The database schema didn't scale well; it's</p>		<p>physical design, you transform the entities into tables, the instances into rows, and the attributes into columns.</p> <p>UML class diagrams: they are diagrams of unified language or UML (unified modeling language). This diagram is a graphical notation used to construct and visualize object-oriented systems.</p> <p>Database</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>back to the drawing board."</p> <p>4. In the same boat Meaning: Facing the same problem or challenge as others. Example: "We're all in the same boat when it comes to optimizing query performance."</p> <p>5. The devil is in the details</p>		<p>normalization: it is the process of organization of data in a data base. Normalization entails organizing the columns (attributes) and tables (relations) of a database to ensure that their dependencies are properly enforced by database integrity constraints.</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Meaning: Small details can cause problems if overlooked. Example: "The database migration seemed easy at first, but the devil is in the details."</p> <p>6. A double-edged sword</p> <p>Meaning: Something that has both advantages and disadvantages.</p>			



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Example: "Using automated indexing is a double-edged sword—great for speed, but it can bloat the database."</p> <p>7. Break the system Meaning: Cause a failure by overwhelming or misusing a system. Example: "If you run that query without</p>			



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>filtering, it'll break the system."</p> <p>8. Up and running</p> <p>Meaning: Something that is operational and functioning properly.</p> <p>Example: "We'll have the new database up and running by the end of the week."</p>			



Curricular Design

Subject Area: English Oriented to Software Applications Development

Grade: Eleventh

CEFR : B1.1

Scenario 2: Data-Base

Theme 2: Relational Data-Base

Time: 20 hours

Essential Question: How do relational databases over any other type of databases help maintain data integrity, accuracy, and flexibility?

Essential Competences: Teamwork

New Citizenship Axis: Strengthening of Planetary Citizenship with Identify

Table 24

Curriculum Pedagogical Design



Goals	Performance Indicator
The learners can...	The student...
Show willingness to work collaboratively to achieve common goals.	<ul style="list-style-type: none">• Follows common objectives depending on the activities that take place in your classroom activities.• Expresses the relationships between collaboration and trustfulness during the development of classroom activities.• Demonstrates diversity and respect for different nationalities, genders, cultures, interests, races, religions, opinions, beliefs and abilities.
Determines how relational databases become a supreme competitive advantage for Artificial Intelligence.	<ul style="list-style-type: none">• Recognizes the importance of relational databases for artificial intelligence environment.• Gives examples of relational databases and their correlation with artificial intelligence.• Explains how databases in Artificial Intelligence contribute to the strengthening of the identity

TABLE 25



Oral and Written Comprehension

Goals		Performance Indicator	
The learners can...		The student...	
Listening: Understand the use of information content of the majority of recorded audio material about relational databases for solving specific challenges delivered in clear standard speech.		<ul style="list-style-type: none">• Identifies relational database language.• Defines the use of relational database language. (SQL Structured Query Language)• Distinguishes Relational Database Management System advantages.	
Reading: Read newspapers/magazine accounts of films, and books, written for a wider audience and understand the main points regarding the		<ul style="list-style-type: none">• Identifies the pros and cons of relational databases.• Recognizes examples of relational database tables.• Evaluates critically how relational databases are used for solving challenges in artificial intelligence.	



examples and uses of
relational databases.

Table 26

Oral and Written Production

Goals		Performance Indicator
The learners can...	The student...	
Spoken Interaction: Follow what is said about the use of relational databases with the help of technological devices, though he/she may occasionally have to ask for repetition or clarification if the other people's talk is rapid or extended.	<ul style="list-style-type: none">• Identifies the technology used to build relational databases.• Discusses the types of relational databases using simple language, grammar accuracy, and good pronunciation of new vocabulary.• Carries out a simple informal interview to know whether relational database technology could be helpful in artificial intelligence.	
Spoken Production: Give simple reasons to	<ul style="list-style-type: none">• Gives a simple update on the use of relational databases in artificial intelligence.	



<p>justify a viewpoint of the applications of relational databases.</p> <p>-Produce unfamiliar sounds and prosodic patterns.</p>	<ul style="list-style-type: none">• Suggests possible uses for grouping in relational databases.• Justify a simple point of view on the different types of grouping in relational databases.• Employs a range of phonological features in the target language by manipulating prosodic features of spoken discourse(e.g. stress, intonation, rhythm) to support the message intended to convey.
<p>Writing: Write a brief standard report conveying factual</p>	<p>Proposes ideas with relevant examples that facilitate the understanding of the effectiveness, efficiency, and disadvantages of relational databases.</p>



information, stating the benefits and disadvantages of relational databases.

Table 27

Learnings of the curriculum pedagogical design

Functions and Discourse Markers	Grammar	Vocabulary	Phonology
Functions Describing the applications of relational databases. Using examples of relational databases for problem-solving in a company related to artificial intelligence.	Past continuous Past Continuous Overview The Past Continuous (also called the Past Progressive) is used to describe actions that were in progress at a	A Relational Database is a type of database that organizes data into tables (relations) consisting of rows (records) and columns (fields). These tables can be related to each other through keys (primary	2. Stress within the sentence. a. Words generally Stressed in Sentence: Content Words (Nouns, verbs, adjectives, adverbs, and question words).



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Understanding advantages and disadvantages of relational databases.</p> <p>Evaluating the effectiveness and efficiency of relational databases.</p> <p>Discourse Markers</p> <p>Phrases used to interrupt and change topics *(see the example above)</p>	<p>specific time in the past or background actions in a story.</p> <p>Forming the Past Continuous</p> <p>Subject + was/were + verb (-ing)</p> <p>Positive Form</p> <p>I was studying at 8 p.m.</p> <p>They were playing football in the park.</p> <p>Negative Form</p> <p>Subject + was/were + not + verb (-ing)</p>	<p>and foreign keys). It is widely used for managing structured data because it ensures data consistency, integrity, and ease of access.</p> <p>Main Features</p> <p>Data Integrity: Ensures accuracy and consistency of data.</p> <p>Relationships: Defines connections between tables (e.g., one-to-one, one-to-many).</p> <p>SQL (Structured Query</p>	<p>b. Words generally unstressed in sentences:</p> <p>Function Words (articles, prepositions, pronouns, conjunctions, helping verbs).</p> <p>c. Stress in adjective/noun combination.</p> <p>Example: he sawed a blackboard.</p> <p>d. Stress in compound nouns</p>



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Managing interaction (interrupting, changing topic) examples of discourse markers used for interrupting and changing topics in conversation. These expressions help speakers manage interactions smoothly:</p> <p>1. Interrupting politely Discourse markers are used to interrupt the flow of conversation and take the floor.</p>	<p>I wasn't watching TV. We weren't working yesterday afternoon.</p> <p>Questions Was/Were + subject + verb (-ing) Was she reading a book at 9 p.m.? What were you doing when the phone rang?</p> <p>Uses of the Past Continuous</p>	<p>Language): The standard language for interacting with relational databases. Use Cases Banking and finance: Transaction management and customer data. Education: Student and course management systems. E-commerce: Managing orders, products, and customer data. Healthcare: Patient records and hospital</p>	<p>Example: The teacher writes on the blackboard.</p> <p>B. RHYTHM a. Contractions / Full form Example: I'll / I will b. Blending and Word Reductions Examples: "How are you?" is often pronounced "howaryou" c. Linking sounds: is the technique for smoothly</p>



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
Examples: “Sorry to interrupt, but...” Example: “Sorry to interrupt, but can we go back to the budget issue?” “If I may add something here...” Example: “If I may add something here, I think we should explore another option.” “Before you go on...” Example: “Before you go on, can I clarify something?”	Action in Progress at a Specific Time in the Past "At 10 p.m., I was sleeping." (The action was ongoing at that time.) Two Actions Happening at the Same Time in the Past "While I was cooking, she was setting the table." (Both actions were happening simultaneously.)	management systems. Aggregate Function – A function that performs a calculation on a set of values and returns a single value (e.g., SUM, AVG). Alias – An alternate name given to a table or column for temporary use in queries. ALTER – A SQL command used to modify an existing database structure, such as adding or deleting a column.	moving from one word into the next during pronunciation. Double consonant: often pronounced as a single consonant. Example: pretty-little-pillow... Phrasing and Pausing: Phrase: a group of words that convey meaning. Pause: a brief moment of silence to emphasize meaning.



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
2. Changing the topic abruptly Discourse markers used to introduce a new topic or steer the conversation in a different direction. Examples: “Speaking of...” Example: “Speaking of holidays, have you decided where you’re traveling this summer?” “That reminds me...” Example: “That reminds me, we need to order more supplies.”	Interrupted Action (Past Continuous + Past Simple) "I was walking home when it started to rain." (An ongoing action is interrupted by a sudden event.) Describing Background in a Story "The sun was shining, and the birds were singing." (Background description in a narrative.)	Attribute – A column in a table that represents a data field. Backup – A copy of database data used for recovery in case of failure. Cardinality – The uniqueness of values in a column or the relationship between two tables. CHECK Constraint – A rule to enforce domain integrity by limiting the values that can be placed in a column.	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>"By the way..."</p> <p>Example: "By the way, have you heard the latest update from the team?"</p> <p>"On a different note..."</p> <p>Example: "On a different note, how is your new project going?"</p> <p>3. Redirecting the conversation back</p> <p>Discourse markers used to return to the original topic after a digression.</p> <p>Examples:</p>	<p>Examples in Context</p> <p>"We were watching a movie when the lights went out."</p> <p>"He was studying all afternoon."</p> <p>"I wasn't listening when the teacher explained the homework."</p> <p>"Were they working on the project yesterday?"</p> <p>"While she was reading, the dog started barking."</p> <p>References:</p> <p>Recommended sources</p>	<p>Clustered Index – A type of index that sorts and stores data rows in the table based on their key values.</p> <p>Column – A vertical set of data values in a table, representing a data field.</p> <p>Commit – A SQL command that saves all changes made to the database during a transaction.</p> <p>Composite Key – A key that consists of two or more columns used to uniquely identify a row.</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>“Anyway...”</p> <p>Example: “Anyway, as I was saying, the deadline is next Friday.”</p> <p>“Where were we?”</p> <p>Example: “Where were we? Oh yes, the marketing strategy.”</p> <p>“To get back on track...”</p> <p>Example: “To get back on track, let’s focus on the action plan.”</p> <p>Idioms and common colloquial expressions</p>	<p>Cambridge University Press. (n.d.). Past continuous tense. Cambridge Grammar. Retrieved from https://dictionary.cambridge.org</p> <p>British Council. (n.d.). Past continuous tense. Learn English – British Council. Retrieved from https://learnenglish.britishcouncil.org</p> <p>Azar, B. S., & Hagen, S. A. (2009). Understanding and using English</p>	<p>Constraint – A rule enforced on data in a table to ensure data integrity.</p> <p>CRUD – An acronym for Create, Read, Update, Delete; basic database operations.</p> <p>Data Definition Language (DDL) – A subset of SQL used to define and modify database structures (e.g., CREATE, ALTER, DROP).</p> <p>Data Manipulation Language (DML) – A subset of SQL used for</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>1. Join the dots Meaning: To connect pieces of information to understand the whole picture. Example: "When designing relational tables, you need to join the dots between related data to avoid redundancy."</p> <p>2. Table the discussion Meaning: To postpone or delay a discussion or decision. Example: "Let's table the discussion on</p>	<p>grammar (4th ed.). Pearson Education. Retrieved from https://www.pearson.com Purdue Online Writing Lab (OWL). (n.d.). Past continuous tense. Purdue OWL. Retrieved from https://owl.purdue.edu Oxford University Press. (n.d.). Past continuous tense. Oxford English Grammar Course.</p>	<p>data manipulation (e.g., SELECT, INSERT, UPDATE, DELETE). Database – An organized collection of data stored and accessed electronically. Database Schema – The structure that defines the organization of data in a database. Deadlock – A situation where two or more transactions are waiting for each other to release resources, preventing progress.</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>normalization until after we finalize the schema."</p> <p>3. The missing link</p> <p>Meaning: A crucial piece of information or connection that is missing.</p> <p>Example: "The foreign key was the missing link between these two tables."</p> <p>4. On the same page</p> <p>Meaning: To have a shared understanding of something.</p> <p>Example: "Before we proceed with the ER</p>	<p>Retrieved from https://elt.oup.com</p>	<p>DELETE – A SQL command used to remove rows from a table.</p> <p>Entity – An object or concept about which data is stored in a database.</p> <p>Entity Relationship Diagram (ERD) – A diagram that shows the relationships between entities in a database.</p> <p>Foreign Key – A column in a table that creates a relationship with a primary key in another table.</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>diagram, let's ensure we're on the same page about the data relationships."</p> <p>5. Built from the ground up</p> <p>Meaning: To create something from the very beginning.</p> <p>Example: "We built this relational database from the ground up to handle complex queries."</p> <p>6. Data dump</p>		<p>Index – A database object used to speed up data retrieval.</p> <p>JOIN – A SQL operation that combines rows from two or more tables based on a related column.</p> <p>Normalization – The process of organizing data to reduce redundancy and improve data integrity.</p> <p>Null – A special marker used in databases to indicate that a value does not exist.</p> <p>Primary Key – A column</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Meaning: A large amount of information given at once, often without filtering. Example: "Don't just do a data dump into the table—make sure it's properly structured."</p> <p>7. Out of sync Meaning: Not aligned or in agreement. Example: "The data in the backup server is out of sync with the production database."</p> <p>8. Keep it in check</p>		<p>or combination of columns that uniquely identifies a row in a table.</p> <p>Query – A request for data or information from a database.</p> <p>RDBMS (Relational Database Management System) – Software for managing relational databases.</p> <p>Referential Integrity – A property that ensures relationships between tables remain consistent.</p> <p>ROLLBACK – A SQL command that undoes</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
Meaning: To control or limit something. Example: "Indexes can speed up queries, but you need to keep them in check to avoid performance issues."		changes made during a transaction. SELECT – A SQL command used to retrieve data from one or more tables. Stored Procedure – A set of SQL statements that can be executed as a program. Table – A collection of related data in rows and columns. Transaction – A sequence of operations performed as a single logical unit of work.	



Curricular Design

Subject Area: English-Oriented to Software Applications Development

Grade: Eleventh

CEFR : B1.2

Scenario 2: Data-Base

Theme 3: Non-Relational Database

Time: 20 hours

Essential Question: How can a non-relational database transform your businesses?

Essential Competences: Empowerment



New Citizenship Axis: Strengthening of Planetary Citizenship with Identify

Table 28

Curriculum Pedagogical Design

Goals	Performance Indicator
The learners can...	The student...
Implement decision-making with empowerment for different processes that require technical skills in database.	<ul style="list-style-type: none">• Identifies the concepts of empowerment and decision-making.• Describes the characteristics of the types of decisions best suited for using the non-relational database.• Empowers team members to practice decision-making techniques for non-relational databases.
Develop skills as app developers for a healthy coexistence in the world respecting human rights.	<ul style="list-style-type: none">• Distinguishes multicultural characteristics, responsibility, and peace.• Describes responsibilities and skills for living in the world.• Performs sociolinguistic skills that consolidate peace and personal and social responsibility.



TABLE 29

Oral and Written Comprehension

Goals		Performance Indicator
The learners can...		The student...
Listening:	Follow a straightforward presentation or demonstration about the use of non-relational databases.	<ul style="list-style-type: none">• Distinguishes basic concepts of non-relational databases (NoSQL) and application development.• Identifies the characteristics of non-relational databases by gathering information presented through media.• Paraphrases the benefits of a non-relational database in contrast with relational databases.
Reading:	Find and understand relevant information about the use of non-relational databases.	<ul style="list-style-type: none">• Interprets the way non-relational databases work.• Mentions common examples of non-relational databases.• Distinguishes the uses for non-relational databases in an enterprise to create business agility.

Table 11

Oral and Written Production

Goals	Performance Indicator
The learners can...	The student...



<p>Spoken Interaction: Start up a conversation and help it to keep going by asking people relatively spontaneous questions about non-relational databases as widely recognized for building hi-tech applications.</p>	<ul style="list-style-type: none">• Starts a conversation explaining the concept and characteristics of non-relational databases (NoSQL)• Talks about the impact of NoSQL as leading new data storage technology.• Participates in comparing the advantages of NoSQL over relational databases.
<p>Spoken Production: Explain the main points and crucial aspects to take into account when using a non-relational database. Produce sounds and prosodic patterns.</p>	<ul style="list-style-type: none">• Describes the reasons that makes NoSQL preferred for eCommerce, retail, entertainment, and other business applications.• Explains the way a non-relational database is used as a data-driven enterprise solution in AI.• Uses techniques to decide when to choose NoSQL vs. Relational Databases.
<p>Writing: Present a topic in a short report or poster,</p>	<p>Writes about non-relational databases by presenting a booklet using photographs and short blocks in order to provide feedback and offer</p>



using photographs and
short blocks of text about
non-relational
databases.

suggestions.

Table 31

Learnings of the curriculum pedagogical design

Functions and Discourse Markers	Grammar	Vocabulary	Phonology
Functions Describing non- relational databases. Describing the characteristics of NoSQL.	Phrasal Verbs Overview Phrasal verbs are combinations of a verb + particle (preposition or adverb) that create a new meaning different from the original verb. They are commonly used	NoSQL: NoSQL databases (aka "not only Structure Query Language") are non-tabular databases and store data differently than relational tables. NoSQL databases come in a variety of types based on their data	2. Stress within the sentence. a. Words generally Stressed in Sentence: Content Words (Nouns, verbs, adjectives, adverbs, and question words).



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
Identifying the common use of NoSQL. Discourse Markers Phrases used to interrupt and change topics *(see the example above) Managing interaction (interrupting, changing topic) examples of discourse markers used for interrupting and changing topics in	in spoken and informal English. Example: "Turn on" → means to start or activate something. "Please turn on the lights." "Give up" → means to stop doing something. "He gave up smoking last year." Types of Phrasal Verbs	model. The main types are document, key- value, wide-column, and graph. They provide flexible schemas and scale easily with large amounts of data and high user loads. JSON: it is an open standard file format and data interchange format that uses human- readable text to store and transmit data objects consisting of attribute–value pairs and arrays (or other	b. Words generally unstressed in sentences: Function Words (articles, prepositions, pronouns, conjunctions, helping verbs). c. Stress in adjective/noun combination. Example: he sawed a blackboard. d. Stress in compound nouns



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>conversation. These expressions help speakers manage interactions smoothly:</p> <p>1. Interrupting politely</p> <p>Discourse markers are used to interrupt the flow of conversation and take the floor.</p> <p>Examples:</p> <p>“Sorry to interrupt, but...”</p> <p>Example: “Sorry to interrupt, but can we</p>	<p>Transitive (with an object)</p> <p>These phrasal verbs need an object to complete their meaning.</p> <p>"I will look after my little sister tonight."</p> <p>"He turned off the TV."</p> <p>Intransitive (without an object)</p> <p>These do not take an object.</p>	<p>serializable values). It is a common data format with a diverse range of functionality in data interchange including communication of web applications with servers. JSON is a language-independent data format. It was derived from JavaScript, but many modern programming languages include code to generate and parse JSON-format data. JSON filenames use the</p>	<p>Example: The teacher writes on the blackboard.</p> <p>B. RHYTHM</p> <p>a. Contractions / Full form</p> <p>Example: I'll / I will</p> <p>b. Blending and Word Reductions</p> <p>Examples: “How are you?” is often pronounced “howaryou”</p> <p>c. Linking sounds: is the technique for smoothly moving from one word into</p>



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
go back to the budget issue?"	"We need to catch up soon."	extension. BSON: it is a computer data interchange format.	the next during pronunciation.
"If I may add something here..."	"The plane took off at 8 a.m."	The name "BSON" is based on the term JSON and stands for "Binary JSON". It is a binary form for representing simple or complex data structures including associative arrays (also known as name-value pairs), integer indexed arrays, and a suite of fundamental scalar types.	Double consonant: often pronounced as a single consonant.
Example: "If I may add something here, I think we should explore another option."	Separable Phrasal Verbs The object can be placed between the verb and the particle.		Example: pretty-little- pillow...
"Before you go on..."	"Turn off the radio." → "Turn it off."		Phrasing and Pausing:
Example: "Before you go on, can I clarify something?"	Inseparable Phrasal Verbs		Phrase: a group of words that convey meaning.
2. Changing the topic abruptly		Collections: it refers to a	Pause: a brief moment of silence to emphasize meaning.



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Discourse markers used to introduce a new topic or steer the conversation in a different direction.</p> <p>Examples:</p> <p>“Speaking of...”</p> <p>Example: “Speaking of holidays, have you decided where you’re traveling this summer?”</p> <p>“That reminds me...”</p>	<p>The verb and particle cannot be separated.</p> <p>“We are looking for a new house.” (NOT: “We are looking a new house for.”)</p> <p>Examples in Context</p> <p>“I gave up trying to fix the car.”</p> <p>“We ran out of time during the exam.”</p>	<p>group of documents. A collection is the equivalent of a RDBMS chart. There is a collection inside a unique data base.</p> <p>Document- oriented database: A document-oriented database, or document store, is a computer program and data storage system designed for storing, retrieving, and managing document-oriented information, also known as semi-structured data.</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
Example: "That reminds me, we need to order more supplies." "By the way..." Example: "By the way, have you heard the latest update from the team?" "On a different note..." Example: "On a different note, how is your new project going?"	"She turned on the computer and started working." "Can you look after my dog while I'm away?" "He picked up his phone and left the room." Tips for Learning Phrasal Verbs Learn them in context, not as single words. Practice using them in sentences.	Key-value database: A key-value database, or key-value store, is a data storage paradigm designed for storing, retrieving, and managing associative arrays, and a data structure more commonly known today as a dictionary or hash table. Dictionaries contain a collection of objects, or records, which in turn have many different fields within them, each containing data. These	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>3. Redirecting the conversation back</p> <p>Discourse markers are used to return to the original topic after a digression.</p> <p>Examples:</p> <p>“Anyway...”</p> <p>Example: “Anyway, as I was saying, the deadline is next Friday.”</p> <p>“Where were we?”</p>	<p>Group them by meaning or theme (e.g., travel, work, daily activities).</p> <p>Cambridge University Press. (n.d.). Phrasal verbs. Cambridge Dictionary. Retrieved from https://dictionary.cambridge.org</p> <p>British Council. (n.d.). Phrasal verbs: Meaning and use. Learn English – British Council. Retrieved from</p>	<p>records are stored and retrieved using a key that uniquely identifies the record and is used to find the data within the database. It is a simple method to store data.</p> <p>Document-oriented database: A document-oriented database, or document store, is a computer program and data storage system designed for storing, retrieving and managing document-oriented information, also known</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Example: "Where were we? Oh yes, the marketing strategy."</p> <p>"To get back on track..."</p> <p>Example: "To get back on track, let's focus on the action plan."</p> <p>Idioms and common colloquial expressions</p> <p>8 idioms and colloquial expressions related to Non-Relational Databases (NoSQL) that help convey</p>	<p>https://learnenglish.britishcouncil.org</p> <p>McCarthy, M., & O'Dell, F. (2017). English Phrasal Verbs in Use: Advanced (2nd ed.). Cambridge University Press. Retrieved from https://www.cambridge.org</p> <p>Oxford University Press. (n.d.). Phrasal verbs: A guide to meaning and use. Oxford Advanced Learner's Dictionary. Retrieved from</p>	<p>as semi-structured data.</p> <p>It is one of the main categories of NoSQL databases, a document-oriented system relies on internal structure in the document in order to extract metadata that the database engine uses for further optimization. Document-store is designed to offer a richer experience with modern programming techniques.</p> <p>Authentication: it is the process of identification</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>concepts in a more engaging way:</p> <p>"Think outside the box"</p> <p>– Refers to how non-relational databases break away from the traditional tabular structure of relational databases and offer flexible data storage solutions.</p> <p>"To handle unstructured data, you need to think outside the box and</p>	<p>https://www.oxfordlearnersdictionaries.com</p> <p>Purdue Online Writing Lab (OWL). (n.d.). Using phrasal verbs in English. Purdue OWL. Retrieved from https://owl.purdue.edu</p>	<p>of an individual regarding his/her personal information (username and password), and authorization (access control) which is the process of deciding whether or not the user has permission to execute an action or not. Obtaining the collections list: this means to obtain the collections list from different documents JSON, each one with a unique password inside</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>adopt a NoSQL approach."</p> <p>"One size doesn't fit all"</p> <p>– Emphasizes that relational databases aren't always the best solution, and sometimes a non-relational approach is more suitable.</p> <p>"When dealing with real-time data streams, remember—one size doesn't fit all."</p> <p>"Data in the wild" –</p> <p>Refers to unstructured</p>		<p>the collection.</p> <p>The usage of a pointer to obtain all types of documents.</p> <p>Obtaining a document with a search:</p> <p>Creation of an index:</p> <p>they are auxiliary structures that allow the database to have Access to certain documents in a faster way. They represent an organized copy of certain keys in a collection. This copy includes the organized</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>or semi-structured data that doesn't fit neatly into rows and columns.</p> <p>"NoSQL databases are perfect for managing data in the wild, like social media feeds or IoT data."</p> <p>"Hit the ground running"</p> <p>– Describes how NoSQL databases can be deployed and scaled quickly without extensive setup.</p>		<p>keys, an index (pointer) to the documents that contains them.</p> <p>Obtaining a list of indexes in a collection:</p> <p>this allows to improve the efficiency in many operations CRUD, avoiding the complete tours of the collections.</p> <p>Creating a database: it allows the creation of a design for several data access patterns that include low latency applications.</p> <p>Creating collections: a</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>"With a NoSQL database, you can hit the ground running and scale as you grow."</p> <p>"Flying under the radar" – Refers to lesser-known or niche NoSQL databases used for specific use cases.</p> <p>"Graph databases were flying under the radar for years, but now they're in high demand for</p>		<p>collection is very similar to a database chart. The chart stores registers (rows) and the collections stores documents.</p> <p>Creating documents: a document-oriented database is constituted by a group of programs that store, recover and manage data from documents or a structured mode. These types of databases are one of the main subcategories inside the</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>relationship-based data."</p> <p>"Go with the flow" – Represents the flexibility of NoSQL databases in handling data schema changes.</p> <p>"When your data model evolves constantly, NoSQL lets you go with the flow."</p> <p>"Bigger fish to fry" – Indicates focusing on more critical tasks, like data analysis, rather than managing</p>		<p>NoSQL databases.</p> <p>CRUD: CRUD is an acronym that comes from the world of computer programming and refers to the four functions that are considered necessary to implement a persistent storage application: create, read, update, and delete.</p> <p>Query: it recovers data from a defined collection or document.</p> <p>Search: it is a concrete search method in NoSQL</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>complex relational schemas.</p> <p>"Why worry about schema migrations when you have bigger fish to fry?"</p> <p>"In real-time" – Used to emphasize the real-time processing capabilities of certain NoSQL databases, such as for live chat or analytics.</p> <p>"Non-relational" databases shine when</p>		<p>databases.</p> <p>Simple search: it is a concrete search method in databases that uses a criterion.</p> <p>Advanced Search: it is a concrete search method in databases that uses more than one criterion. Operators are the conditions that can be added to a search. They can be classified as comparison, element, and arrangement, among other categories.</p> <p>Update Operators: they</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
you need to process data in real-time."		<p>are operators of modification in a database.</p> <p>Benefits of a Non- relational database</p> <ul style="list-style-type: none">-Massive database organization-Flexible database expansion-Multiple data structures-Built for the cloud <p>NoSQL databases have the following characteristics:</p> <ul style="list-style-type: none">-Non-relational.-NoSQL is schemaless.-Most implement an	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>aggregate pattern.</p> <p>-It is running well on clusters.</p> <p>-Open source/fully managed cloud.</p> <p>-Have high scalability.</p> <p>-Use distributed computing.</p> <p>-Cost-effective.</p> <p>-Offers easy-to-use interfaces for storage and querying of provided data.</p> <p>-Able to process both unstructured and semi-structured data.</p> <p>-No complex</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		relationships.	

Curricular Design

Subject Area: English Oriented to Software Applications Development

Grade: Eleventh

CEFR : B1.2

Scenario 3: Software Development

Theme 1: Agile and Quality Development

Time: 20 hours

Essential Question: How can Agile methodologies and quality development principles enhance problem-solving in software and product development?



Essential Competences: Problem solving

New Citizenship Axis: Sustainable Development Education

Table 32

Curriculum Pedagogical Design

Goals	Performance Indicator
The learners can...	The student...
Develop the ability to apply problem-solving techniques in Agile software development by identifying quality issues, analyzing root causes, and implementing continuous improvement strategies to enhance product quality and team performance.	<ul style="list-style-type: none">Identifies and explains common quality issues in Agile software development using clear and structured language, providing simple examples and possible causes.Describes step-by-step problem-solving techniques in Agile, using basic technical vocabulary to suggest improvements and contribute to team discussions on continuous improvement strategies.



Goals	Performance Indicator
The learners can...	The student...
Understand and apply sustainable development principles in Agile software development by promoting resource-efficient practices, fostering collaboration, and ensuring the delivery of high-quality software that supports long-term social, economic, and environmental sustainability.	<ul style="list-style-type: none">Explains how Agile software development can support sustainability by promoting resource efficiency, teamwork, and long-term quality.Describes basic sustainable development practices in Agile projects using clear and structured language, providing examples of how they contribute to environmental, social, and economic sustainability.

TABLE 33

Oral and Written Comprehension

Goals	Performance Indicator
The learners can...	The student...
Listening: -Understand the main ideas and key points in discussions and presentations on Agile	<ul style="list-style-type: none">Identifies key concepts and main ideas in spoken presentations on Agile practices and quality assurance, demonstrating understanding by summarizing key points in simple terms.



<p>practices and quality assurance in software development when clear, standard language is used.</p> <p>-Follow detailed explanations and instructions for Agile workflows and quality improvement processes, identifying important information even in moderately complex contexts.</p>	<ul style="list-style-type: none">• Recognizes commonly used terms and phrases in Agile discussions and can follow the general flow of conversation, asking for clarification when needed.• Extracts key steps and essential details from spoken explanations about Agile workflows and quality improvement, demonstrating understanding by restating instructions or applying them in practice.• Distinguishes between main points and supporting details in spoken descriptions of Agile processes, identifying relevant information to complete a given task or solve a problem.
<p>Reading: - Identify and understand key information in texts related to Agile methodologies and</p>	<ul style="list-style-type: none">• Recognizes and extracts essential information from Agile-related texts, such as key terms, main ideas, and important details, demonstrating understanding by summarizing or answering simple questions.



quality assurance, such as guides, reports, and process documentation.	
-Interpret detailed descriptions of Agile practices and quality improvement strategies, recognizing relevant concepts and instructions within moderately complex written materials.	<ul style="list-style-type: none">• Understands step-by-step explanations of Agile practices and quality strategies, identifying key concepts and instructions and applying them to a given task or scenario.

Table 34

Oral and Written Production

Goals		Performance Indicator
The learners can...	The student...	
Spoken Interaction:		
-Participate in discussions about Agile	<ul style="list-style-type: none">• Expresses opinions and asks relevant questions in Agile discussions using simple but structured language, demonstrating understanding by	



<p>practices and quality improvement, expressing opinions, asking for clarification, and responding to others clearly and coherently.</p> <p>-Engage in team conversations on Agile processes and quality issues, using appropriate vocabulary to collaborate, share ideas, and solve problems in familiar work-related situations.</p>	<p>responding appropriately to others.</p> <ul style="list-style-type: none">• Uses common Agile-related terms to contribute to team discussions, offering suggestions, and responding to teammates clearly and cooperatively.
<p>Spoken Production:</p> <p>-Explain Agile concepts and quality</p>	<ul style="list-style-type: none">• Describes Agile concepts and quality management practices using simple, structured sentences and relevant terminology, ensuring clarity



<p>management practices in software development clearly and in an organized way, using appropriate terminology.</p>	<p>and logical flow.</p>
<p>-Prepare and deliver short presentations or reports on Agile processes and quality improvement strategies, highlighting key points and providing relevant examples.</p> <p>-Produce sounds and prosodic patterns.</p>	<ul style="list-style-type: none">• Presents key aspects of Agile processes and quality improvement strategies in a clear and structured manner, using basic technical vocabulary and supporting ideas with simple examples.• Speaks with generally clear pronunciation, rhythm, and intonation, making it easy to follow, even if some minor mispronunciations occur.
<p>Writing: Write clear and structured texts, such as</p>	<p>Writes clear and structured reports or summaries on Agile practices and quality assurance processes, using appropriate vocabulary and simple linking words to</p>



reports or summaries, on Agile practices and quality assurance processes, using appropriate vocabulary and linking ideas coherently.	connect ideas logically and coherently.
--	---

Table 35

Learnings of the curriculum pedagogical design

Functions and Discourse Markers	Grammar	Vocabulary	Phonology
Functions Learning how to work collaboratively with developers, testers, and other stakeholders to ensure quality is	Past perfect Past Perfect Overview The Past Perfect is used to describe an action that happened before	Agile Software Acceptance Criteria: Conditions that a feature must meet to be considered complete. Agile Development:	C.INTONATION 1. Phrases ending with a Falling Pitch(↘) •Statements



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>maintained throughout the Agile development cycle.</p> <p>Developing the ability to recognize quality concerns early in the Agile process, such as through code reviews, automated testing, and user feedback, and applying strategies to resolve them.</p> <p>Acquire skills to clearly communicate progress, quality metrics, and potential issues to team members and</p>	<p>another action or time in the past. It shows the sequence of past events.</p> <p>Time Expressions Used with the Past Perfect:</p> <p>before, after, already, by the time, when</p> <p>"I had finished my homework before the movie started."</p> <p>"By the time we arrived, they had left."</p> <p>Forming the Past Perfect</p>	<p>Iterative and flexible methodology for software creation, focused on collaboration and continuous value delivery.</p> <p>Adaptive Evolution: The team's ability to adjust scope and priorities based on changing project needs.</p> <p>Automation Testing: Use of scripts and tools to continuously and quickly test software.</p> <p>Backlog Refinement: Ongoing review and</p>	<p>Nice to meet \you.</p> <p>I'll be back in a \minute.</p> <p>•Commands</p> <p>Write your name \here.</p> <p>Leave it on the \desk.</p> <p>•Wh- questions (requesting information.)</p> <p>What country do you come \from?</p> <p>Where do you \work?</p>



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>stakeholders, using structured reports, presentations, and Agile-specific documentation like sprint reviews or retrospectives.</p> <p>Discourse Markers Phrases used to give opinions</p> <p>Contrast On the contrary, contrarily, notwithstanding, but, however, nevertheless, in spite of, in contrast,</p>	<p>Subject + had + past participle (V3)</p> <p>Positive Form</p> <p>"She had eaten breakfast before going to school."</p> <p>"We had seen the movie before it was released on DVD."</p> <p>Negative Form</p> <p>Subject + had not (hadn't) + past participle</p> <p>"I hadn't finished my project by the deadline."</p>	<p>adjustment of the backlog to ensure tasks are updated and prioritized.</p> <p>Burndown Chart: A graph that shows the remaining work in a sprint, helping to monitor progress.</p> <p>Business Stakeholders: Individuals interested in or affected by software development, such as clients, users, and sponsors.</p> <p>CI/CD Pipeline: An automated workflow that includes continuous</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>yet, on one hand, on the other hand, rather, or, nor, conversely, at the same time, while this may be true.</p> <p>Addition</p> <p>And, in addition to, furthermore, moreover, besides, than, too, also, both-and, another, further, last, as well as, in the same way, for example, for instance, however, thus, Phrases used in reported speech</p>	<p>"They hadn't met before the conference."</p> <p>Questions</p> <p>Had + subject + past participle</p> <p>"Had you visited Paris before 2019?"</p> <p>"What had they done before the accident?"</p> <p>Uses of the Past Perfect</p> <p>Action Completed Before Another Past Action</p>	<p>integration, testing, and deployment of software.</p> <p>Continuous Deployment:</p> <p>Practice of regularly releasing updated versions of software to end users.</p> <p>Continuous Delivery:</p> <p>Extension of continuous integration that ensures the software is always ready for deployment.</p> <p>Continuous Integration:</p> <p>Automated process of merging developers' work into a common repository to quickly</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Idioms and common colloquial expressions</p> <p>"Back to the drawing board" – When something doesn't go as planned and needs to be reworked or redesigned.</p> <p>Example: "The initial user feedback wasn't great, so it's back to the drawing board for the new feature."</p> <p>"Get the ball rolling" – To begin or start</p>	<p>"I had studied English before I moved to London."</p> <p>(The action of studying happened first, followed by moving.)</p> <p>To Give Background Information</p> <p>"When we arrived, the train had already left."</p> <p>(The train left before we arrived.)</p> <p>With Time Expressions (e.g., already, never, just, by the time)</p>	<p>detect errors.</p> <p>Change Log: Record of modifications made to the software during the development cycle.</p> <p>Daily Stand-Up (Daily Meeting): Short meeting where the team syncs progress, identifies blockers, and plans the day.</p> <p>Definition of Done: A list of criteria that a task or feature must meet to be considered complete.</p> <p>Development Team: A multidisciplinary group of</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>something, especially a project or process.</p> <p>Example: "Let's get the ball rolling on the new sprint planning for the upcoming release."</p> <p>"In the weeds" – Getting caught up in small, detailed issues instead of focusing on the bigger picture.</p> <p>Example: "We've been in the weeds with debugging for hours; let's step back and</p>	<p>"She had never seen the sea before that day."</p> <p>"By the time he called, I had gone to bed."</p> <p>Examples in Context</p> <p>"They had packed everything before the taxi arrived."</p> <p>"I hadn't been to that restaurant until last month."</p> <p>"Had you ever tried sushi before you went to Japan?"</p>	<p>professionals responsible for designing, building, and testing software.</p> <p>DevOps: Culture that combines development and operations to automate and optimize software delivery.</p> <p>Epic: A large feature or set of user stories divided into smaller tasks to facilitate development.</p> <p>Evolutionary Development (Iterative & Incremental): A development principle that combines</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>assess the overall progress."</p> <p>"Touch base" – To briefly discuss or check in on something, often used for progress updates. Example: "Let's touch base after the daily standup to review the quality metrics."</p> <p>"Fail fast" – The principle of quickly identifying problems or failures early on in the development process</p>	<p>"By the time we got to the airport, the plane had taken off."</p> <p>References: Recommended sources Cambridge University Press. (n.d.). Past perfect tense. Cambridge Grammar. Retrieved from https://dictionary.cambridge.org British Council. (n.d.). Past perfect tense. Learn English – British Council. Retrieved from</p>	<p>continuous improvement with value delivery in small functional parts.</p> <p>Feedback: Input from stakeholders or users to improve the software.</p> <p>Historical Changes (Change Log): A record of modifications made to the software throughout the development cycle.</p> <p>Increment: A functional product delivered at the end of each sprint, adding value to the existing system.</p> <p>Iteration: A development</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>so they can be corrected sooner.</p> <p>Example: "We need to fail fast in this testing phase, so we can focus on improving before the release."</p> <p>"On the same page" – To ensure everyone has the same understanding or is aligned on a topic or decision.</p>	<p>https://learnenglish.britishcouncil.org</p> <p>Azar, B. S., & Hagen, S. A. (2009). Understanding and using English grammar (4th ed.). Pearson Education. Retrieved from https://www.pearson.com</p> <p>Purdue Online Writing Lab (OWL). (n.d.). The past perfect tense. Purdue OWL. Retrieved from https://owl.purdue.edu</p>	<p>cycle where a part of the software is built, tested, and improved.</p> <p>Kanban: A visual method for managing workflow and prioritizing tasks in an Agile project.</p> <p>Kanban Board: A visual tool that organizes tasks into columns based on their status (e.g., to-do, in progress, completed).</p> <p>Minimum Viable Product (MVP): A basic version of a software product with essential features to validate its market</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
	<p>Oxford University Press. (n.d.). Past perfect tense. Oxford English Grammar Course. Retrieved from https://elt.oup.com</p>	<p>viability.</p> <p>Pair Programming: A technique where two developers work together on the same code to improve quality and share knowledge.</p> <p>Pipeline CI/CD: Automated workflow covering software integration, testing, and deployment.</p> <p>Planning Poker: A collaborative estimation technique used to assign effort values to tasks.</p> <p>Prioritization: The process</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>of determining the order of importance for backlog tasks or features.</p> <p>Product Backlog: A prioritized list of features, tasks, and improvements to be implemented in the software.</p> <p>Product Owner (PO): The person responsible for defining and prioritizing product requirements from the customer's perspective.</p> <p>Prototyping (Rapid Prototyping): Creating preliminary versions of</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>software to validate ideas before full development.</p> <p>Quality Assurance (QA): Processes and activities ensuring the software meets the required standards and works as expected.</p> <p>Regression Testing: Verification that new updates do not negatively impact existing functionalities.</p> <p>Release Management: The process of planning, scheduling, and</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>controlling software version releases.</p> <p>Retrospective (Sprint Retrospective): A meeting where the team reflects on what worked well and what can be improved in the next sprint.</p> <p>Review (Sprint Review): A meeting at the end of each sprint where the team presents the completed work to stakeholders.</p> <p>Scrum: An Agile framework that organizes</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>development into short cycles and focuses on team collaboration.</p> <p>Scrum Master: The facilitator who ensures the correct application of Agile practices and removes obstacles in the process.</p> <p>Spike: An exploratory task to investigate technology, risk, or functionality before implementation.</p> <p>Sprint: A work iteration in an Agile framework, typically lasting 1 to 4</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>weeks, delivering a complete functionality.</p> <p>Sprint Planning: A meeting where the team defines which tasks will be completed in the next sprint.</p> <p>Stakeholders: People who are interested in or affected by the software development process, such as customers, users, and sponsors.</p> <p>Technical Debt: Additional work created when shortcuts are taken in development that</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>must be corrected later.</p> <p>Velocity (Team Velocity):</p> <p>A metric that measures the amount of work completed in a sprint.</p> <p>Quality Software</p> <ul style="list-style-type: none">• Calidad del Software: Grado en el que un software cumple con los requisitos funcionales, no funcionales y las expectativas del usuario.• Requisitos Funcionales: Funcionalidades específicas que el	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>software debe cumplir (ej. autenticación, generación de reportes).</p> <ul style="list-style-type: none">• Requisitos No Funcionales: Características de calidad como rendimiento, seguridad, usabilidad y escalabilidad.• Aseguramiento de Calidad (QA): Conjunto de actividades diseñadas para garantizar que el software cumple con los estándares de calidad.	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<ul style="list-style-type: none">• Pruebas de Software: Proceso de evaluar el software para identificar errores y verificar su correcto funcionamiento.• Pruebas Unitarias: Verificación de pequeños componentes del software de forma individual.• Pruebas de Integración: Evaluación de cómo interactúan los diferentes módulos o componentes del software.	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<ul style="list-style-type: none">• Pruebas de Sistema: Validación del software completo para garantizar que cumple con los requisitos establecidos.• Pruebas de Usuario: Evaluación realizada por usuarios finales para verificar que el software satisface sus necesidades.• Pruebas de Rendimiento: Evaluación del software para medir su velocidad, capacidad de respuesta y	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>estabilidad bajo carga.</p> <ul style="list-style-type: none">• Pruebas de Seguridad: Evaluación del software para identificar y mitigar vulnerabilidades que puedan comprometer la información.• Pruebas de Regresión: Verificación de que las modificaciones realizadas al software no afecten negativamente las funcionalidades existentes.• Defectos: Errores	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>en el software que causan un funcionamiento incorrecto o inesperado.</p> <ul style="list-style-type: none">• Depuración: Proceso de identificar, analizar y corregir defectos en el código del software.• Control de Calidad (QC): Inspección directa del producto final para garantizar que cumpla con los estándares de calidad.• Criterios de Aceptación:	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>Condiciones específicas que el software debe cumplir para ser aceptado como terminado.</p> <ul style="list-style-type: none">• Documentación de Pruebas: Registro de los casos de prueba, resultados y hallazgos durante el proceso de evaluación.• Trazabilidad: Relación entre los requisitos, su implementación y las pruebas realizadas para verificarlos.	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<ul style="list-style-type: none">• ISO 25010: Norma internacional que define las características de calidad del software, como funcionalidad, confiabilidad y mantenibilidad.• Mantenibilidad: Capacidad del software para ser modificado o actualizado fácilmente.• Confiabilidad: Grado en el que un software puede realizar sus funciones sin fallar durante un periodo específico.	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<ul style="list-style-type: none">• Portabilidad: Facilidad con la que el software puede trasladarse y ejecutarse en diferentes entornos o plataformas.• Usabilidad: Facilidad de uso y aprendizaje del software para los usuarios finales.• Escalabilidad: Capacidad del software para manejar un aumento en el número de usuarios o carga de trabajo sin afectar su rendimiento.	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<ul style="list-style-type: none">• Automatización de Pruebas: Uso de herramientas para ejecutar pruebas de software de manera automática y repetitiva.• Casos de Prueba: Conjunto de condiciones o escenarios diseñados para verificar una funcionalidad específica del software.• Ciclo de Vida del Desarrollo de Software (SDLC): Serie de etapas en el desarrollo del software que incluye	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>análisis, diseño, desarrollo, pruebas y mantenimiento.</p> <ul style="list-style-type: none">Integración Continua (CI): Práctica de combinar y probar el código con regularidad para detectar problemas de calidad rápidamente.Entrega Continua (CD): Automatización de la liberación de nuevas versiones del software para garantizar calidad y rapidez.Métricas de Calidad: Indicadores	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>utilizados para medir y evaluar la calidad del software (ej. defectos por línea de código, tiempo de respuesta).</p> <ul style="list-style-type: none">• Mejora Continua: Proceso de identificar y aplicar cambios para aumentar la calidad del software de forma iterativa.• Ciclo Deming (PDCA): Metodología de mejora continua que sigue las fases Planificar, Hacer, Verificar y Actuar.• Test-Driven	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>Development (TDD): Técnica de desarrollo donde se escriben pruebas antes de implementar el código.</p> <ul style="list-style-type: none">• Análisis Estático: Evaluación del código fuente sin ejecutarlo para identificar problemas de calidad.• Análisis Dinámico: Evaluación del software en ejecución para identificar problemas relacionados con el rendimiento o comportamiento.	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<ul style="list-style-type: none">• Auditoría de Calidad: Revisión independiente de los procesos y productos de software para verificar que cumplan con estándares establecidos.• Benchmarking: Comparación del software con estándares de la industria o productos similares para evaluar su calidad.• Metodologías Ágiles: Prácticas que incluyen pruebas continuas y	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>retroalimentación para garantizar la calidad en cada iteración de desarrollo.</p> <ul style="list-style-type: none">• Plan de Pruebas: Documento que detalla la estrategia, alcance y cronograma para evaluar la calidad del software.• Refactorización: Proceso de mejorar el diseño interno del código sin alterar su funcionalidad externa.• Deuda Técnica: Compromisos a corto	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>plazo en calidad que requieren trabajo adicional para ser corregidos en el futuro.</p> <ul style="list-style-type: none">• Conformidad <p>Normativa: Cumplimiento de regulaciones y estándares en el desarrollo de software (ej. GDPR, ISO).</p>	

Curricular Design

Subject Area: English Oriented to Software Applications DevelopmentGrade: Tenth



CEFR : B1.2

Scenario 3: Software Development

Theme 2: Object-Oriented Programming

Time: 32 hours

Essential Question: In what way can object-oriented programming be helpful in industry?

Essential Competences: Empowerment

New Citizenship Axis: Sustainable Development Education

Table 36

Curriculum Pedagogical Design

Goals	Performance Indicator
The learners can...	The student...
Implement preventive techniques aimed at maintaining self-control and empowerment.	<ul style="list-style-type: none">Explains the concept of self-control.Differentiates the ways to lose or regain control.Uses self-control techniques in daily situations.
Demonstrate actions that promote sustainable development	<ul style="list-style-type: none">Defines the concept of sustainable development.



Goals	Performance Indicator
The learners can...	The student...
	<ul style="list-style-type: none">Explains ways to stop the misuse of resources.Applies techniques to promote sustainable development.

TABLE 12

Oral and Written Comprehension

Goals	Performance Indicator
The learners can...	The student...
Listening: Follow much of everyday conversation and discussion about object-oriented programming (OOP) provided it takes place in standard speech and is	<ul style="list-style-type: none">Identifies object-oriented programming languages.Recognizes the characteristics of the best object-oriented programming languages.Compares information given in different texts about the OOP history.



clearly articulated in a familiar accent.	
Reading: Scan longer texts to gather information from different parts of the text, or different manuals to fulfill a specific task related to object-oriented programming languages.	<ul style="list-style-type: none">• Identifies object-oriented programming languages.• Recognizes the characteristics of the best object-oriented programming languages.• Compares information given in different texts about the OOP history.

Table 37

Oral and Written Production

Goals		Performance Indicator
The learners can...	The student...	
Spoken Interaction: Take part in routine formal discussions that are conducted in clearly	<ul style="list-style-type: none">• Identifies the concept of class, method, constructors, and destructors in OOP.• Describes the differences between constructors and destructors.	



articulated speech in the standard form of the language and which involve the exchange of factual information, receiving instructions, or the discussion about object-oriented programming and problem-solving.

- Suggests the most common differences between method and constructor.

Spoken Production:

Deliver short, rehearsed announcements despite possibly very foreign stress and intonation, are nevertheless clearly intelligible when talking about object-oriented programming and

- Describes Abstract Data Types (ADTs) such as queues, stacks, trees, and graphs that form building blocks for solving problems.
- Lists a wide range of real problems solving with OOP.
- Describes the top languages used in large organizations for OOP.
- Employs a range of phonological features in the target language by manipulating prosodic features of spoken discourse (e.g. stress, intonation, rhythm) to support the message intended to convey.



problem-solving. Produce sounds and prosodic patterns.	
Writing: Write a very brief report in a standard conventionalized format about the applications of Object-Oriented Programming, articulating one's own efforts with others.	<ul style="list-style-type: none"> • Uses vocabulary, grammar, and discourse markers studied in this unit to write a brief report about the applications of Object-Oriented Programming. • Writes a simple, structured informational brochure that contains information about the applications of Object-Oriented Programming.

Table 38

Learnings of the curriculum pedagogical design

Functions and Discourse Markers	Grammar	Vocabulary	Phonology
Functions	Past tense responses	Object-Oriented Programming: refers to	C.INTONATION



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Describing use of Object-Oriented Programming.</p> <p>Checking understanding of the elements of OOP.</p> <p>Taking the initiative in interaction.</p> <p>Expressing opinions about how to work effectively with others to achieve the goal.</p>	<p>Past Tense Responses Overview</p> <p>Past tense responses are used to answer questions or confirm actions/events that happened in the past. These responses can be given in different past tenses depending on the situation.</p> <p>Common Past Tenses for Responses:</p> <p>Past Simple – for actions completed at a specific time in the past.</p>	<p>an approach of programming that guides the users on how to work with it.</p> <p>Object-Oriented Programming (OOP) Vocabulary</p> <p>Abstraction – Process of hiding internal details of a class and exposing only essential functionalities.</p> <p>Access Modifiers – Keywords that control the visibility of a class's attributes and methods (e.g., public, private, protected).</p> <p>Aggregation –</p>	<p>1. Phrases ending with a Falling Pitch(↘)</p> <p>•Statements</p> <p>Nice to meet ↘you.</p> <p>I'll be back in a ↘minute.</p> <p>•Commands</p> <p>Write your name ↘here.</p> <p>Leave it on the ↘desk.</p> <p>•Wh- questions (requesting information.)</p>



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
Discourse Markers Giving a result Phrases used to give opinions Contrast On the contrary, contrarily, notwithstanding, but, however, nevertheless, in spite of, in contrast, yet, on one hand, on the other hand, rather, or, nor, conversely, at the same time, while this may be true. Addition And, in addition to, furthermore, moreover,	Question: "Did you go to the party?" Response: "Yes, I went to the party last night." Time expressions: yesterday, last week, two days ago, in 2020 "I saw that movie yesterday." "We moved to this city two years ago." Past Continuous – for actions that were in	Relationship where one object contains references to others, but they can exist independently. Association – Relationship between classes defining how their objects interact without strong dependency. Casting – Conversion of an object from one class to another within an inheritance hierarchy. Class – Template or model defining the properties (attributes)	What country do you come ↘from? Where do you ↘work?



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>besides, than, too, also, both-and, another, further, last, as well as, in the same way, for example, for instance, however, thus, Phrases used in reported speech</p> <p>Idioms and common colloquial expressions</p> <p>8 idioms and common colloquial expressions that may be used within the topic of Object-Oriented Programming (OOP):</p>	<p>progress at a specific time in the past.</p> <p>Question: "What were you doing at 8 p.m.?"</p> <p>Response: "I was reading a book."</p> <p>Past Perfect – to indicate an action that happened before another action in the past.</p> <p>Question: "Had you finished the project before the meeting?"</p> <p>Response: "Yes, I had finished it by then."</p>	<p>and behaviors (methods) of an object.</p> <p>Class Cohesion –</p> <p>Measure of how closely related and focused the responsibilities of a class are.</p> <p>Class Diagram (UML – Unified Modeling Language) – Visual tool for modeling classes, objects, and relationships in OOP.</p> <p>Class Final – A class that cannot be inherited by other classes.</p> <p>Class Instantiation –</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>"Don't reinvent the wheel" – Avoid creating something from scratch when a solution already exists.</p> <p>Example: "Instead of writing your own sorting algorithm, just use the built-in methods—don't reinvent the wheel."</p> <p>"A well-oiled machine" – A system or program that runs smoothly and efficiently.</p>	<p>Past Perfect Continuous – for actions that were ongoing up until another point in the past (less common at B1.1, but useful).</p> <p>Question: "How long had you been waiting?"</p> <p>Response: "I had been waiting for an hour when she arrived."</p> <p>Examples of Past Tense Responses in Context:</p> <p>Past Simple:</p>	<p>Process of creating an object from a class.</p> <p>Class Generic – Classes that allow working with any data type without specifying it until instantiation.</p> <p>Class Singleton – A design pattern that ensures a class has only one instance in the entire program.</p> <p>Composition – Relationship where an object depends entirely on another for its existence.</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Example: "With proper encapsulation and modular design, your OOP project will run like a well-oiled machine."</p> <p>"Break it down" – To simplify or divide a complex task into smaller, manageable parts.</p> <p>Example: "If your class is getting too big, break it down into multiple smaller classes with clear responsibilities."</p>	<p>"Did you call your friend?"</p> <p>– "Yes, I called her yesterday."</p> <p>Past Continuous:</p> <p>"Were you studying at 10 p.m.?"</p> <p>– "No, I was watching TV."</p> <p>Past Perfect:</p> <p>"Had they left before you arrived?"</p> <p>– "Yes, they had already left."</p>	<p>Constructor – Special method in a class that executes automatically when an object is created, initializing attributes.</p> <p>Coupling – Degree of dependency between classes; lower coupling is preferred for better reusability and maintenance.</p> <p>Data Serialization – Process of converting an object into a format that can be stored or transmitted, such as</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>"Spaghetti code" – Messy, unstructured, and difficult-to-maintain code.</p> <p>Example: "Without proper class design, your project will turn into a pile of spaghetti code."</p> <p>"Keep it DRY" (Don't Repeat Yourself) – A principle in OOP that encourages reusable code to avoid redundancy.</p>	<p>Mixed Tenses:</p> <p>"What happened during the meeting?"</p> <p>– "I was presenting when the fire alarm went off."</p> <p>– "Luckily, we had saved all the documents before the power cut."</p> <p>Short Responses in Past Tense:</p> <p>Sometimes short answers are enough:</p>	<p>JSON or XML.</p> <p>Data Deserialization – Reverse process of serialization, converting stored formats back into an object of the original class.</p> <p>Dependency Inversion – Principle stating that high-level modules should not depend on low-level modules but on abstractions.</p> <p>Destructor – Special method that executes when an object is no longer needed and</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>Example: "Instead of copying and pasting the same code, use inheritance and keep it DRY."</p> <p>"The blueprint for success" – A solid plan or design that ensures a successful outcome (similar to how a class is a blueprint for objects).</p> <p>Example: "Your class structure is the blueprint for success—make sure you design it well."</p>	<p>"Did you go to the gym?" → "Yes, I did." / "No, I didn't."</p> <p>"Was it raining?" → "Yes, it was." / "No, it wasn't."</p> <p>"Had they finished the project?" → "Yes, they had." / "No, they hadn't."</p> <p>References:</p> <p>Recommended sources</p> <p>Cambridge University Press. (n.d.). Past perfect tense: Questions and responses. Cambridge Grammar. Retrieved from</p>	<p>releases associated resources.</p> <p>Encapsulation – Principle that restricts direct access to a class's attributes and methods, providing control through public methods.</p> <p>Exception Propagation – Sending detected errors to higher-level methods for proper handling.</p> <p>Event – Actions handled by objects of a class in response to user interactions or system changes.</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
<p>"Loose coupling, high cohesion" – A best practice in OOP where components are independent but focused on their specific roles.</p> <p>Example: "Your modules should follow the principle of loose coupling and high cohesion for better maintainability."</p> <p>"A jack of all trades, master of none" – A class or object that tries</p>	<p>https://dictionary.cambridge.org</p> <p>British Council. (n.d.). Past perfect tense: Questions and negative responses. Learn English – British Council. Retrieved from https://learnenglish.britishcouncil.org</p> <p>Azar, B. S., & Hagen, S. A. (2009). Understanding and using English grammar (4th ed.). Pearson Education. Retrieved from</p>	<p>Factory Pattern – A design pattern that delegates object creation to a specialized method or class.</p> <p>Final Method – A method that cannot be overridden in derived classes.</p> <p>Graphical User Interfaces (GUI) – Applications developed with OOP that present interactive visual elements to users.</p> <p>Inheritance – Mechanism allowing a class (subclass) to derive</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
to do too many things and doesn't do any of them well. Example: "Your class is handling too many responsibilities—avoid making it a jack of all trades, master of none."	https://www.pearson.com Purdue Online Writing Lab (OWL). (n.d.). Forming questions and responses with the past perfect tense. Purdue OWL. Retrieved from https://owl.purdue.edu Oxford University Press. (n.d.). Past perfect tense: Forming questions and short responses. Oxford English Grammar Course. Retrieved from https://elt.oup.com	properties and methods from another class (superclass). Instance – Concrete representation of a class in memory. Interfaces – Set of abstract methods that must be implemented by a class that inherits from it. Iterators – Tools allowing the traversal of elements in a collection or data structure. Method – Function within a class defining an	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>object's behavior.</p> <p>Method Overloading – Defining multiple methods with the same name but different parameters in the same class.</p> <p>Method Overriding – Modifying an inherited method from the base class in a derived class to change its behavior.</p> <p>Method Static – A method that belongs to the class instead of objects created from it and can be called</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>without instantiation.</p> <p>Object – Instance of a class containing specific values for its attributes and capable of executing its methods.</p> <p>Observer Pattern – A design pattern where an object (observer) automatically updates when another object (subject) changes.</p> <p>Pattern Design – Reusable solutions for common problems in object-oriented software development (e.g.,</p>	



Functions and Discourse Markers	Grammar	Vocabulary	Phonology
		<p>Singleton, Factory, Observer).</p> <p>Polymorphism – Ability of a method or function to behave differently depending on the object or context in which it is used.</p> <p>Properties (Getters and Setters) – Methods used to access and modify a class's private attributes.</p> <p>Static Variable – An attribute that belongs to the class and is shared by all instances of that class.</p>	



References

Business Opportunities and Models

Finneo. (n.d.). Glossary of business development terms. Retrieved from <https://www.finneo.com/blog/glossary-of-business-development-terms>

Investopedia. (n.d.). Business model. Retrieved from <https://www.investopedia.com/terms/b/businessmodel.asp>

FIRMSconsulting. (n.d.). Business terms glossary - A to Z. Retrieved from <https://www.firmsconsulting.com/quarterly/business-terms-glossary/>

Wikipedia contributors. (2025, January 15). Business model. In Wikipedia, The Free Encyclopedia. Retrieved from https://en.wikipedia.org/wiki/Business_model

Vogue Business. (2025, February 6). The Vogue Business Glossary. Retrieved from <https://www.voguebusiness.com/story/fashion/the-vogue-business-glossary>



SWOT Analysis How to Develop a Strategy For Success, by the Mind Tools Content Team; Retrieved 2020, Sep 22, from https://www.mindtools.com/pages/article/newTMC_05.htm

Vocabulario que todo emprendedor debe conocer, 08/07/2013 en Emprendedores, by EmpreneRioja, Formación; Retrieved 2020, Sep 22, from <http://emprenderioja.es/blog/2013/07/08/vocabulario-que-todo-emprendedor-debe-conocer/>

Concept and functions of marketing, by marketingsuccessonline.com, June 9, 2020; Retrieved 2020, Sep 22, from <https://www.marketingsuccessonline.com/concept-and-functions-of-marketing-4-3/>

What is Branding?, by Shital M; Retrieved 2020, Sep 22, from <https://www.economicdiscussion.net/marketing-2/brand/what-is-branding/32274>

Product Life Cycle Stages by Shital M; Retrieved 2020, Sep 22, from <https://www.economicdiscussion.net/marketing-management/product-life-cycle/product-life-cycle-stages/32286>

Creation of a Company for a Living



Vocabulario que todo emprendedor debe conocer, 08/07/2013 en Emprendedores, by EmprendeRioja, Formación; Retrieved 2020, Sep 22, from <http://emprenderioja.es/blog/2013/07/08/vocabulario-que-todo-emprendedor-debe-conocer/>

Currency Exchange Rates Explained by Travelex Emirates Exchange LLC; Retrieved 2020, Sep 22, from <https://www.travelexae.com/AE/Foreign-Currency/Rates/Online-Rates/Currency-Exchange-Rates-Explained/>

Atlan. (2023, September 20). Business glossary template: How to create one in 2024? Retrieved from <https://atlan.com/business-glossary-template/>

ER/Studio. (2023, August 15). Building a business glossary: 7 step guide. Retrieved from <https://erstudio.com/blog/building-a-business-glossary-7-step-guide/>

Goodwall. (2023, July 10). Entrepreneur glossary: 33+ entrepreneurship vocabulary terms. Retrieved from <https://www.goodwall.io/blog/entrepreneur-glossary/>

Quizlet. (n.d.). Entrepreneurship key terms flashcards. Retrieved from <https://quizlet.com/9513798/entrepreneurship-key-terms-flash-cards/>

Universidad Mayor. (2022). 100-word entrepreneurial glossary [PDF]. Retrieved from <https://lafabrica.umayor.cl/images/entrepreneurship-glossary.pdf>



Data-Base Design

Raima. (n.d.). Database terminology - Top 150 database terms. Retrieved from <https://raima.com/database-terminology/>

Prisma. (n.d.). Database glossary. In Prisma's Data Guide. Retrieved from <https://www.prisma.io/dataguide/intro/database-glossary>

Bytebase. (n.d.). Database glossary. Retrieved from <https://www.bytebase.com/database-glossary/>

BCcampus Open Publishing. (n.d.). Glossary. In Database design – 3rd edition. Retrieved from <https://pressbooks.bccampus.ca/nelson/chapter/glossary/>

Noble Desktop. (n.d.). Database management systems, tables, & keys. Retrieved from <https://www.nobledesktop.com/learn/sql/database-terminology>

Relational Data-Based



IBM. (n.d.). SQL relational database and system terminology. Retrieved from

<https://www.ibm.com/docs/en/i/7.3?topic=concepts-sql-relational-database-system-terminology>

Prisma. (n.d.). Database glossary. In Prisma's Data Guide. Retrieved from

<https://www.prisma.io/dataguide/intro/database-glossary>

MariaDB. (n.d.). Relational databases: Basic terms. Retrieved from <https://mariadb.com/kb/en/relational-databases-basic-terms/>

Noble Desktop. (n.d.). Database management systems, tables, & keys. Retrieved from

<https://www.nobledesktop.com/learn/sql/database-terminology>

SolarWinds. (n.d.). What is a relational database?—IT glossary. Retrieved from

<https://www.solarwinds.com/resources/it-glossary/relational-database>

Non-Relational DataBase

MongoDB. (s. f.). What is a Non-Relational Database? Recuperado 1 de octubre de 2021, de

<https://www.mongodb.com/databases/non-relational>

Vilchynska, H. (2020, 9 noviembre). What Are NoSQL Databases And Why Enterprises Rely On NoSQL? DevCom.

<https://devcom.com/tech-blog/what-are-nosql-databases/>



Microsoft. (2023). Non-relational data and NoSQL. Azure Architecture Center. <https://learn.microsoft.com/en-us/azure/architecture/data-guide/big-data/non-relational-data>

SolarWinds. (2023). What is a NoSQL database? IT Glossary. <https://www.solarwinds.com/resources/it-glossary/nosql-database>

MongoDB. (2023). What is a non-relational database? <https://www.mongodb.com/resources/basics/databases/non-relational>

Amazon Web Services. (2023). What is a NoSQL database? <https://aws.amazon.com/nosql/>

IBM. (2023). What is a NoSQL database? <https://www.ibm.com/think/topics/nosql-databases>

Agile and Quality Development

Agile Alliance. (n.d.). Agile glossary. Retrieved February 18, 2025, from <https://www.agilealliance.org/agile101/agile-glossary/>

Atlassian. (n.d.). A glossary of 26 agile terms to boost your project management game. Retrieved February 18, 2025, from <https://www.atlassian.com/blog/project-management/agile-terms-project-management>



Testlio. (n.d.). Glossary of QA terms. Retrieved February 18, 2025, from <https://help.testlio.com/en/articles/4283292-glossary-of-qa-terms>

ClickUp. (n.d.). 50 Agile Scrum terms everyone must know (detailed glossary). Retrieved February 18, 2025, from <https://clickup.com/blog/agile-scrum-terms/>

Association for Project Management. (n.d.). Agile project management glossary. Retrieved February 18, 2025, from <https://www.apm.org.uk/resources/find-a-resource/agile-project-management/glossary/>

ETQ. (n.d.). *A glossary of quality and quality management terms*. Retrieved February 18, 2025, from <https://www.etq.com/a-glossary-of-quality-and-quality-management-terms/>

Wrike. (n.d.). A glossary of Agile terminology. Retrieved February 18, 2025, from <https://www.wrike.com/agile-guide/glossary/>

Zenhub. (2023, December 15). Agile terminology: 100 Agile definitions. Retrieved February 18, 2025, from <https://blog.zenhub.com/agile-terminology-100-agile-definitions/>

Coursera. (n.d.). Agile terms: A to Z glossary. Retrieved February 18, 2025, from <https://www.coursera.org/collections/agile-terms>



Indeed. (2023, August 15). 12 Key Object-Oriented Programming Terms (With Definitions). Retrieved February 18, 2025, from <https://www.indeed.com/career-advice/career-development/oop-terminology>

University of Cincinnati. (n.d.). Object Oriented (OO) Terminology/Definitions. Retrieved February 18, 2025, from https://homepages.uc.edu/~thomam/Misc/OO_Terminology.html

University of Minnesota Duluth. (n.d.). Object-Oriented Terminology. Retrieved February 18, 2025, from <https://www.d.umn.edu/~gshute/softeng/object-oriented.html>

University of Kent. (n.d.). Glossary of Java and Object-Oriented Terminology. Retrieved February 18, 2025, from <https://www.cs.kent.ac.uk/people/staff/djb/oop/glossary.html>

Mozilla Developer Network. (2023, December 15). OOP - MDN Web Docs Glossary: Definitions of Web-related terms. Retrieved February 18, 2025, from <https://developer.mozilla.org/en-US/docs/Glossary/OOP>



Referencias Bibliográficas

Referencias Generales

Adam, S. (julio de 2004). Using Learning Outcomes: A Consideration of the Nature, Role, Application and Implications for European Education of Employing "Learning Outcomes" at the Local, National and International Levels.

[https://www.scirp.org/\(S\(351jmbntvnsjt1aadkposzje\)\)/reference/ReferencesPapers.aspx?ReferenceID=1692948](https://www.scirp.org/(S(351jmbntvnsjt1aadkposzje))/reference/ReferencesPapers.aspx?ReferenceID=1692948)

Álvarez-Galván, J. L. (2015). Revisión Destrezas más allá de la Escuela en Costa Rica. OCDE.

<https://www.comex.go.cr/media/8069/libro-sbs-cr-versi%C3%B3n-espa%C3%B1ol-digital.pdf>

AZ Revista de Educación y Cultura. (28 de noviembre de 2014). ¿Cuál es el rol del docente en el desarrollo de las competencias genéricas? <https://educacionyculturaaz.com/cual-es-el-rol-del-docente-en-el-desarrollo-de-las-competencias-genericas/>

Biggs, J. (1996). Enhancing teaching through constructive alignment. *Higher Education*, 32(3), 347–364.

Biggs, J. (2004). Calidad del aprendizaje universitario. Narcea. <https://barajasvictor.wordpress.com/wp-content/uploads/2014/05/libro-j-biggs.pdf>



Cabrerizo, J. y Castillo, S. (2010). Evaluación educativa de aprendizajes y competencias. Pearson Educación, S. A.

Carlino, F. (2020). De la alineación al alineamiento constructivo. Más allá de la trampa mecanicista. Cuaderno de Pedagogía Universitaria, 18(35), 58-70. file:///C:/Users/srojass/Downloads/413-Texto%20del%20art%C3%ADulo-1456-1-10-20210121.pdf

Carrasco, M. A. (2016). Aprendizaje, competencias y TIC. Pearson.

Castillo, S. y Cabrerizo, J. (2010). Evaluación educativa de aprendizajes y competencias. Pearson.
https://gc.scalahed.com/recursos/files/r161r/w25469w/Doctorado/L_evaluac3b3n_educativa-de-aprendizajes-y-competencias.pdf

Consejo Superior de Educación. (18 de julio de 2016). Acuerdo CSE N° 06-37-2016: Marco Nacional De Cualificaciones Educación y Formación Técnica Profesional. <http://cse.go.cr/marco-nacional-de-cualificaciones-educacion-y-formacion-tecnica-profesional>

Consejo Superior Universitario Centroamericano (CSUCA). (2018). Marco de Cualificaciones para la Educación Superior Centroamericana (MCESCA): resultados de aprendizaje esperados para los niveles técnico. Serviprensa.

De Zubiría, J. (2010). Los modelos pedagógicos. Neisa.



Delors, J. (1994). La educación encierra un tesoro. UNESCO

https://unesdoc.unesco.org/ark:/48223/pf0000109590_spa

Espejo, R. y Sarmiento, R. (2017). Metodologías activas para el aprendizaje. Universidad Central de Chile.

https://www.postgradosucentral.cl/profesores/download/manual_metodologias.pdf

Ferreiro, R. (2007). Nuevas alternativas de aprender y enseñar. Aprendizaje cooperativo. Trillas.

Ferreiro, R. (2009). El ABC del aprendizaje cooperativo. Trabajo en equipo para aprender y enseñar. Trillas.

Gómez, J., Monroy, L. y Bonilla, C. (2019). Caracterización de los modelos pedagógicos y su pertinencia en una educación contable crítica. Entramado, 15(1). 1-42.

<https://www.redalyc.org/journal/2654/265460762011/265460762011.pdf>

López, E. (2016). En torno al concepto de competencia: un análisis de fuentes. Revista de Currículum y Formación de Profesorado, 20(1). 311-322. <https://www.redalyc.org/pdf/567/56745576016.pdf>

López, M. (2013). Aprendizaje, competencias y TIC. Pearson.

<https://ie42003cgalbarracin.edu.pe/biblioteca/LIBR-NIV312062023225715.pdf>

Manpower Group. (2018). Resolviendo la Escasez de Talento Construir, adquirir, tomar prestado y tender puentes. <https://www.manpowergroup.com.ar/wps/wcm/connect/manpowergroup/ced492e5-ffa1-4538-9192->



613ceeda22f4/Encuesta+de+Escasez+de+Talento+2018.pdf?MOD=AJPERES&CONVERT_TO=url&CACHEID=ced492e5-ffa1-4538-9192-613ceeda22f4

Mckeown, R. (2002). Manual de Educación para el Desarrollo Sostenible. Universidad de Tennessee.

http://www.esdtoolkit.org/manual_edsp01.pdf

MEP - MTSS - INA - CONARE - UCCAEP - UNIRE (noviembre de 2018). Marco Nacional de Cualificaciones de la Educación y Formación Técnica Profesional de Costa Rica.

http://www.detce.mep.go.cr/sites/all/files/detce_mep_go_cr/adjuntos/marco_nacional_cualificaciones_.pdf

Ministerio de Educación Pública. (2015). Transformación curricular: fundamentos conceptuales en el marco de la Visión Educar para una Nueva Ciudadanía.

https://ddc.mep.go.cr/sites/all/files/ddc_mep_go_cr/archivos/transf-curricular-v-academico-vf_0.pdf

Ministerio de Educación Pública. (2016). Política Educativa: La persona: centro del proceso educativo y sujeto transformador de la sociedad. San José, Costa Rica.

Ministerio de Educación Pública. (2022). Orientaciones y lineamientos para el desarrollo de actividades pedagógicas fuera del centro educativo en la ETP.

<https://drea.mep.go.cr/sites/default/files/publicaciones-anexos-2023/Orientaciones%20y%20lineamientos%20actividades%20fuera%20del%20CE.pdf>



Ministerio de Educación Pública (2022). Reglamento de Evaluación de los Aprendizajes.

https://www.pgrweb.go.cr/scij/Busqueda/Normativa/Normas/nrm_texto_completo.aspx?nValor1=1&nValor2=85815

Ministerio de Educación Pública (2023). Compendio de estrategias para la mediación pedagógica de la ETP.

https://detce.mep.go.cr/sites/all/files/detce_mep_go_cr/adjuntos/compendio-mediacion-pedagogica-2023.pdf

Muñoz, L. (2012). Enfoque por competencias y mercado de trabajo. Nuevas tendencias para la educación universitaria. Revista Actualidades Investigativas en Educación, 12(2), 1-30.

<https://revistas.ucr.ac.cr/index.php/aie/article/view/10283/18155>

OCDE (2021). Costa Rica – Nota del País. https://www.oecd.org/espanol/noticias/EAG2021_CN_CRI_ES.pdf

OpenAI. (2024). ChatGPT (versión del 6 de junio) [Talleres de escenario y futuro]. <https://chat.openai.com/chat>

Perrenoud, P. (2008). Diez nuevas competencias para enseñar.

<https://www.uv.mx/dgdaie/files/2013/09/Philippe-Perrenoud-Diez-nuevas-competencias-para-ensenar.pdf>

Ramírez-Díaz, J. (2020). El enfoque por competencias y su relevancia en la actualidad: Consideraciones desde la orientación ocupacional en contextos educativos. Revista Electrónica Educare, 24(2). 1-14.

<https://www.revistas.una.ac.cr/index.php/EDUCARE/article/view/10728/19230>



Robles, B. y Estévez, E. (2016). Enfoque por competencias: Problemáticas didácticas que enfrenan el profesorado. *Revista Electrónica Educare*, 20(1). 1-12.

<https://www.revistas.una.ac.cr/index.php/EDUCARE/article/view/7495/16434>

Tobón, S. (2007). *El enfoque complejo de las competencias y el diseño curricular por ciclos propedéuticos*. Grupo CIFE.

Unesco (2017). *Escuelas en acción. Ciudadanos del mundo para el desarrollo sostenible. Guía para el profesorado*. <https://unesdoc.unesco.org/ark:/48223/pf0000249129>

World Bank Group (2023). *Building better formal TVET Systems: Principles and Practice in Low-and Middle-Income Countries*. <https://www.worldbank.org/en/topic/skillsdevelopment/publication/better-technical-vocational-education-training-TVET>



Bibliografía complementaria

Yogesh Kumar Sharma, Sriramula Nagaprasad, Ramakrishna Kumar. (2022). *Fundamentos de la tecnologías de la información*. Ediciones Nuestro Conocimiento.

Aditi Sharma, Rahul Sharma. (2022). *Computación en la nube avanzada: Computación en la nube: una visión rápida*. Ediciones Nuestro Conocimiento.

Alcides Andres Riera Mora, Laura Sofía Andraus Buenhaber. (2021). *Arquitectura de algoritmos y desarrollo de software con Python 3*. Independently published.

Alejandro Salazar Yábar, Manuel Salazar Santibañez . (2022). *Programación orientada a objetos: Conceptos, clasificación, metodología / Algoritmos para representación de programas / Desarrollo de aplicaciones y ejemplos completos en JAVA* . Lima, Perú: Almasaya.

Álvarez, A. C. (2021). *Python 3, curso completo de programación. 2 Edición*. Independently published.

Balaza, J. (2023). *Domina JavaScript y React La guía completa para el desarrollo web moderno*. Independently published.

Black, M. (2024). *Diseño de Experiencia de Usuario (UX): Fundamentos, Estrategias y Tendencias para el Futuro Digital*. Independently published.

Briceño, E. V. (2021). *Seguridad de la Información*. Alicante: Editorial Área de Innovación y Desarrollo, S.L.



Burgos, A. (2023). *UX/UI Fundamentos del diseño*. Astudio; 1er edición.

Carrasco, J. (2020). *Desarrollo de aplicaciones móviles en Kotlin*. Independently published.

Chamorro, J. B. (2025). *IDEAR UX: Arquitectura de la información, Diseño de interacción para diseñar experiencia de usuarios*. Estados Unidos: Independently published.

Ernesto Galvis Lista, Alexander Bustamante Martínez. (2023). *Bases de Datos Relacionales Un enfoque aplicado y orientado a resultados de aprendizaje*. Universidad del Magdalena: Editorial Unimagdalena.

Farré, M. N. (2022). *Estructura de Datos y Algoritmos. Guía ilustrada para programadores*. Madrid, España: Anaya Multimedia.

Ferrari, L. (2022). *TCP / IP: Introducción a redes informáticas*. Independently published.

García, B. C. (2025). *Desarrollo de componentes software para servicios de comunicación UF1288*. Tutor Formación.

Gómez, M. R. (2021). *Curso de desarrollo Web. HTML, CSS y JavaScript. Edición 2021*. Anaya Multimedia.

Martínez, J. (2025). *Fundamentos del diseño web. Una guía para empezar desde cero*. Independently published.

Martínez, J. B. (2020). *Alfabetización digital e iniciación a la informática*. RA-MA Editorial.



Porras, A. A. (2023). *Metodologías ágiles para el desarrollo de software*.

R., R. (2024). *Base de datos sencilla Introducción a SQL y NoSQL para principiantes*. Kindle.

Sanchis, J. V. (2023). *Aprende programación con C++: Desarrolla tus propias aplicaciones (Aprende C y C++)*. Independently published.

Sanchis, J. V. (2024). *C++: Aprende a programar de forma práctica (Aprende C y C++)*. Independently published.

Serrano, C. R. (2021). *Aprende mongodb con no-sql desde principiante a experto : Comprende las bases de datos no relacionales con mongodb*. Kindle.

Serrano, C. R. (2024). *Calidad del software: Diseño del sistema de calidad del software*. Kindle.



MINISTERIO DE
EDUCACIÓN PÚBLICA

GOBIERNO
DE COSTA RICA

Dirección de Educación
Técnica y Capacidades
Emprendedoras

Apéndices

Apéndice A. Estándar de Cualificación de Desarrollo de aplicaciones de software



Fuente: [chrome-](#)

[extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.cualificaciones.cr/mnc/index.php/catalogo-nacional-de-cualificaciones/06-tecnologias-de-la-informacion-y-la-comunicacion/1-tecnologias-de-la-informacion-y-la-comunicacion/3-desarrollo-y-analisis-de-software-y-aplicaciones/47-0613-01-02-4-desarrollo-aplicaciones-software/file](https://efaidnbmnnnibpcajpcglclefindmkaj/https://www.cualificaciones.cr/mnc/index.php/catalogo-nacional-de-cualificaciones/06-tecnologias-de-la-informacion-y-la-comunicacion/1-tecnologias-de-la-informacion-y-la-comunicacion/3-desarrollo-y-analisis-de-software-y-aplicaciones/47-0613-01-02-4-desarrollo-aplicaciones-software/file)



Glosario de Términos

Concepto	Definición
DÉCIMO	
Subárea herramientas para la producción	
Unidad: Alfabetización digital para el bienestar y la seguridad en línea	
Vocabulario de Alfabetización Digital para Desarrollo de Aplicaciones de Software	
Software: Conjunto de instrucciones o programas que permiten a una computadora realizar tareas específicas.	
Aplicación: Programa o conjunto de programas diseñados para realizar funciones específicas para el usuario.	
Código: Texto escrito en un lenguaje de programación que define las instrucciones que ejecutará una computadora.	
Interfaz de usuario (UI): Elementos visuales e interactivos con los que un usuario interactúa en una aplicación.	
Experiencia del usuario (UX): Sensaciones y respuestas del usuario al interactuar con una aplicación, incluyendo usabilidad y diseño.	



Algoritmo: Conjunto ordenado de pasos o reglas para resolver un problema o realizar una tarea.

Depuración: Proceso de identificar y corregir errores en el código de un programa.

Repositorio: Almacenamiento digital donde se guarda y gestiona el código fuente de un proyecto, generalmente con control de versiones.

HTML (HyperText Markup Language): Lenguaje estándar para estructurar y presentar contenido en la web.

CSS (Cascading Style Sheets): Lenguaje utilizado para diseñar y dar estilo a documentos web creados con HTML.

JavaScript: Lenguaje de programación que agrega interactividad y funcionalidad avanzada a sitios y aplicaciones web.

Python: Lenguaje de programación versátil, utilizado en desarrollo web, análisis de datos e inteligencia artificial.

Java: Lenguaje de programación ampliamente usado para aplicaciones móviles, empresariales y web.

C#: Lenguaje desarrollado por Microsoft para aplicaciones empresariales y de escritorio.

Swift: Lenguaje creado por Apple para desarrollar aplicaciones para iOS y macOS.



Desarrollo Ágil: Método de desarrollo de software basado en iteraciones rápidas, colaboración y adaptabilidad.

Scrum: Marco de trabajo ágil que organiza tareas en ciclos cortos llamados "sprints".

Kanban: Método visual para organizar y gestionar el flujo de trabajo mediante tableros y tarjetas.

MVP (Producto Mínimo Viable): Versión básica de un producto que incluye solo las funciones esenciales para validar su viabilidad.

Control de versiones: Sistema para rastrear cambios en el código fuente y colaborar con otros desarrolladores (p. ej., Git).

Integración Continua: Práctica de integrar cambios frecuentes en el código y probarlos automáticamente.

IDE (Entorno de Desarrollo Integrado): Software que facilita la escritura, depuración y prueba de código (p. ej., Visual Studio, Eclipse).

Framework: Conjunto de herramientas y bibliotecas predefinidas que simplifican el desarrollo de software (p. ej., React, Angular).

API (Interfaz de Programación de Aplicaciones): Conjunto de reglas que permite que diferentes aplicaciones se comuniquen entre sí.



Backend: Parte del software que gestiona la lógica, el almacenamiento y el procesamiento de datos en un servidor.

Frontend: Parte visible y funcional de una aplicación que interactúa directamente con el usuario.

Cloud Computing: Uso de recursos informáticos como almacenamiento y procesamiento en servidores remotos accesibles a través de internet.

Base de datos: Sistema organizado para almacenar y gestionar datos que pueden ser consultados y manipulados.

Pruebas Unitarias: Verificación de pequeñas unidades de código para asegurar que funcionan correctamente.

Pruebas de Integración: Comprobación de la interacción entre diferentes componentes del sistema.

Pruebas de Usuario: Evaluación de la aplicación por usuarios finales para garantizar que cumpla con sus expectativas.

QA (Aseguramiento de Calidad): Procesos destinados a asegurar que el software cumple con los estándares de calidad requeridos.

Inteligencia Artificial (IA): Tecnología que simula procesos de inteligencia humana en sistemas informáticos.



Machine Learning: Rama de la IA donde los sistemas aprenden y mejoran automáticamente a partir de datos.

DevOps: Cultura y prácticas que integran desarrollo y operaciones para entregar software más rápidamente.

Microservicios: Arquitectura de software donde las aplicaciones se dividen en servicios pequeños e independientes.

CI/CD (Integración y Entrega Continua): Automatización del desarrollo, pruebas y despliegue de software.

Ciberseguridad: Protección de sistemas, redes y datos frente a ataques digitales.

Protección de Datos: Prácticas y normas para garantizar la privacidad y seguridad de la información de los usuarios.

Licencias de Software: Normas legales que regulan el uso, distribución y modificación del software.

Código Ético: Principios que guían el desarrollo de software de manera responsable y profesional.



Unidad: Transformación Digital, Seguridad y Análisis de la Información

Transformación Digital: Proceso de adopción de tecnologías digitales para mejorar procesos, productos y servicios.

Big Data: Conjunto masivo de datos que requieren técnicas avanzadas para su almacenamiento, análisis y visualización.

Analítica de Datos: Proceso de examinar datos para identificar patrones, tendencias y extraer conclusiones.

Inteligencia Artificial (IA): Tecnología que permite a las máquinas realizar tareas que normalmente requieren inteligencia humana.

Machine Learning: Subcampo de la IA que permite a los sistemas aprender y mejorar automáticamente a partir de los datos.

Ciberseguridad: Prácticas y tecnologías diseñadas para proteger sistemas, redes y datos de accesos no autorizados.

Firewall: Sistema que actúa como una barrera entre una red confiable y otra no confiable para filtrar el tráfico.

Phishing: Técnica de fraude utilizada para engañar a las personas y obtener información personal o financiera.



Gestión de Identidad: Conjunto de procesos y herramientas para autenticar y autorizar el acceso de usuarios a sistemas.

Criptografía: Técnica de codificar información para protegerla de accesos no autorizados.

Blockchain: Sistema de registro distribuido que asegura la integridad de los datos mediante bloques enlazados y asegurados criptográficamente.

Computación en la Nube (Cloud Computing): Uso de recursos informáticos a través de internet en lugar de servidores locales.

IoT (Internet de las Cosas): Red de dispositivos conectados que recopilan e intercambian datos.

Automatización de Procesos: Uso de software y tecnología para realizar tareas repetitivas de manera eficiente.

Protección de Datos: Prácticas para garantizar la privacidad y seguridad de la información sensible.

Normas ISO 27001: Estándar internacional para la gestión de la seguridad de la información.

Hacker Ético: Profesional que utiliza sus habilidades para identificar vulnerabilidades y proteger sistemas.

VPN (Red Privada Virtual): Tecnología que crea una conexión segura y encriptada a través de internet.



Análisis Predictivo: Uso de datos históricos y algoritmos para predecir eventos futuros.

Data Mining: Proceso de explorar grandes conjuntos de datos para encontrar patrones y relaciones útiles.

Gobernanza de Datos: Estrategias y políticas para asegurar el uso eficiente, seguro y ético de los datos.

Deep Learning: Subcampo de Machine Learning basado en redes neuronales profundas para procesar datos complejos.

Transformación de Procesos: Optimización de métodos operativos mediante tecnología digital.

Seguridad de la Información: Prácticas que aseguran la confidencialidad, integridad y disponibilidad de los datos.

Análisis Forense Digital: Investigación técnica para recuperar, analizar y preservar datos en caso de incidentes de seguridad.

Ingeniería Social: Manipulación psicológica utilizada para engañar a las personas y obtener acceso no autorizado.

Ransomware: Tipo de malware que bloquea el acceso a datos hasta que se pague un rescate.

Autenticación Multifactor (MFA): Método de verificación que requiere dos o más formas de autenticación para acceder a un sistema.



Datos Estructurados: Información organizada en bases de datos relacionales, con filas y columnas.

Datos No Estructurados: Información sin formato definido, como correos electrónicos, videos y documentos.

Regulación General de Protección de Datos (GDPR): Ley europea que protege la privacidad de los datos personales.

Resiliencia Digital: Capacidad de una organización para recuperarse de incidentes cibernéticos o interrupciones digitales.

DevSecOps: Integración de prácticas de seguridad en cada etapa del desarrollo y operación de software.



Unidad: Herramientas para la Producción de Documentos

Procesador de Textos: Software utilizado para crear, editar y formatear documentos escritos (ej. Microsoft Word, Google Docs).

Formato de Documento: Estilo y diseño que define la apariencia de un documento, incluyendo márgenes, tipografía y espaciado.

Plantilla: Documento preformateado que sirve como base para crear nuevos documentos con diseño uniforme.

Estilo: Conjunto de características predefinidas para texto o párrafos, como fuente, tamaño y alineación.

Tipografía: Arte y técnica de diseñar y organizar texto, incluyendo tipo de letra, tamaño y espaciado.

Tabla de Contenidos: Lista estructurada de secciones y subsecciones de un documento con sus respectivas páginas.

Encabezado y Pie de Página: Elementos de texto o gráficos que se colocan en la parte superior o inferior de cada página del documento.

Numeración de Páginas: Inserción automática de números en las páginas del documento para facilitar la organización.



Formato APA: Estilo de redacción académica que regula citas, referencias y presentación de documentos.

Ortografía y Gramática: Funciones de los procesadores de texto para revisar y corregir errores en el documento.

Tabla: Elemento estructurado en filas y columnas para organizar datos en un documento.

Gráficos: Representaciones visuales como diagramas, gráficos o imágenes incorporadas en un documento.

Combinar Correspondencia: Función que permite crear múltiples documentos personalizados desde una plantilla y una base de datos.

Revisión de Cambios: Herramienta que permite realizar modificaciones en un documento y resaltar las diferencias para su seguimiento.

Exportar: Guardar un documento en diferentes formatos (ej. PDF, DOCX, HTML).

Compartir en Línea: Función que permite a varios usuarios trabajar en el mismo documento simultáneamente a través de la nube.

Nube: Almacenamiento remoto de documentos accesible desde cualquier dispositivo con conexión a internet (ej. Google Drive, OneDrive).



PDF (Portable Document Format): Formato de documento que preserva el diseño original y es ampliamente utilizado para compartir archivos.

Insertar Hipervínculo: Función que permite incluir enlaces clicables en un documento para acceder a otras páginas o archivos.

Edición Colaborativa: Método de trabajo donde varias personas contribuyen simultáneamente en la creación o revisión de un documento.

Índice: Lista ordenada de términos o conceptos clave con sus ubicaciones en el documento.

Marca de Agua: Texto o imagen semitransparente que se superpone al fondo de un documento como medida de seguridad o identificación.

Copiar y Pegar: Función básica para duplicar texto o elementos y colocarlos en otra parte del documento.

Alineación: Disposición del texto o elementos dentro de una página, ya sea a la izquierda, derecha, centrado o justificado.

Tabulación: Configuración de saltos específicos en el texto para crear listas o columnas alineadas.

Combinar Documentos: Función para unir varios archivos en un solo documento.

Cita: Referencia directa a una fuente utilizada en el documento.



Bibliografía: Lista de fuentes consultadas y citadas en el documento.

Comandos Rápidos: Atajos de teclado que permiten realizar acciones comunes de manera eficiente (ej. Ctrl+C para copiar, Ctrl+S para guardar).

Diagramación: Organización visual del contenido en el documento para mejorar la presentación y legibilidad.

Pie de Imagen: Texto explicativo colocado debajo de una imagen en el documento.

Bloques de Texto: Secciones predefinidas de texto que pueden insertarse en un documento (ej. encabezados, notas al pie).

Macropágina: Documento diseñado con amplios márgenes y estilo visual para fines de impresión o publicación.

Autoformato: Función que aplica automáticamente un conjunto de estilos predefinidos al texto o párrafos del documento.

Versión del Documento: Función que permite almacenar y revisar diferentes ediciones de un documento.



Subárea Tecnologías de Información

Unidad: Fundamentos de Tecnologías de la Información

Tecnología de la Información (TI): Conjunto de herramientas, procesos y sistemas utilizados para gestionar y procesar información.

Sistema Informático: Conjunto de hardware, software, datos y usuarios que trabajan juntos para realizar tareas específicas.

Hardware: Componentes físicos de un sistema informático, como computadoras, servidores y dispositivos periféricos.

Software: Programas y aplicaciones que ejecutan instrucciones en los dispositivos de hardware.

Red Informática: Conjunto de dispositivos interconectados para compartir recursos e información.

Servidor: Computadora o programa que proporciona servicios a otros dispositivos en una red.

Cliente: Dispositivo o programa que solicita y utiliza los servicios de un servidor.

Datos: Representación de hechos, conceptos o instrucciones de manera organizada para su procesamiento.

Base de Datos: Colección organizada de datos que puede ser accesada, gestionada y actualizada.



Sistema Operativo: Software que gestiona los recursos de hardware y proporciona servicios a otras aplicaciones (ej. Windows, Linux).

Procesamiento de Datos: Conjunto de operaciones realizadas en datos para convertirlos en información útil.

Almacenamiento: Capacidad de guardar datos e información en dispositivos físicos o en la nube.

Computación en la Nube (Cloud Computing): Uso de servicios de TI a través de internet, como almacenamiento, procesamiento y aplicaciones.

Redes Locales (LAN): Redes que conectan dispositivos en un área limitada, como una oficina o edificio.

Protocolo: Conjunto de reglas que regulan cómo se comunican los dispositivos en una red (ej. TCP/IP).

IP (Dirección de Protocolo de Internet): Identificación única de un dispositivo en una red.

Firewall: Herramienta de seguridad que monitorea y controla el tráfico de red según reglas predefinidas.

Ciberseguridad: Prácticas para proteger sistemas, redes y datos frente a amenazas digitales.

Criptografía: Técnica de codificación de datos para protegerlos de accesos no autorizados.

Virtualización: Tecnología que permite ejecutar múltiples sistemas operativos o aplicaciones en una sola máquina física.



Internet: Red global de computadoras interconectadas que permite la comunicación y el intercambio de información.

Intranet: Red privada utilizada dentro de una organización para compartir información y recursos.

Extranet: Red privada que permite el acceso controlado a usuarios externos.

Algoritmo: Serie de pasos lógicos diseñados para resolver un problema o realizar una tarea.

Bits y Bytes: Unidades básicas de información digital, donde 1 byte equivale a 8 bits.

Procesador: Componente de hardware que ejecuta instrucciones y realiza cálculos en un sistema informático.

Memoria RAM: Memoria de acceso rápido que permite al sistema procesar datos temporalmente.

Disco Duro (HDD): Dispositivo de almacenamiento de datos permanente en un sistema informático.

SSD (Unidad de Estado Sólido): Dispositivo de almacenamiento más rápido y duradero que un disco duro tradicional.

Backup (Copia de Seguridad): Duplicado de datos importantes para protegerlos en caso de pérdida o daño.

Virtual Private Network (VPN): Tecnología que asegura conexiones en línea creando un túnel de datos encriptado.



Interfaz de Usuario: Punto de interacción entre el usuario y un sistema informático o aplicación.

Código Fuente: Conjunto de instrucciones escritas en un lenguaje de programación.

Inteligencia Artificial (IA): Tecnología que permite a las máquinas realizar tareas similares a las humanas, como el reconocimiento de patrones.

Machine Learning: Rama de la IA que permite a los sistemas aprender y mejorar a partir de datos.

IoT (Internet de las Cosas): Red de dispositivos conectados que recopilan y comparten datos en tiempo real.

Red 5G: Quinta generación de redes móviles que ofrece velocidades de conexión más rápidas y menor latencia.

Interoperabilidad: Capacidad de diferentes sistemas y aplicaciones para comunicarse y trabajar juntos de manera efectiva.

Normas ISO: Estándares internacionales que aseguran la calidad, seguridad y eficiencia de sistemas y procesos informáticos.

TIC (Tecnologías de la Información y Comunicación): Tecnologías que integran hardware, software y redes para facilitar el acceso a la información y la comunicación.



Unidad: Fundamentos de Ciberseguridad

Ciberseguridad: Conjunto de prácticas, tecnologías y procesos diseñados para proteger sistemas, redes y datos frente a ataques digitales.

Amenaza: Potencial riesgo o peligro que podría comprometer la seguridad de un sistema o información.

Vulnerabilidad: Debilidad en un sistema o software que puede ser explotada por una amenaza.

Ataque Cibernético: Acción maliciosa destinada a dañar, robar o interrumpir sistemas informáticos.

Malware: Software malicioso diseñado para infiltrarse, dañar o explotar un sistema (ej. virus, troyanos, ransomware).

Phishing: Técnica de engaño que busca obtener información confidencial simulando ser una entidad confiable.

Ransomware: Tipo de malware que cifra los datos de una víctima y exige un rescate para liberarlos.

Spyware: Software que recopila información de un sistema sin el conocimiento del usuario.

Firewall: Herramienta de seguridad que filtra el tráfico de red para bloquear accesos no autorizados.



Antivirus: Programa que detecta, previene y elimina software malicioso de un sistema.

Ingeniería Social: Manipulación psicológica utilizada para obtener acceso no autorizado o información confidencial.

Autenticación: Proceso de verificar la identidad de un usuario antes de permitirle acceso a un sistema.

Autenticación Multifactor (MFA): Método de seguridad que requiere dos o más verificaciones para acceder a un sistema.

Contraseña Segura: Clave que incluye una combinación de caracteres alfanuméricos y símbolos para mayor protección.

Cifrado: Técnica que convierte datos en un formato ilegible para proteger su confidencialidad.

Llave de Cifrado: Código utilizado para cifrar y descifrar datos.

Privacidad de los Datos: Protección de la información personal y sensible contra accesos no autorizados.

Protección de Datos: Conjunto de medidas para garantizar que la información esté segura y disponible solo para usuarios autorizados.

Incidente de Seguridad: Evento que compromete la confidencialidad, integridad o disponibilidad de la información.



Resiliencia Cibernética: Capacidad de un sistema para resistir y recuperarse de incidentes de seguridad.

Seguridad en la Nube: Prácticas diseñadas para proteger datos y aplicaciones almacenados en servicios de computación en la nube.

VPN (Red Privada Virtual): Tecnología que asegura la conexión de un dispositivo a una red pública mediante un túnel encriptado.

Hacker Ético: Profesional que identifica vulnerabilidades en sistemas con el propósito de mejorarlos y protegerlos.

Penetration Testing (Pentesting): Simulación de ataques cibernéticos para evaluar la seguridad de un sistema.

Backdoor: Acceso oculto a un sistema que puede ser explotado por atacantes.

Zero-Day: Vulnerabilidad desconocida por el desarrollador del software que puede ser explotada antes de ser parcheada.

Parche de Seguridad: Actualización del software que corrige vulnerabilidades y mejora la seguridad.

SOC (Centro de Operaciones de Seguridad): Equipo dedicado a monitorear y responder a incidentes de seguridad en tiempo real.

IDS (Sistema de Detección de Intrusos): Herramienta que identifica posibles intrusiones en un sistema o red.



IPS (Sistema de Prevención de Intrusos): Tecnología que no solo detecta, sino que bloquea intrusiones en un sistema.

Normas ISO 27001: Estándar internacional para la gestión de la seguridad de la información.

GDPR (Reglamento General de Protección de Datos): Ley europea que regula la privacidad y protección de datos personales.

CISO (Chief Information Security Officer): Responsable de liderar las estrategias de ciberseguridad de una organización.

Seguridad de la Información: Disciplina que asegura la confidencialidad, integridad y disponibilidad de los datos.

Ataque DDoS (Denegación de Servicio Distribuido): Ataque que satura un sistema con tráfico para hacerlo inoperativo.

Whitelisting: Práctica de permitir solo el acceso a aplicaciones o usuarios previamente aprobados.

Blacklisting: Práctica de bloquear acceso a usuarios, aplicaciones o sitios considerados no seguros.

Forense Digital: Análisis de sistemas informáticos para investigar y recopilar evidencia tras incidentes de seguridad.



**MINISTERIO DE
EDUCACIÓN PÚBLICA**

**GOBIERNO
DE COSTA RICA**

**Dirección de Educación
Técnica y Capacidades
Emprendedoras**

Ingeniería de Seguridad: Diseño de sistemas y software con medidas integradas de protección.

Monitorización de Red: Supervisión constante de tráfico y actividades en una red para identificar anomalías.



Subárea Desarrollo de componentes de software

Unidad: Herramientas Lógicas y robótica

Lógica: Estudio de principios de razonamiento válido y estructuras para resolver problemas.

Algoritmo: Conjunto de pasos ordenados que describen cómo resolver un problema o realizar una tarea.

Diagrama de Flujo: Representación gráfica de un proceso o algoritmo mediante símbolos y flechas que indican el flujo de ejecución.

Pseudocódigo: Representación textual de un algoritmo que utiliza una combinación de lenguaje natural y términos de programación.

Operador Lógico: Símbolos utilizados para realizar operaciones lógicas (ej. AND, OR, NOT).

Tabla de Verdad: Herramienta que muestra todas las posibles combinaciones de valores de entrada y sus resultados lógicos.

Condicional: Declaración que permite tomar decisiones en un algoritmo basado en una condición (ej. if-else).



Ciclo: Estructura de control que repite un bloque de instrucciones mientras se cumpla una condición (ej. for, while).

Estructura Secuencial: Ejecución de instrucciones en un orden lineal, una tras otra.

Estructura Condicional: Ejecución de bloques de código dependiendo del cumplimiento de una condición específica.

Estructura Repetitiva: Bloque de instrucciones que se ejecuta varias veces según una condición definida.

Función: Bloque de código reutilizable que realiza una tarea específica y puede devolver un resultado.

Recursividad: Técnica donde una función se llama a sí misma para resolver problemas más pequeños del mismo tipo.

Lógica Proposicional: Rama de la lógica que trabaja con proposiciones o declaraciones que pueden ser verdaderas o falsas.

Lógica Booleana: Sistema de álgebra basado en valores binarios (verdadero o falso) utilizado en informática.

Variable: Espacio en memoria que almacena un valor que puede cambiar durante la ejecución de un programa.

Constante: Valor que no cambia durante la ejecución de un programa.



Arreglo: Conjunto de elementos de datos organizados en una estructura indexada.

Iteración: Repetición de un conjunto de instrucciones hasta que se cumpla una condición de salida.

Depuración: Proceso de identificar y corregir errores en un algoritmo o código.

Modelo de Decisión: Representación lógica que utiliza reglas y condiciones para tomar decisiones.

Paradigma de Programación: Enfoque o estilo utilizado para resolver problemas computacionales (ej. programación estructurada, programación orientada a objetos).

Lógica Difusa (Fuzzy Logic): Extensión de la lógica booleana que maneja valores de verdad parciales entre 0 y 1.

Simulación: Reproducción virtual de un proceso o sistema real para analizar su comportamiento.

Abstracción: Técnica que permite simplificar un problema complejo al enfocarse en los elementos esenciales.

Diagramas UML: Lenguaje visual utilizado para modelar y diseñar sistemas informáticos, incluyendo diagramas de casos de uso y secuencias.

Sistema de Reglas: Conjunto de premisas y conclusiones utilizadas para tomar decisiones basadas en lógica.

Inferencia: Proceso lógico de derivar conclusiones a partir de premisas o datos.



Resolución de Problemas: Aplicación de estrategias lógicas para identificar soluciones efectivas y eficientes.

Automatización: Uso de herramientas lógicas y algoritmos para realizar tareas repetitivas de manera autónoma.

Lenguaje de Programación: Herramienta lógica utilizada para traducir ideas o algoritmos en código ejecutable por una máquina.

Diagrama de Causa-Efecto: Herramienta lógica para identificar y analizar las causas de un problema específico.

Modelo Relacional: Representación lógica de datos organizados en tablas interconectadas mediante relaciones.

Pensamiento Computacional: Habilidad para descomponer problemas complejos y diseñar soluciones utilizando herramientas lógicas.



Unidad: Algoritmos y Diagramas de Flujo

Algoritmo: Conjunto de pasos organizados de manera lógica para resolver un problema o realizar una tarea.

Diagrama de Flujo: Representación gráfica de un algoritmo mediante símbolos que indican el flujo de procesos.

Pseudocódigo: Descripción textual de un algoritmo utilizando lenguaje natural combinado con términos de programación.

Proceso: Acción o conjunto de instrucciones realizadas dentro de un algoritmo.

Condición: Declaración que evalúa una situación para decidir qué acción tomar (ej. if-else).

Ciclo: Repetición de un bloque de instrucciones mientras se cumpla una condición (ej. while, for).

Entrada: Información inicial que se proporciona a un algoritmo para ser procesada.

Salida: Resultado o información generada por un algoritmo tras su ejecución.

Inicio/Fin: Puntos de entrada y salida de un algoritmo representados en un diagrama de flujo.

Decisión: Punto donde el flujo del algoritmo puede bifurcarse en función de una condición lógica.

Iteración: Repetición de una secuencia de pasos en un algoritmo.



Secuencia: Ejecución de instrucciones en el orden en que están definidas.

Variable: Espacio en memoria donde se almacena un valor que puede cambiar durante la ejecución de un algoritmo.

Constante: Valor fijo que no cambia durante la ejecución de un algoritmo.

Asignación: Proceso de almacenar un valor en una variable.

Símbolo de Proceso: Representado por un rectángulo en un diagrama de flujo, indica una acción o instrucción.

Símbolo de Decisión: Representado por un rombo en un diagrama de flujo, indica una bifurcación basada en una condición.

Símbolo de Entrada/Salida: Representado por un paralelogramo, indica la recepción de datos o la presentación de resultados.

Conector: Representado por un círculo, une diferentes partes de un diagrama de flujo cuando no se encuentran en la misma línea.

Flujo: Dirección del proceso en un algoritmo, representada por flechas en un diagrama de flujo.

Estructura Secuencial: Serie de pasos ejecutados uno tras otro en un algoritmo.



Estructura Condicional: Bloques de instrucciones ejecutados dependiendo del cumplimiento de una condición.

Estructura Repetitiva: Bloques de instrucciones que se ejecutan varias veces según una condición específica.

Depuración: Proceso de identificar y corregir errores en un algoritmo.

Abstracción: Técnica que simplifica problemas complejos enfocándose en los elementos esenciales.

Optimización: Mejora de un algoritmo para que sea más eficiente en tiempo o recursos.

Complejidad Algorítmica: Medida de los recursos requeridos por un algoritmo, como tiempo de ejecución o uso de memoria.

Recursividad: Técnica donde un algoritmo o función se llama a sí mismo para resolver problemas más pequeños del mismo tipo.

Paralelismo: Ejecución simultánea de múltiples instrucciones de un algoritmo para mejorar su rendimiento.

Diagrama de Causa y Efecto: Representación gráfica para identificar causas y soluciones de problemas dentro de un proceso algorítmico.

Caso de Prueba: Escenario diseñado para evaluar la validez de un algoritmo.



Errores Lógicos: Fallas en un algoritmo que producen resultados incorrectos debido a una lógica defectuosa.

Pruebas de Escritorio: Ejecución manual de un algoritmo con datos de ejemplo para verificar su funcionamiento.

Estructura Modular: División de un algoritmo en partes independientes llamadas módulos o funciones.

Simulación: Uso de algoritmos para modelar y analizar situaciones del mundo real.

Diagramas UML: Herramientas gráficas que complementan los diagramas de flujo para el diseño de sistemas y procesos.

Unidad: Software y su Ingeniería

Software: Conjunto de programas, instrucciones y datos que controlan el funcionamiento de un sistema informático.

Ingeniería de Software: Disciplina que aplica principios de ingeniería para el desarrollo, mantenimiento y gestión de software.

Ciclo de Vida del Software (SDLC): Serie de etapas que incluyen el diseño, desarrollo, prueba, implementación y mantenimiento de un sistema de software.



Requisitos de Software: Descripción detallada de las funcionalidades y restricciones que debe cumplir un sistema.

Análisis de Requisitos: Proceso de identificar, documentar y validar las necesidades del cliente o usuario final.

Diseño de Software: Fase donde se define la estructura, componentes y comportamiento del sistema antes de su implementación.

Arquitectura de Software: Organización estructural de un sistema, incluyendo sus componentes y relaciones.

Programación: Proceso de escribir código para implementar las funcionalidades de un software.

Pruebas de Software: Actividades realizadas para verificar que el software funciona correctamente y cumple con los requisitos establecidos.

Mantenimiento de Software: Actividades realizadas para actualizar, corregir y mejorar el software después de su implementación.

Despliegue: Proceso de instalar y configurar el software en el entorno de producción.

Modelo en Cascada: Enfoque de desarrollo de software lineal y secuencial, donde cada etapa depende de la anterior.



Modelo Ágil: Metodología de desarrollo iterativa que se enfoca en la entrega incremental de funcionalidades y la colaboración con el cliente.

Scrum: Marco de trabajo ágil que organiza el desarrollo en ciclos cortos llamados "sprints".

Kanban: Método visual que gestiona y optimiza el flujo de trabajo en proyectos de software.

Producto Mínimo Viable (MVP): Versión básica de un software con las funcionalidades esenciales para validarlo en el mercado.

Backend: Parte del software que maneja la lógica del negocio, almacenamiento y procesamiento de datos.

Frontend: Parte del software que interactúa directamente con el usuario, generalmente mediante interfaces gráficas.

Base de Datos: Sistema que almacena y organiza los datos utilizados por un software.

Control de Versiones: Herramienta que gestiona los cambios en el código fuente a lo largo del tiempo (ej. Git).

Integración Continua: Práctica de combinar y probar el código de manera frecuente para detectar errores rápidamente.

Entrega Continua: Proceso de automatizar el despliegue del software después de cada actualización.



Pruebas Unitarias: Evaluación de componentes individuales del software para garantizar su correcto funcionamiento.

Pruebas de Integración: Verificación de la interacción entre múltiples componentes del software.

Pruebas de Usuario: Evaluación realizada por usuarios finales para garantizar que el software cumple con sus expectativas.

Calidad de Software: Medida de la efectividad, eficiencia, usabilidad y confiabilidad de un sistema de software.

Documentación: Conjunto de materiales escritos que describen el diseño, uso y mantenimiento del software.

Lenguaje de Programación: Herramienta utilizada para escribir el código fuente del software (ej. Python, Java, C++).

Framework: Conjunto de herramientas y bibliotecas predefinidas que facilitan el desarrollo de software (ej. Angular, React, Django).

Biblioteca: Conjunto de funciones reutilizables que simplifican el desarrollo de software.

API (Interfaz de Programación de Aplicaciones): Conjunto de reglas y herramientas para que diferentes sistemas interactúen entre sí.



DevOps: Cultura y prácticas que combinan desarrollo y operaciones para entregar software más rápido y de mejor calidad.

Seguridad de Software: Prácticas para proteger el software contra accesos no autorizados y vulnerabilidades.

Errores de Software (Bugs): Fallas en el software que impiden su correcto funcionamiento.

Depuración: Proceso de identificar y corregir errores en el código de un software.

Metodología de Desarrollo: Conjunto de principios y prácticas utilizadas para organizar y gestionar el proceso de creación de software.

Cloud Computing: Uso de recursos de software y almacenamiento en servidores remotos accesibles por internet.

Escalabilidad: Capacidad de un software para manejar un aumento en la carga de trabajo o usuarios.

Usabilidad: Facilidad con la que un usuario puede interactuar con el software para completar tareas.

Portabilidad: Capacidad del software para funcionar en diferentes sistemas operativos o plataformas.

Prototipado: Creación de una versión inicial o modelo del software para evaluar su diseño o funcionalidad.



Ética en el Desarrollo de Software: Principios para garantizar que el software sea desarrollado y utilizado de manera responsable.

Unidad: Levantamiento de Requerimientos

Requerimientos: Necesidades y expectativas documentadas que un sistema debe cumplir.

Levantamiento de Requerimientos: Proceso de identificar, recopilar y documentar las necesidades y objetivos de los usuarios y partes interesadas para desarrollar un sistema o software.

Stakeholders: Personas o grupos interesados o afectados por el desarrollo del sistema, como usuarios finales, clientes y desarrolladores.

Requerimientos Funcionales: Especificaciones de las funciones que el sistema debe realizar (ej. "El sistema debe permitir al usuario iniciar sesión").

Requerimientos No Funcionales: Características de calidad del sistema, como rendimiento, seguridad o usabilidad.

Elicitación: Técnica para recopilar información sobre los requerimientos a través de entrevistas, talleres, encuestas, observaciones y análisis de documentos.

Análisis de Requerimientos: Proceso de evaluar, priorizar y refinar los requerimientos para garantizar que sean claros y factibles.



Validación de Requerimientos: Proceso de confirmar que los requerimientos recopilados son precisos, completos y cumplen con las expectativas de los stakeholders.

Especificación de Requerimientos: Documento formal que detalla los requerimientos del sistema y sirve como referencia para el diseño y desarrollo.

Priorización de Requerimientos: Clasificación de los requerimientos en función de su importancia y urgencia para el proyecto.

Diagrama de Casos de Uso: Representación visual de las interacciones entre los usuarios (actores) y las funcionalidades del sistema.

Historia de Usuario: Descripción breve de una funcionalidad desde la perspectiva del usuario final (ej. "Como usuario, quiero registrarme para acceder al contenido exclusivo").

Entrevistas: Método de recopilación de requerimientos basado en conversaciones directas con los stakeholders para entender sus necesidades.

Observación Directa: Técnica para identificar requerimientos mediante la observación de cómo los usuarios interactúan con sistemas existentes.

Cuestionarios: Herramienta para recopilar información estandarizada sobre necesidades y expectativas.



Taller de Requerimientos: Sesión colaborativa en la que los stakeholders y el equipo de desarrollo trabajan juntos para identificar y priorizar requerimientos.

Prototipo: Modelo visual o funcional inicial del sistema que ayuda a los stakeholders a comprender y validar los requerimientos.

Modelo de Negocio: Descripción de cómo opera la organización, lo que ayuda a contextualizar los requerimientos.

Análisis GAP: Comparación entre el estado actual y el estado deseado para identificar los requerimientos necesarios para cerrar esa brecha.

Diagrama de Contexto: Representación gráfica que muestra cómo el sistema interactúa con sus entornos externos.

Actores: Personas, sistemas u organizaciones que interactúan con el sistema y que tienen necesidades específicas.

Restricciones: Limitaciones técnicas, legales, financieras o de tiempo que afectan al desarrollo del sistema.

Criterios de Aceptación: Condiciones que un requerimiento o funcionalidad debe cumplir para ser aceptado por los stakeholders.



Ambigüedad: Falta de claridad en los requerimientos que puede llevar a malentendidos y errores en el desarrollo.

Matrices de Trazabilidad: Herramienta que relaciona los requerimientos con sus respectivas implementaciones y pruebas para asegurar su cumplimiento.

Requerimientos Volátiles: Necesidades que pueden cambiar durante el ciclo de vida del proyecto.

Feedback: Información proporcionada por los stakeholders sobre los requerimientos o prototipos para realizar ajustes necesarios.

Alcance del Proyecto: Definición clara de los límites del sistema y las funcionalidades que se incluirán en su desarrollo.

Documento de Requerimientos del Software (SRS): Documento que describe detalladamente los requerimientos funcionales y no funcionales del sistema.

Conflictos de Requerimientos: Discrepancias o diferencias entre las necesidades de diferentes stakeholders que deben resolverse.

Iteración: Proceso repetitivo de ajustar los requerimientos mediante validaciones y retroalimentaciones continuas.



Escenarios: Descripciones narrativas de cómo los usuarios interactuarán con el sistema en situaciones específicas.

User Persona: Representación ficticia de un usuario típico del sistema, basada en datos reales para guiar el levantamiento de requerimientos.



Unidad: Introducción a la Programación

Programación: Proceso de escribir, depurar y mantener el código que las computadoras pueden interpretar y ejecutar.

Lenguaje de Programación: Herramienta utilizada para escribir instrucciones que una computadora puede entender (ej. Python, Java, C++).

Código Fuente: Conjunto de instrucciones escritas en un lenguaje de programación que definen el comportamiento de un programa.

Compilador: Programa que traduce el código fuente a un lenguaje máquina que la computadora pueda ejecutar.

Interprete: Herramienta que ejecuta el código línea por línea sin necesidad de compilarlo previamente.

Variable: Espacio en memoria utilizado para almacenar un dato que puede cambiar durante la ejecución del programa.

Constante: Valor fijo que no puede ser modificado durante la ejecución del programa.

Tipo de Dato: Categoría que define el tipo de valores que puede contener una variable (ej. entero, cadena, booleano).



Entrada: Información proporcionada por el usuario al programa durante su ejecución.

Salida: Resultado que un programa genera tras procesar datos.

Estructura Secuencial: Ejecución de instrucciones en el orden en que aparecen en el código.

Estructura Condicional: Bloque de código que se ejecuta solo si se cumple una condición específica (ej. if-else).

Estructura Repetitiva: Bloque de código que se ejecuta varias veces mientras se cumpla una condición (ej. while, for).

Función: Bloque de código reutilizable que realiza una tarea específica y puede devolver un valor.

Parámetro: Dato que se pasa a una función para que este sea utilizado dentro de su lógica.

Lógica Booleana: Rama de la lógica que utiliza valores binarios (verdadero o falso) en las operaciones y condiciones.

Operador: Símbolo que realiza operaciones en uno o más valores (ej. +, -, *, /, ==, &&).

Condición: Declaración que evalúa una expresión para determinar si es verdadera o falsa.

Ciclo: Instrucción que permite ejecutar un conjunto de instrucciones repetidamente (ej. for, while).



Pseudocódigo: Representación textual de un programa que utiliza lenguaje natural y términos de programación.

Diagrama de Flujo: Representación gráfica de un algoritmo mediante símbolos y flechas para mostrar su flujo lógico.

Array: Conjunto de datos organizados en una estructura indexada que permite almacenar múltiples valores.

Matriz: Estructura de datos bidimensional similar a una tabla.

Depuración: Proceso de identificar y corregir errores en el código de un programa.

Errores Sintácticos: Fallos en el código causados por el incumplimiento de las reglas del lenguaje de programación.

Errores Lógicos: Fallos en el código que producen resultados incorrectos debido a errores en la lógica del programa.

Bucles Anidados: Ciclos dentro de otros ciclos que permiten realizar iteraciones múltiples.

Entrada/Salida (I/O): Operaciones de un programa para recibir datos de entrada o mostrar datos de salida.

IDE (Entorno de Desarrollo Integrado): Software que proporciona herramientas para escribir, depurar y ejecutar código (ej. Visual Studio Code, PyCharm).



Compilación: Proceso de traducción del código fuente a lenguaje máquina antes de su ejecución.

Objeto: Instancia de una clase en programación orientada a objetos, que combina datos y métodos.

Clase: Modelo que define un objeto y sus comportamientos en la programación orientada a objetos.

Herencia: Propiedad de una clase para compartir características con otra clase en la programación orientada a objetos.

Polimorfismo: Capacidad de un método para comportarse de diferentes maneras según el contexto.

Encapsulación: Concepto que oculta los detalles internos de un objeto y solo expone lo necesario para su uso.

Recursividad: Técnica en la que una función se llama a sí misma para resolver un problema.

Compilador vs Intérprete: Herramientas que traducen código; el compilador convierte todo el código antes de ejecutarlo, mientras que el intérprete lo ejecuta línea por línea.

Algoritmo: Conjunto ordenado de pasos que describen cómo resolver un problema.

Biblioteca: Conjunto de funciones y herramientas predefinidas que se pueden utilizar en un programa.

Comentarios: Líneas en el código que no se ejecutan y se usan para explicar o documentar partes del programa.



Flujo de Control: Dirección en la que se ejecutan las instrucciones de un programa dependiendo de las estructuras de control.

Tipos de Bucles: While (mientras), For (para un rango definido), y Do-While (ejecuta al menos una vez antes de verificar la condición).



Subárea Infraestructura y operaciones de servicios de software

Unidad: Virtualización y Computación en la Nube

Virtualización: Tecnología que permite crear versiones virtuales de recursos físicos, como servidores, almacenamiento o redes.

Hipervisor: Software que permite crear y gestionar máquinas virtuales en un sistema físico (ej. VMware, Hyper-V).

Máquina Virtual (VM): Sistema operativo y aplicaciones que se ejecutan de manera independiente dentro de un entorno virtualizado.

Contenedores: Unidades ligeras de virtualización que empaquetan aplicaciones junto con sus dependencias (ej. Docker).

Computación en la Nube (Cloud Computing): Uso de servicios de TI como almacenamiento, procesamiento y aplicaciones a través de internet.

Infraestructura como Servicio (IaaS): Modelo de nube que ofrece recursos de infraestructura virtualizados como servidores y almacenamiento.

Plataforma como Servicio (PaaS): Modelo de nube que proporciona plataformas completas para desarrollar, probar y desplegar aplicaciones.



Software como Servicio (SaaS): Modelo de nube que ofrece aplicaciones listas para usar a través de internet (ej. Google Workspace, Microsoft 365).

Nube Pública: Infraestructura de nube compartida por múltiples organizaciones y gestionada por un proveedor externo (ej. AWS, Azure).

Nube Privada: Infraestructura de nube dedicada a una sola organización, gestionada internamente o por un proveedor.

Nube Híbrida: Combinación de nubes públicas y privadas que trabajan juntas para ofrecer flexibilidad y optimización de recursos.

Multinube: Uso de servicios de múltiples proveedores de nube para diversificar recursos y capacidades.

Escalabilidad: Capacidad de un sistema en la nube para aumentar o disminuir recursos según la demanda.

Elasticidad: Capacidad de ajustar automáticamente los recursos asignados en función de las necesidades actuales.

Almacenamiento en la Nube: Servicio que permite guardar datos en servidores remotos accesibles a través de internet (ej. Google Drive, Dropbox).

Servidor Virtual: Instancia virtual de un servidor físico que permite ejecutar aplicaciones o servicios de manera independiente.



Red Virtual: Infraestructura de red simulada que opera sobre hardware físico para conectar recursos virtualizados.

Balanceo de Carga: Distribución del tráfico entre varios servidores para optimizar el rendimiento y la disponibilidad.

Despliegue en la Nube: Proceso de instalar y configurar aplicaciones o servicios en una infraestructura de nube.

Cómputo Sin Servidor (Serverless): Modelo en el que el proveedor de nube gestiona automáticamente la infraestructura necesaria para ejecutar aplicaciones.

Orquestación de Contenedores: Gestión automatizada de la implementación, escalado y mantenimiento de contenedores (ej. Kubernetes).

Seguridad en la Nube: Prácticas y herramientas para proteger los datos, aplicaciones y recursos en entornos de nube.

Backup en la Nube: Copias de seguridad almacenadas en servidores remotos para proteger los datos contra pérdida o daño.

Disaster Recovery as a Service (DRaaS): Solución basada en la nube para restaurar sistemas y datos en caso de fallos catastróficos.



Proveedor de Servicios en la Nube: Empresa que ofrece recursos y servicios de computación en la nube (ej. AWS, Google Cloud, Microsoft Azure).

Pay-As-You-Go: Modelo de facturación en la nube donde los clientes solo pagan por los recursos utilizados.

API en la Nube: Interfaz que permite a las aplicaciones interactuar con los servicios de la nube.

DevOps en la Nube: Prácticas de integración y entrega continua habilitadas por herramientas y servicios en la nube.

Migración a la Nube: Proceso de trasladar datos, aplicaciones y cargas de trabajo desde sistemas locales a la infraestructura en la nube.

Latencia: Tiempo que tarda en transmitirse un dato desde el cliente hasta el servidor en la nube.

Alta Disponibilidad: Característica de la nube para garantizar que los servicios estén disponibles incluso en caso de fallos.

Computación Edge: Procesamiento de datos cerca de la fuente de generación en lugar de depender exclusivamente de la nube centralizada.

Redundancia: Duplicación de recursos en la nube para garantizar la continuidad del servicio ante fallos.



Virtual Desktop Infrastructure (VDI): Tecnología que permite a los usuarios acceder a escritorios virtuales alojados en servidores remotos.

Autoservicio en la Nube: Capacidad de los usuarios para aprovisionar recursos y servicios en la nube sin intervención del proveedor.

Cloud Native: Aplicaciones diseñadas específicamente para operar de manera eficiente en entornos de nube.

Regiones y Zonas de Disponibilidad: Ubicaciones geográficas donde los proveedores de nube ofrecen servicios para garantizar proximidad y redundancia.

Compliance en la Nube: Asegurarse de que los servicios en la nube cumplan con las regulaciones legales y normativas aplicables.

Machine Learning en la Nube: Uso de servicios basados en la nube para entrenar y desplegar modelos de aprendizaje automático.

Monitorización de Recursos: Herramientas para supervisar el uso y el rendimiento de los servicios en la nube.



Unidad: Introducción a las Redes

Red de Computadoras: Conjunto de dispositivos conectados entre sí para compartir recursos e información.

Protocolo: Conjunto de reglas que regulan cómo los dispositivos se comunican en una red (ej. TCP/IP, HTTP).

TCP/IP: Modelo de comunicación estándar para las redes que define cómo se transmiten los datos.

Dirección IP: Identificación única asignada a cada dispositivo en una red, que permite su localización y comunicación.

Máscara de Subred: Configuración que define cómo se dividen las redes en subredes más pequeñas.

DNS (Sistema de Nombres de Dominio): Servicio que traduce nombres de dominio (ej. www.google.com) a direcciones IP.

DHCP (Protocolo de Configuración Dinámica de Host): Protocolo que asigna automáticamente direcciones IP a los dispositivos en una red.

LAN (Red de Área Local): Red que conecta dispositivos en un área limitada, como una oficina o edificio.

WAN (Red de Área Amplia): Red que conecta dispositivos a través de grandes distancias, como internet.

MAN (Red de Área Metropolitana): Red que conecta varias LAN dentro de una ciudad o región.



Switch: Dispositivo que conecta múltiples dispositivos en una red local y dirige el tráfico entre ellos.

Router: Dispositivo que conecta diferentes redes y dirige el tráfico entre ellas.

Hub: Dispositivo que conecta múltiples dispositivos en una red y envía datos a todos ellos indiscriminadamente.

Topología de Red: Forma en que los dispositivos están organizados en una red (ej. estrella, anillo, bus, malla).

SSID (Identificador de Conjunto de Servicios): Nombre de una red inalámbrica que permite a los dispositivos identificarse y conectarse.

Firewall: Sistema de seguridad que controla el tráfico de red para proteger contra accesos no autorizados.

Wi-Fi: Tecnología inalámbrica que permite a los dispositivos conectarse a una red mediante señales de radio.

Ethernet: Tecnología de red más común para redes de área local, que utiliza cables para la conexión.

Banda Ancha: Tipo de conexión a internet que permite transmitir grandes cantidades de datos a alta velocidad.

Ancho de Banda: Cantidad máxima de datos que se pueden transferir a través de una conexión en un tiempo dado.

Latencia: Tiempo que tarda un dato en viajar desde su origen hasta su destino en una red.



Ping: Herramienta que mide la latencia o el tiempo de respuesta entre dos dispositivos en una red.

Paquete: Unidad básica de datos que se transmite en una red.

Puerta de Enlace (Gateway): Punto de acceso que conecta una red local con otra red, como internet.

Segmentación de Red: Proceso de dividir una red en partes más pequeñas para mejorar su rendimiento y seguridad.

VLAN (Red de Área Local Virtual): Subred lógica que agrupa dispositivos dentro de una red física.

VPN (Red Privada Virtual): Tecnología que permite conexiones seguras y privadas a través de una red pública.

Dirección MAC: Identificador único asignado a cada dispositivo de red, utilizado en la capa física.

Capa OSI: Modelo conceptual de 7 capas que describe cómo los datos viajan a través de una red (Física, Enlace de Datos, Red, Transporte, Sesión, Presentación, Aplicación).

Puertos: Números que identifican servicios específicos en un dispositivo de red (ej. puerto 80 para HTTP).

ARP (Protocolo de Resolución de Direcciones): Protocolo que traduce direcciones IP a direcciones MAC en una red local.

Tráfico de Red: Flujo de datos entre dispositivos en una red.



QoS (Calidad de Servicio): Técnicas para priorizar ciertos tipos de tráfico en una red para garantizar su rendimiento.

Seguridad de Redes: Prácticas y herramientas para proteger una red contra amenazas y accesos no autorizados.

Cifrado: Proceso de codificar datos para protegerlos durante la transmisión.

Red Punto a Punto: Conexión directa entre dos dispositivos sin necesidad de un intermediario.

Red Cliente-Servidor: Modelo en el que un dispositivo (servidor) proporciona recursos a otros dispositivos (clientes).

Red Peer-to-Peer (P2P): Modelo en el que todos los dispositivos de la red comparten recursos directamente entre sí.

Proxy: Servidor intermedio que actúa como puente entre el cliente y el servidor final.

Broadcast: Envío de datos desde un dispositivo a todos los demás dispositivos en una red.

Multicast: Envío de datos desde un dispositivo a un grupo específico de dispositivos en una red.

Red Definida por Software (SDN): Enfoque que permite gestionar redes de forma centralizada mediante software.



IoT (Internet de las Cosas): Red de dispositivos interconectados que recopilan y comparten datos en tiempo real.

IPv4: Versión del Protocolo de Internet que utiliza direcciones de 32 bits.

IPv6: Versión más reciente del Protocolo de Internet que utiliza direcciones de 128 bits para abordar la escasez de direcciones IPv4.



UNDÉCIMO

Subárea Herramientas e innovación aplicadas al desarrollo de aplicaciones de software

Unidad: Oportunidades de Negocios

Emprendimiento: Proceso de identificar, desarrollar y llevar a cabo una idea de negocio para ofrecer soluciones innovadoras.

Idea de Negocio: Concepto inicial que describe un producto o servicio que satisface una necesidad en el mercado.

Oportunidad de Negocio: Situación favorable que puede ser explotada para crear un producto o servicio que genere valor.

Plan de Negocio: Documento que detalla la estrategia, objetivos, análisis de mercado, estructura y proyecciones financieras de un emprendimiento.

Modelo de Negocio: Representación de cómo una empresa crea, entrega y captura valor (ej. Canvas de Modelo de Negocio).

Propuesta de Valor: Beneficio único y diferenciador que una empresa ofrece a sus clientes.

Segmento de Clientes: Grupo específico de personas u organizaciones al que se dirige un producto o servicio.



Estudio de Mercado: Investigación que analiza las necesidades, preferencias y comportamiento de los consumidores.

Análisis FODA: Herramienta que identifica las Fortalezas, Oportunidades, Debilidades y Amenazas de un negocio.

Innovación: Proceso de crear soluciones nuevas o mejoradas para satisfacer necesidades del mercado.

Estrategia de Negocios: Plan diseñado para alcanzar los objetivos de una empresa y mantenerse competitiva en el mercado.

Mercado Meta: Público objetivo o clientes potenciales para un producto o servicio.

Canales de Distribución: Medios a través de los cuales un producto o servicio llega al cliente final.

Costo Fijo: Gastos que permanecen constantes independientemente del nivel de producción o ventas (ej. alquiler, salarios).

Costo Variable: Gastos que cambian en función del nivel de producción o ventas (ej. materia prima).

Margen de Ganancia: Diferencia entre los ingresos generados y los costos totales de un negocio.

Capital Semilla: Fondo inicial necesario para poner en marcha un emprendimiento.



Inversión: Aporte de recursos financieros, materiales o humanos para iniciar o hacer crecer un negocio.

Financiamiento: Obtención de recursos económicos para desarrollar un proyecto o negocio (ej. préstamos, inversores).

Rentabilidad: Relación entre las ganancias obtenidas y la inversión realizada en un negocio.

Socios Estratégicos: Aliados que colaboran para aportar recursos o habilidades clave a un negocio.

Competencia: Empresas o individuos que ofrecen productos o servicios similares en el mercado.

Diferenciación: Características únicas que hacen que un producto o servicio se destaque frente a la competencia.

Ecosistema Emprendedor: Entorno que fomenta el emprendimiento mediante la interacción de instituciones, recursos y cultura.

Responsabilidad Social Empresarial (RSE): Prácticas de negocio que buscan generar impacto positivo en la sociedad y el medio ambiente.

Estrategia de Marketing: Plan para promocionar y posicionar un producto o servicio en el mercado.

Validación de Ideas: Proceso de evaluar si una idea de negocio tiene potencial para ser exitosa en el mercado.



Pitch de Negocios: Presentación breve y persuasiva para captar la atención de inversores o socios.

Prototipo: Versión inicial de un producto o servicio que permite probar su funcionalidad y aceptación en el mercado.

Emprendimiento Social: Negocio que busca generar impacto positivo en la sociedad mientras es financieramente sostenible.

Emprendimiento Digital: Modelo de negocio basado en el uso de tecnologías digitales para ofrecer productos o servicios.

Factibilidad: Evaluación de si una idea de negocio es viable técnica, económica y comercialmente.

Crowdfunding: Financiación colectiva donde personas contribuyen económicamente para apoyar un proyecto o negocio.

Economía Circular: Modelo de negocio que maximiza el uso de recursos, reduce desechos y promueve la sostenibilidad.

Escalabilidad: Capacidad de un negocio para crecer en ingresos sin aumentar significativamente los costos.

Propiedad Intelectual: Derechos legales que protegen las creaciones y productos intelectuales, como marcas, patentes y diseños.



Benchmarking: Proceso de analizar las mejores prácticas de otras empresas para mejorar el desempeño del negocio.

Freelancer: Profesional independiente que ofrece servicios especializados sin estar empleado directamente por una empresa.

Lean Startup: Metodología para lanzar negocios con rapidez, minimizando riesgos y ajustándose a las necesidades del mercado.

Producto Mínimo Viable (MVP): Versión inicial de un producto con características esenciales para probar su viabilidad en el mercado.

Escucha Activa: Habilidad para entender las necesidades de los clientes o partes interesadas en el desarrollo de negocios.

Economías de Escala: Reducción de costos por unidad a medida que aumenta la producción.

Networking: Construcción de relaciones profesionales para intercambiar información y oportunidades de negocio.

Estrategia de Salida: Plan para que los emprendedores o inversores abandonen un negocio, obteniendo un retorno de su inversión.



Unidad: Modelos de Negocios

Modelo de Negocio: Representación de cómo una organización crea, entrega y captura valor, describiendo las estrategias y estructuras utilizadas.

Canvas de Modelo de Negocio: Herramienta visual que permite desarrollar y estructurar un modelo de negocio en nueve componentes clave (ej. segmentos de clientes, propuesta de valor, canales).

Propuesta de Valor: Beneficio único que un negocio ofrece a sus clientes, diferenciándolo de la competencia.

Segmento de Clientes: Grupo específico de consumidores a los que un negocio dirige su producto o servicio.

Canales: Medios a través de los cuales un producto o servicio llega a los clientes (ej. tienda física, e-commerce).

Flujo de Ingresos: Formas en que un negocio genera ingresos, como ventas directas, suscripciones o licencias.

Estructura de Costos: Identificación de los costos necesarios para operar un modelo de negocio (fijos y variables).

Recursos Clave: Activos esenciales que un negocio necesita para operar (ej. infraestructura, personal, tecnología).



Actividades Clave: Procesos y tareas fundamentales que debe realizar un negocio para entregar su propuesta de valor.

Socios Clave: Organizaciones o individuos externos que ayudan a un negocio a operar de manera eficiente (ej. proveedores, aliados estratégicos).

Innovación: Proceso de crear ideas, productos o métodos nuevos que generen valor y mejoren el desempeño del negocio.

Innovación Disruptiva: Creación de productos o servicios que cambian radicalmente una industria al ofrecer soluciones más accesibles o económicas.

Innovación Incremental: Mejoras pequeñas y continuas a productos, procesos o servicios existentes.

Innovación de Producto: Desarrollo de nuevos bienes o servicios que generan valor para los clientes.

Innovación de Proceso: Mejora en la forma en que un negocio realiza sus operaciones para aumentar eficiencia o reducir costos.

Transformación Digital: Adopción de tecnologías digitales para optimizar procesos, mejorar productos y transformar el modelo de negocio.

Design Thinking: Metodología centrada en el usuario que utiliza la creatividad para resolver problemas y diseñar soluciones innovadoras.



Lean Startup: Enfoque de creación de negocios basado en la validación rápida de ideas y la minimización de riesgos mediante iteraciones continuas.

Producto Mínimo Viable (MVP): Versión básica de un producto con características esenciales para probar su viabilidad en el mercado.

Pivot: Cambio estratégico en el enfoque de un negocio basado en el aprendizaje obtenido del mercado.

Economía Circular: Modelo de negocio que promueve la reutilización de recursos para reducir desechos y maximizar el valor.

Benchmarking: Análisis de las mejores prácticas de la industria para mejorar el rendimiento del propio negocio.

Estrategia Blue Ocean: Creación de un nuevo mercado o espacio competitivo donde la competencia es irrelevante.

Crowdsourcing: Utilización de la colaboración de un grupo amplio de personas para generar ideas, soluciones o contenido.

Crowdfunding: Modelo de financiamiento en el que una comunidad aporta fondos para apoyar un proyecto o negocio.

Escalabilidad: Capacidad de un modelo de negocio para crecer sin incrementar proporcionalmente los costos.



Sostenibilidad: Integración de prácticas responsables para minimizar el impacto ambiental y social del negocio.

Mercado de Nicho: Segmento de mercado específico y especializado con necesidades particulares.

Ecosistema de Negocios: Red de empresas, organizaciones y personas que interactúan para generar valor dentro de un sector.

Prototipo: Modelo inicial o versión preliminar de un producto que permite probar su funcionalidad o aceptación.

Economía Colaborativa: Modelo de negocio basado en compartir recursos y servicios a través de plataformas digitales (ej. Airbnb, Uber).

Intrapreneurship: Innovación y emprendimiento dentro de una organización existente, promovido por sus empleados.

Modelo Freemium: Estrategia donde se ofrece un producto básico gratuito, con opciones avanzadas de pago.

Plataforma de Dos Lados: Modelo donde un negocio conecta a dos grupos de usuarios distintos pero interdependientes (ej. compradores y vendedores en Amazon).

Modelo B2B (Business to Business): Negocios que venden productos o servicios a otras empresas.



Modelo B2C (Business to Consumer): Negocios que venden productos o servicios directamente al consumidor final.

Modelo C2C (Consumer to Consumer): Negocios donde los consumidores compran y venden entre sí (ej. eBay).

Red de Valor: Conjunto de relaciones y alianzas que un negocio utiliza para maximizar el valor generado para los clientes.

Indicadores de Rendimiento (KPIs): Métricas utilizadas para medir el éxito de un modelo de negocio o innovación.

Estrategia de Monetización: Plan para convertir usuarios o recursos en ingresos sostenibles.

Propiedad Intelectual: Protección legal de las ideas, innovaciones y productos desarrollados por un negocio.

Responsabilidad Social Empresarial (RSE): Incorporación de prácticas éticas y sostenibles en el modelo de negocio.



Unidad: Creación de Empresas

Conceptos Básicos

Emprendimiento Innovador: Creación de negocios que utilizan nuevas tecnologías o enfoques disruptivos para resolver problemas o satisfacer necesidades del mercado.

Startup Tecnológica: Empresa emergente que desarrolla soluciones tecnológicas escalables, como aplicaciones de software.

Propuesta de Valor Tecnológica: Beneficio único ofrecido por una solución de software que resuelve problemas de manera más eficiente o accesible que las opciones existentes.

MVP (Producto Mínimo Viable): Versión inicial de una aplicación de software con características esenciales para probar su viabilidad en el mercado.

Iteración: Proceso continuo de mejora y refinamiento de una aplicación basado en la retroalimentación de los usuarios.

Modelos de Negocio Aplicados al Software

Freemium: Estrategia en la que una aplicación ofrece una versión básica gratuita con funciones premium de pago (ej. Spotify, Canva).



SaaS (Software como Servicio): Modelo en el que el software se ofrece como un servicio accesible a través de internet mediante suscripción (ej. Google Workspace, Salesforce).

Licenciamiento: Venta de derechos de uso del software a clientes, ya sea con pagos únicos o recurrentes.

Marketplace: Plataforma digital que conecta a usuarios finales con proveedores de productos o servicios (ej. App Store, Google Play).

Plataforma de Dos Lados: Modelo que conecta a dos grupos de usuarios, como compradores y vendedores, utilizando software (ej. Uber, Airbnb).

Innovación y Desarrollo

Metodología Ágil: Enfoque iterativo y flexible para el desarrollo de software, como Scrum o Kanban.

Design Thinking: Metodología para diseñar aplicaciones centradas en las necesidades del usuario.

Transformación Digital: Proceso de integrar soluciones de software para modernizar y optimizar procesos empresariales.

Machine Learning: Aplicación de algoritmos en software para analizar datos y aprender automáticamente.

Interfaz de Usuario (UI): Diseño visual y funcional de cómo los usuarios interactúan con la aplicación.



Experiencia de Usuario (UX): Percepción y satisfacción del usuario al interactuar con la aplicación.

Escalabilidad: Capacidad del software y el modelo de negocio para manejar un crecimiento exponencial en usuarios o datos.

Financiamiento y Recursos

Capital Semilla: Financiación inicial para desarrollar la primera versión de una aplicación de software o poner en marcha la empresa.

Inversionistas Ángeles: Individuos que aportan capital y, a menudo, asesoría en etapas tempranas de la empresa.

Rondas de Inversión: Proceso de recaudar fondos en etapas sucesivas para financiar el desarrollo y escalamiento del software.

Crowdfunding: Obtención de financiamiento colectivo para desarrollar aplicaciones o soluciones tecnológicas.

Técnicas de Validación

Pruebas Beta: Evaluación del software por parte de usuarios reales para identificar errores y obtener retroalimentación.



Landing Pages: Páginas web diseñadas para medir el interés de los usuarios en una aplicación antes de su desarrollo completo.

Métricas de Rendimiento: Indicadores clave (KPIs) para evaluar el éxito de la aplicación, como número de descargas, usuarios activos y tasas de retención.

Aspectos Técnicos

Backend: Parte del software que gestiona la lógica del negocio, bases de datos y servidores.

Frontend: Parte del software que interactúa directamente con los usuarios, generalmente a través de interfaces gráficas.

API (Interfaz de Programación de Aplicaciones): Conjunto de reglas que permiten que diferentes aplicaciones interactúen entre sí.

DevOps: Prácticas que integran desarrollo y operaciones para garantizar la entrega continua y eficiente del software.

Nube (Cloud Computing): Uso de servidores remotos para alojar, procesar y almacenar datos de aplicaciones.

Estrategias de Marketing y Comercialización



Adquisición de Usuarios: Estrategias para atraer usuarios a la aplicación, como campañas de marketing digital o publicidad en redes sociales.

Optimización ASO (App Store Optimization): Prácticas para mejorar la visibilidad de una aplicación en tiendas de aplicaciones como Google Play o App Store.

Gamificación: Uso de elementos de juego en aplicaciones para aumentar el compromiso y la interacción del usuario.

Modelo Freemium: Ofrecimiento de funciones básicas gratuitas con opciones avanzadas de pago para monetizar la aplicación.

Regulación y Propiedad Intelectual

Propiedad Intelectual: Protección legal para software, incluyendo derechos de autor, patentes y marcas comerciales.

Licencias de Software: Normas legales que definen cómo los usuarios pueden usar la aplicación (ej. licencia de uso único o abierta).

Protección de Datos: Conformidad con regulaciones para proteger la información personal de los usuarios (ej. GDPR, CCPA).

Cultura de Innovación



Ecosistema de Startups: Entorno que fomenta la creación de empresas tecnológicas, incluyendo incubadoras, aceleradoras y redes de mentores.

Colaboración Abierta: Desarrollo de software en comunidad o mediante contribuciones externas, como proyectos de código abierto.

Intraemprendimiento: Innovación desde dentro de una organización establecida para desarrollar nuevas aplicaciones o servicios digitales.

Sostenibilidad y Futuro

Sostenibilidad Tecnológica: Creación de software que optimice el uso de recursos digitales y minimice el impacto ambiental.

Inteligencia Artificial (IA): Integración de IA para automatizar procesos y personalizar experiencias dentro de aplicaciones.

IoT (Internet de las Cosas): Desarrollo de software para conectar dispositivos inteligentes a través de redes.



Unidad: Plan de Vida

Plan de Vida: Proceso de reflexión y estructuración de objetivos personales, profesionales y sociales a lo largo del tiempo.

Autoconocimiento: Capacidad de identificar y comprender las propias fortalezas, debilidades, valores y metas.

Visión: Imagen clara y motivadora de cómo una persona desea ser y lo que quiere lograr en el futuro.

Misión Personal: Declaración que describe el propósito y los principios que guían la vida de una persona.

Objetivos: Metas específicas que una persona desea alcanzar en diferentes áreas de su vida.

Metas a Corto Plazo: Objetivos que se pueden lograr en un periodo de tiempo cercano, generalmente entre semanas o meses.

Metas a Mediano Plazo: Logros planeados para un periodo intermedio, generalmente entre uno y cinco años.

Metas a Largo Plazo: Objetivos que requieren un compromiso prolongado, usualmente más de cinco años.

Prioridades: Aspectos o actividades que una persona considera más importantes en su vida.

Valores Personales: Principios fundamentales que guían las decisiones y acciones de una persona.



Planificación: Proceso de definir pasos concretos para alcanzar los objetivos personales o profesionales.

Propósito de Vida: Razón de ser que da sentido y dirección a las acciones y decisiones de una persona.

Resiliencia: Capacidad de adaptarse y superar desafíos o adversidades en la vida.

Motivación: Fuerza interna que impulsa a una persona a actuar y alcanzar sus metas.

Toma de Decisiones: Proceso de elegir entre diferentes opciones para alcanzar un objetivo deseado.

Habilidades Blandas: Competencias personales como la comunicación, liderazgo y trabajo en equipo, esenciales para el desarrollo integral.

Autoestima: Percepción y valoración que una persona tiene sobre sí misma.

Bienestar: Estado de satisfacción y equilibrio emocional, físico y social.

Propósito Profesional: Metas relacionadas con la carrera o trayectoria laboral de una persona.

Educación Continua: Compromiso con el aprendizaje y desarrollo de habilidades a lo largo de la vida.

Plan Financiero: Estrategia para gestionar ingresos, gastos y ahorros con el fin de cumplir metas económicas.

Equilibrio Vida-Trabajo: Lograr armonía entre las responsabilidades laborales y personales.



Evaluación Personal: Análisis periódico de logros, desafíos y áreas de mejora en relación con el plan de vida.

Mentalidad de Crecimiento: Actitud que considera que las habilidades y talentos se pueden desarrollar con esfuerzo y aprendizaje.

Hábitos: Comportamientos regulares que contribuyen al desarrollo y cumplimiento de objetivos.

Autodisciplina: Capacidad de mantener el enfoque y esfuerzo constante para lograr metas personales o profesionales.

Red de Apoyo: Grupo de personas (familia, amigos, mentores) que brinda orientación y soporte en el desarrollo personal.

Adaptabilidad: Habilidad para ajustarse a los cambios y aprovechar nuevas oportunidades.

Gestión del Tiempo: Habilidad para organizar y priorizar actividades de manera eficiente.

Zona de Confort: Estado en el que una persona se siente segura, pero que puede limitar el crecimiento personal.

Proactividad: Capacidad de tomar iniciativa y actuar para influir en los resultados de la vida.

Mindfulness: Práctica de estar plenamente presente en el momento para mejorar el bienestar y la claridad mental.



Éxito Personal: Logro de metas y objetivos alineados con los valores y propósitos de vida de una persona.

Autogestión: Capacidad de planificar, ejecutar y controlar las propias acciones para alcanzar objetivos.

Plan de Acción: Estrategia detallada que define los pasos necesarios para alcanzar una meta específica.

Reflexión Personal: Proceso de analizar experiencias pasadas para aprender y ajustar decisiones futuras.

Propósito Social: Metas relacionadas con el impacto positivo en la comunidad o el entorno.

Legado: Contribución o impacto duradero que una persona desea dejar en el mundo.

Autonomía: Capacidad de tomar decisiones y actuar de manera independiente.

Gratitud: Actitud de aprecio hacia las experiencias, logros y relaciones positivas en la vida.



Subárea Desarrollo de Componentes

Unidad: Desarrollo Ágil del Software

Desarrollo Ágil: Metodología iterativa y flexible para la creación de software, enfocada en la colaboración y entrega continua de valor.

Sprint: Iteración de trabajo en un marco ágil, generalmente de 1 a 4 semanas, en la que se entrega una funcionalidad completa.

Scrum: Marco de trabajo ágil que organiza el desarrollo en ciclos cortos y se centra en la colaboración del equipo.

Kanban: Método visual para gestionar el flujo de trabajo y priorizar tareas en un proyecto ágil.

Backlog del Producto: Lista priorizada de funcionalidades, tareas y mejoras que se desean implementar en el software.

Product Owner (PO): Persona responsable de definir y priorizar los requerimientos del producto desde la perspectiva del cliente.

Scrum Master: Facilitador del equipo que asegura la correcta aplicación de las prácticas ágiles y elimina obstáculos en el proceso.



Equipo de Desarrollo: Grupo de profesionales multidisciplinarios responsables de diseñar, construir y probar el software.

Revisión del Sprint: Reunión al final de cada sprint donde el equipo muestra el trabajo completado a los stakeholders.

Retrospectiva del Sprint: Sesión de análisis donde el equipo reflexiona sobre lo que funcionó bien y lo que puede mejorar en el próximo sprint.

Historia de Usuario: Descripción breve de una funcionalidad del software desde la perspectiva del usuario final (ej. "Como usuario, quiero registrarme para acceder al contenido exclusivo").

Criterios de Aceptación: Condiciones que una funcionalidad debe cumplir para ser aceptada como completada.

Incremento: Producto funcional entregado al final de cada sprint, que añade valor al sistema existente.

MVP (Producto Mínimo Viable): Versión básica de un software con las características esenciales para validar su viabilidad en el mercado.

Priorización: Proceso de determinar el orden de importancia de las tareas o funcionalidades del backlog.

Despliegue Continuo: Práctica de liberar versiones actualizadas del software con regularidad para los usuarios finales.



Integración Continua: Proceso automatizado de combinar el trabajo de todos los desarrolladores en un repositorio común para detectar errores rápidamente.

Entrega Continua: Extensión de la integración continua que asegura que el software esté siempre listo para ser desplegado.

Iteración: Ciclo de desarrollo en el que se construye, prueba y mejora una parte del software.

DevOps: Cultura que combina desarrollo y operaciones para automatizar y optimizar la entrega del software.

Planificación del Sprint: Reunión donde el equipo define qué tareas se completarán en el próximo sprint.

Tablero Kanban: Herramienta visual que organiza las tareas en columnas según su estado (ej. por hacer, en progreso, completado).

Daily Stand-Up (Reunión Diaria): Reunión breve donde el equipo sincroniza su progreso, identifica bloqueos y planifica el día.

Feedback: Retroalimentación proporcionada por los stakeholders o usuarios para mejorar el software.

Iterativo e Incremental: Principio de desarrollo que combina la mejora continua con la entrega de valor en pequeñas partes funcionales.



Burndown Chart: Gráfico que muestra la cantidad de trabajo restante en un sprint, ayudando a monitorear el progreso.

Definición de Hecho (Definition of Done): Lista de criterios que una tarea o funcionalidad debe cumplir para considerarse terminada.

Epic: Gran funcionalidad o conjunto de historias de usuario que se divide en tareas más pequeñas para facilitar su desarrollo.

Spike: Tarea exploratoria para investigar una tecnología, riesgo o funcionalidad antes de su implementación.

Technical Debt (Deuda Técnica): Trabajo adicional que surge cuando se toman atajos en el desarrollo que deben ser corregidos en el futuro.

Pair Programming: Técnica donde dos desarrolladores trabajan juntos en el mismo código para mejorar la calidad y compartir conocimiento.

Refinamiento del Backlog: Revisión y ajuste continuo del backlog para asegurar que las tareas estén actualizadas y priorizadas.

Prototipado Rápido: Creación de versiones preliminares del software para validar ideas antes de su desarrollo completo.

Pipeline de CI/CD: Flujo automatizado que abarca la integración, prueba y despliegue continuo del software.



Automatización de Pruebas: Uso de scripts y herramientas para probar el software de forma continua y rápida.

Velocidad del Equipo: Métrica que mide la cantidad de trabajo completado en un sprint.

Historial de Cambios: Registro de modificaciones realizadas al software durante el ciclo de desarrollo.

Pruebas de Regresión: Verificación de que las nuevas actualizaciones no afecten negativamente las funcionalidades existentes.

Release Management: Proceso de planificar, programar y controlar la entrega de versiones del software.

Stakeholders: Personas interesadas o afectadas por el desarrollo del software, como clientes, usuarios y patrocinadores.

Evolución Adaptativa: Capacidad del equipo para ajustar el alcance y las prioridades según las necesidades cambiantes del proyecto.



Unidad: Calidad del Software

Calidad del Software: Grado en el que un software cumple con los requisitos funcionales, no funcionales y las expectativas del usuario.

Requisitos Funcionales: Funcionalidades específicas que el software debe cumplir (ej. autenticación, generación de reportes).

Requisitos No Funcionales: Características de calidad como rendimiento, seguridad, usabilidad y escalabilidad.

Aseguramiento de Calidad (QA): Conjunto de actividades diseñadas para garantizar que el software cumple con los estándares de calidad.

Pruebas de Software: Proceso de evaluar el software para identificar errores y verificar su correcto funcionamiento.

Pruebas Unitarias: Verificación de pequeños componentes del software de forma individual.

Pruebas de Integración: Evaluación de cómo interactúan los diferentes módulos o componentes del software.

Pruebas de Sistema: Validación del software completo para garantizar que cumple con los requisitos establecidos.



Pruebas de Usuario: Evaluación realizada por usuarios finales para verificar que el software satisface sus necesidades.

Pruebas de Rendimiento: Evaluación del software para medir su velocidad, capacidad de respuesta y estabilidad bajo carga.

Pruebas de Seguridad: Evaluación del software para identificar y mitigar vulnerabilidades que puedan comprometer la información.

Pruebas de Regresión: Verificación de que las modificaciones realizadas al software no afecten negativamente las funcionalidades existentes.

Defectos: Errores en el software que causan un funcionamiento incorrecto o inesperado.

Depuración: Proceso de identificar, analizar y corregir defectos en el código del software.

Control de Calidad (QC): Inspección directa del producto final para garantizar que cumpla con los estándares de calidad.

Criterios de Aceptación: Condiciones específicas que el software debe cumplir para ser aceptado como terminado.

Documentación de Pruebas: Registro de los casos de prueba, resultados y hallazgos durante el proceso de evaluación.



Trazabilidad: Relación entre los requisitos, su implementación y las pruebas realizadas para verificarlos.

ISO 25010: Norma internacional que define las características de calidad del software, como funcionalidad, confiabilidad y mantenibilidad.

Mantenibilidad: Capacidad del software para ser modificado o actualizado fácilmente.

Confiabilidad: Grado en el que un software puede realizar sus funciones sin fallar durante un periodo específico.

Portabilidad: Facilidad con la que el software puede trasladarse y ejecutarse en diferentes entornos o plataformas.

Usabilidad: Facilidad de uso y aprendizaje del software para los usuarios finales.

Escalabilidad: Capacidad del software para manejar un aumento en el número de usuarios o carga de trabajo sin afectar su rendimiento.

Automatización de Pruebas: Uso de herramientas para ejecutar pruebas de software de manera automática y repetitiva.

Casos de Prueba: Conjunto de condiciones o escenarios diseñados para verificar una funcionalidad específica del software.



Ciclo de Vida del Desarrollo de Software (SDLC): Serie de etapas en el desarrollo del software que incluye análisis, diseño, desarrollo, pruebas y mantenimiento.

Integración Continua (CI): Práctica de combinar y probar el código con regularidad para detectar problemas de calidad rápidamente.

Entrega Continua (CD): Automatización de la liberación de nuevas versiones del software para garantizar calidad y rapidez.

Métricas de Calidad: Indicadores utilizados para medir y evaluar la calidad del software (ej. defectos por línea de código, tiempo de respuesta).

Mejora Continua: Proceso de identificar y aplicar cambios para aumentar la calidad del software de forma iterativa.

Ciclo Deming (PDCA): Metodología de mejora continua que sigue las fases Planificar, Hacer, Verificar y Actuar.

Test-Driven Development (TDD): Técnica de desarrollo donde se escriben pruebas antes de implementar el código.

Análisis Estático: Evaluación del código fuente sin ejecutarlo para identificar problemas de calidad.



Análisis Dinámico: Evaluación del software en ejecución para identificar problemas relacionados con el rendimiento o comportamiento.

Auditoría de Calidad: Revisión independiente de los procesos y productos de software para verificar que cumplan con estándares establecidos.

Benchmarking: Comparación del software con estándares de la industria o productos similares para evaluar su calidad.

Metodologías Ágiles: Prácticas que incluyen pruebas continuas y retroalimentación para garantizar la calidad en cada iteración de desarrollo.

Plan de Pruebas: Documento que detalla la estrategia, alcance y cronograma para evaluar la calidad del software.

Refactorización: Proceso de mejorar el diseño interno del código sin alterar su funcionalidad externa.

Deuda Técnica: Compromisos a corto plazo en calidad que requieren trabajo adicional para ser corregidos en el futuro.

Conformidad Normativa: Cumplimiento de regulaciones y estándares en el desarrollo de software (ej. GDPR, ISO).



Unidad: Programación Orientada a Objetos (POO)

Programación Orientada a Objetos (POO): Paradigma de programación que organiza el software en objetos que combinan datos y comportamientos.

Clase: Plantilla o modelo que define las propiedades (atributos) y comportamientos (métodos) de un objeto.

Objeto: Instancia de una clase que contiene valores específicos para sus atributos y puede ejecutar sus métodos.

Atributo: Propiedad o característica de una clase que almacena información sobre un objeto (ej. nombre, edad).

Método: Función dentro de una clase que define el comportamiento de un objeto.

Instanciación: Proceso de crear un objeto a partir de una clase.

Encapsulación: Principio que restringe el acceso directo a los atributos y métodos de una clase, proporcionando control a través de métodos públicos.

Abstracción: Proceso de ocultar los detalles internos de una clase y exponer solo las funcionalidades esenciales.



Herencia: Mecanismo que permite que una clase (subclase) derive propiedades y métodos de otra clase (superclase).

Polimorfismo: Capacidad de un método o función para comportarse de manera diferente según el objeto o contexto en el que se utiliza.

Sobrecarga: Definición de múltiples métodos con el mismo nombre pero diferentes parámetros en la misma clase.

Sobrescritura: Modificación de un método heredado de la clase base en una clase derivada para cambiar su comportamiento.

Constructor: Método especial en una clase que se ejecuta automáticamente cuando se crea un objeto, inicializando sus atributos.

Destructor: Método especial que se ejecuta cuando un objeto deja de ser necesario y libera recursos asociados.

Clase Abstracta: Clase que no se puede instanciar directamente y sirve como base para otras clases.

Interfaz: Conjunto de métodos abstractos que deben ser implementados por una clase que hereda de ella.

Modificadores de Acceso: Palabras clave que controlan la visibilidad de los atributos y métodos de una clase (ej. público, privado, protegido).



Propiedades (Getters y Setters): Métodos utilizados para acceder y modificar los atributos privados de una clase.

Agregación: Relación entre objetos en la que uno contiene referencias a otros, pero estos pueden existir independientemente.

Composición: Relación entre objetos donde uno depende completamente del otro para su existencia.

Asociación: Relación entre clases que define cómo interactúan sus objetos sin una dependencia fuerte.

Diagrama UML (Unified Modeling Language): Herramienta visual para modelar clases, objetos y relaciones en POO.

Instancia: Representación concreta de una clase en la memoria.

Método Estático: Método que pertenece a la clase en lugar de a los objetos creados a partir de ella y puede llamarse sin instanciar.

Variable Estática: Atributo que pertenece a la clase y es compartido por todas las instancias de esa clase.

Método Final: Método que no puede ser sobrescrito en las clases derivadas.

Clase Final: Clase que no puede ser heredada por otras clases.



Casting: Conversión de un objeto de una clase a otra dentro de una jerarquía de herencia.

Propagación de Excepciones: Envío de errores detectados a métodos superiores para su manejo adecuado.

Serialización: Proceso de convertir un objeto en un formato que se pueda almacenar o transmitir, como JSON o XML.

Deserialización: Proceso inverso de la serialización, en el que se convierte un formato almacenado en un objeto de la clase original.

Iteradores: Herramientas que permiten recorrer elementos de una colección o estructura de datos.

Clases Genéricas: Clases que permiten trabajar con cualquier tipo de dato sin especificarlo hasta que se instancien.

Interfaces Gráficas de Usuario (GUI): Aplicaciones desarrolladas con POO que presentan elementos visuales interactivos para los usuarios.

Eventos: Acciones que pueden ser manejadas por objetos de una clase en respuesta a interacciones del usuario o cambios en el sistema.

Patrón de Diseño: Soluciones reutilizables para problemas comunes en el desarrollo de software orientado a objetos (ej. Singleton, Factory, Observer).



Singleton: Patrón de diseño que asegura que una clase tenga una única instancia en todo el programa.

Factory: Patrón que delega la creación de objetos a un método o clase especializada.

Observer: Patrón en el que un objeto (observador) se actualiza automáticamente cuando otro objeto (sujeto) cambia.

Inversión de Dependencia: Principio que sugiere que las clases de alto nivel no deben depender de clases de bajo nivel, sino de abstracciones.

Coherencia de Clase: Medida de qué tan relacionadas y enfocadas están las responsabilidades de una clase.

Acoplamiento: Grado de dependencia entre clases; se busca mantenerlo bajo para facilitar la reutilización y mantenimiento.



Unidad: Estructura de Datos

Estructura de Datos: Forma de organizar y almacenar datos para facilitar su acceso y manipulación eficiente en aplicaciones de software.

Lista: Colección ordenada de elementos donde cada uno puede ser accedido mediante un índice (ej. lista enlazada, lista doblemente enlazada).

Arreglo (Array): Estructura de datos que almacena elementos de un mismo tipo en posiciones contiguas de memoria.

Pila (Stack): Estructura de datos LIFO (Last In, First Out), donde el último elemento en entrar es el primero en salir.

Cola (Queue): Estructura de datos FIFO (First In, First Out), donde el primer elemento en entrar es el primero en salir.

Cola de Prioridad: Variante de la cola en la que cada elemento tiene una prioridad y el de mayor prioridad se procesa primero.

Lista Enlazada: Estructura en la que los elementos (nodos) están conectados mediante punteros, permitiendo inserciones y eliminaciones dinámicas.



Lista Doblemente Enlazada: Variante de lista enlazada en la que cada nodo tiene referencias al siguiente y al anterior nodo.

Árbol: Estructura jerárquica de datos donde cada nodo tiene un valor y enlaces a nodos "hijos".

Árbol Binario: Árbol en el que cada nodo tiene como máximo dos hijos, llamados "izquierdo" y "derecho".

Árbol de Búsqueda Binaria (BST): Árbol binario donde los nodos están organizados de manera que el valor de la izquierda es menor que el nodo raíz y el de la derecha es mayor.

Árbol AVL: Árbol de búsqueda binaria que se equilibra automáticamente tras inserciones y eliminaciones para mantener la eficiencia.

Árbol Rojo-Negro: Árbol balanceado que asegura que la profundidad de cualquier rama no sea significativamente mayor que la de otra.

Árbol N-ario: Árbol donde cada nodo puede tener hasta "N" hijos.

Heap: Árbol binario completo utilizado para implementar colas de prioridad, donde el nodo raíz es el valor máximo (Max Heap) o mínimo (Min Heap).

Grafo: Conjunto de nodos (vértices) conectados por aristas, que pueden ser dirigidas o no dirigidas.

Grafo Dirigido: Grafo donde las aristas tienen una dirección específica entre nodos.



Grafo No Dirigido: Grafo donde las aristas no tienen dirección, permitiendo una conexión bidireccional.

Grafo Ponderado: Grafo donde cada arista tiene un peso asociado, utilizado en problemas de optimización.

Hashing: Técnica que asigna valores únicos a claves mediante una función hash para optimizar búsquedas.

Tabla Hash: Estructura de datos que almacena pares clave-valor y utiliza una función hash para acceder a los elementos.

Resolución de Colisiones: Métodos para manejar situaciones donde dos claves generan el mismo valor hash (ej. encadenamiento, direccionamiento abierto).

Diccionario: Estructura de datos que almacena pares clave-valor, comúnmente utilizada para búsquedas rápidas.

Conjunto (Set): Colección de elementos únicos donde no se permiten duplicados.

Mapa (Map): Estructura de datos que asocia claves únicas con valores específicos, similar a un diccionario.

Matriz (Matrix): Estructura bidimensional que organiza datos en filas y columnas, comúnmente utilizada en álgebra y gráficos computacionales.

Trie: Estructura de datos especializada para almacenar y buscar cadenas de texto, útil en aplicaciones como autocompletado.



Algoritmo de Ordenamiento: Métodos para organizar elementos en una estructura de datos (ej. burbuja, inserción, selección, rápido, fusión).

Búsqueda Binaria: Algoritmo eficiente para encontrar un elemento en una lista ordenada, dividiendo el rango de búsqueda en cada paso.

Búsqueda Lineal: Método de búsqueda que recorre secuencialmente todos los elementos de una lista hasta encontrar el deseado.

Recursividad: Técnica donde una función se llama a sí misma para resolver problemas mediante la división en subproblemas más pequeños.

Complejidad Temporal: Medida de la cantidad de tiempo que un algoritmo requiere en función del tamaño de la entrada.

Complejidad Espacial: Medida de la cantidad de memoria que un algoritmo utiliza durante su ejecución.

Estructuras Dinámicas: Estructuras de datos que pueden cambiar su tamaño durante la ejecución (ej. listas enlazadas, pilas dinámicas).

Estructuras Estáticas: Estructuras de datos con tamaño fijo definido al inicio de la ejecución (ej. arreglos).

Buffer Circular: Variante de una cola que reutiliza la memoria de manera cíclica, evitando el desperdicio de espacio.



Árbol de Decisión: Representación de decisiones y sus posibles resultados, útil en análisis y aprendizaje automático.

Árbol de Segmentación: Tipo de árbol utilizado para representar intervalos y realizar consultas rápidas sobre ellos.

Set Disjoint: Estructura para representar particiones de conjuntos, utilizada en algoritmos de grafos como Kruskal.

Cola Asíncrona: Cola diseñada para manejar procesos concurrentes en aplicaciones multihilo o distribuidas.

Cache: Almacenamiento temporal que permite acceder rápidamente a datos utilizados con frecuencia.

Memoria Asociativa: Técnica que permite buscar datos en estructuras de forma directa, sin recorrerlas completamente.



Subárea Bases de datos para proyectos de software

Unidad: Introducción a los Datos

Dato: Unidad básica de información que se puede almacenar, procesar y analizar en el desarrollo de software.

Información: Conjunto de datos procesados y organizados que tienen un significado para los usuarios o sistemas.

Tipo de Dato: Clasificación que define el tipo de valor que un dato puede representar (ej. entero, flotante, cadena, booleano).

Variable: Espacio en memoria que almacena un dato cuyo valor puede cambiar durante la ejecución de un programa.

Constante: Espacio en memoria que almacena un dato cuyo valor no cambia durante la ejecución del programa.

Estructura de Datos: Forma de organizar y almacenar datos en una aplicación para optimizar su uso (ej. listas, pilas, colas).

Arreglo (Array): Colección de datos del mismo tipo almacenados en ubicaciones contiguas de memoria.



Diccionario: Estructura de datos que almacena pares clave-valor para búsquedas rápidas.

Base de Datos: Sistema organizado para almacenar y gestionar datos estructurados.

CRUD: Operaciones básicas para trabajar con datos: Crear, Leer, Actualizar y Eliminar.

SQL (Structured Query Language): Lenguaje estándar para manipular y consultar bases de datos relacionales.

Datos Estructurados: Datos organizados en un formato definido, como tablas con filas y columnas.

Datos No Estructurados: Datos sin un formato definido, como imágenes, videos o correos electrónicos.

Datos Semi-Estructurados: Datos con cierta organización pero sin una estructura estricta, como JSON o XML.

JSON (JavaScript Object Notation): Formato de texto ligero para almacenar e intercambiar datos en aplicaciones web.

XML (Extensible Markup Language): Formato de texto que organiza datos en una estructura jerárquica mediante etiquetas.

CSV (Comma-Separated Values): Formato de archivo que almacena datos tabulares en texto plano separados por comas.



Normalización de Datos: Proceso para organizar una base de datos, eliminando redundancias y mejorando su integridad.

Denormalización: Técnica para optimizar consultas en bases de datos al duplicar ciertos datos para evitar uniones complejas.

Modelo Relacional: Representación de datos mediante tablas interrelacionadas en una base de datos.

Modelo NoSQL: Base de datos no relacional diseñada para manejar grandes volúmenes de datos no estructurados o semi-estructurados.

Big Data: Conjunto de datos demasiado grandes o complejos para ser procesados por métodos tradicionales.

Extracción de Datos: Proceso de recuperar datos de una fuente para su análisis o procesamiento.

Almacenamiento en Caché: Técnica para guardar temporalmente datos frecuentes en memoria para acelerar su acceso.

Flujo de Datos: Movimiento de datos entre sistemas, aplicaciones o componentes en un entorno de software.

Búsqueda de Datos: Técnica para encontrar elementos específicos en una colección de datos.

Filtro de Datos: Proceso para seleccionar o excluir datos basados en criterios específicos.



Metadatos: Información descriptiva sobre los datos, como fecha de creación, formato o propietario.

Índice: Estructura que mejora la velocidad de búsqueda y recuperación de datos en bases de datos.

Data Pipeline: Serie de procesos que automatizan la recopilación, transformación y almacenamiento de datos.

ETL (Extract, Transform, Load): Proceso para extraer datos de una fuente, transformarlos para análisis y cargarlos en un sistema de almacenamiento.

Integridad de Datos: Garantía de que los datos son precisos, consistentes y confiables.

Validación de Datos: Proceso para asegurar que los datos cumplen con ciertos criterios o formatos esperados.

Transformación de Datos: Proceso para convertir datos en un formato adecuado para su análisis o uso.

Visualización de Datos: Representación gráfica de datos para facilitar su interpretación y análisis.

Series Temporales: Conjunto de datos organizados en función del tiempo, como temperaturas diarias o precios por hora.

Datos Encriptados: Información codificada para protegerla contra accesos no autorizados.



API de Datos: Interfaz que permite a las aplicaciones acceder a datos de otros sistemas de manera controlada.

Data Warehouse: Almacenamiento centralizado diseñado para análisis y consultas de grandes volúmenes de datos.

Data Lake: Repositorio que almacena datos en su formato bruto y original, estructurados y no estructurados.

Analítica de Datos: Proceso de examinar datos para extraer información útil y patrones.

Aprendizaje Automático: Uso de algoritmos para analizar datos y hacer predicciones basadas en patrones.

Transformación Digital: Uso de datos y tecnologías para optimizar procesos y crear nuevos modelos de negocio.



Unidad: Diagramas, Estructuras y Modelos de Bases de Datos

Base de Datos: Sistema organizado para almacenar, gestionar y recuperar datos de manera eficiente.

Modelo de Datos: Representación conceptual que define cómo se estructuran, organizan y relacionan los datos.

Modelo Relacional: Estructura de datos basada en tablas (relaciones) con filas y columnas que se interconectan mediante claves.

Modelo Jerárquico: Organización de datos en una estructura de árbol donde cada nodo tiene un único nodo padre.

Modelo en Red: Representación de datos como un grafo, donde los nodos son entidades y las aristas son relaciones.

Modelo NoSQL: Enfoque no relacional que incluye bases de datos orientadas a documentos, clave-valor, gráficas y columnas.

Entidad: Objeto del mundo real que puede representarse en una base de datos, como una persona o producto.

Relación: Asociación entre dos o más entidades en un modelo de datos.



Atributo: Propiedad o característica de una entidad o relación (ej. nombre, edad, dirección).

Clave Primaria (Primary Key): Atributo único que identifica de manera exclusiva a una fila en una tabla.

Clave Foránea (Foreign Key): Atributo en una tabla que referencia la clave primaria de otra tabla, estableciendo una relación.

Cardinalidad: Indica la cantidad de entidades relacionadas entre dos conjuntos (uno a uno, uno a muchos, muchos a muchos).

Normalización: Proceso de estructurar las tablas en una base de datos para reducir la redundancia y mejorar la integridad de los datos.

Denormalización: Técnica que introduce redundancia intencionalmente para optimizar el rendimiento en consultas frecuentes.

Diagrama Entidad-Relación (ER): Representación gráfica que muestra las entidades, sus atributos y relaciones.

Diagrama Relacional: Representación tabular que detalla las relaciones entre las tablas en una base de datos relacional.

Diagrama UML: Herramienta gráfica que describe las clases, relaciones y diagramas de casos de uso en bases de datos orientadas a objetos.



Índice: Estructura que mejora la velocidad de búsqueda y recuperación de datos en una base de datos.

Esquema de Base de Datos: Diseño lógico que define la estructura y organización de una base de datos.

Integridad de Datos: Garantía de que los datos son precisos, completos y consistentes en la base de datos.

Consultas (Queries): Comandos utilizados para recuperar o manipular datos de una base de datos, generalmente mediante SQL.

Subconsultas: Consultas anidadas dentro de otras consultas para realizar operaciones más complejas.

Transacciones: Secuencia de operaciones que se ejecutan como una unidad lógica en una base de datos, asegurando consistencia y atomicidad.

ACID: Conjunto de propiedades de las transacciones en bases de datos: Atomicidad, Consistencia, Aislamiento y Durabilidad.

Almacenamiento de Datos: Espacio físico o virtual donde se guardan las bases de datos.

Data Warehouse: Repositorio centralizado optimizado para análisis de datos.

Data Mart: Subconjunto del repositorio, enfocado en un área específica de negocio.



Data Lake: Almacenamiento masivo que permite guardar datos en su formato bruto, estructurados y no estructurados.

Esquema Estrella: Diseño para bases de datos de análisis donde una tabla central (hechos) se conecta con varias tablas periféricas (dimensiones).

Esquema Copo de Nieve: Variante del esquema estrella donde las dimensiones se normalizan en múltiples tablas relacionadas.

Sharding: Técnica para dividir una base de datos en fragmentos más pequeños que se distribuyen entre múltiples servidores.

Replica de Datos: Copias de bases de datos en múltiples ubicaciones para mejorar la disponibilidad y redundancia.

Bases de Datos Orientadas a Documentos: Tipo de NoSQL que almacena datos como documentos JSON o XML.

Bases de Datos Clave-Valor: Tipo de NoSQL que organiza datos como pares clave-valor, ideal para búsquedas rápidas.

Bases de Datos Columnar: Tipo de NoSQL que almacena datos en columnas en lugar de filas, optimizado para análisis masivo.



Bases de Datos Gráficas: Modelo basado en nodos y relaciones, utilizado para representar redes y conexiones (ej. Neo4j).

ETL (Extract, Transform, Load): Proceso para extraer datos de múltiples fuentes, transformarlos y cargarlos en una base de datos.

Procedimientos Almacenados: Conjunto de comandos SQL precompilados que se almacenan en la base de datos y se ejecutan como una unidad.

Triggers: Procedimientos automáticos que se ejecutan en respuesta a eventos específicos en una tabla.

Consultas de Agregación: Operaciones que resumen datos (ej. SUM, COUNT, AVG, MAX, MIN).

Clúster de Bases de Datos: Grupo de servidores que trabajan juntos para gestionar una base de datos y mejorar su disponibilidad y rendimiento.

Optimización de Consultas: Proceso para mejorar el rendimiento de las consultas mediante índices, particiones o reestructuración del esquema.

Modelo ER Extendido: Ampliación del modelo entidad-relación que incluye conceptos como jerarquías y herencia.

Sistema de Gestión de Bases de Datos (DBMS): Software que permite crear, gestionar y consultar bases de datos (ej. MySQL, PostgreSQL, MongoDB).



Integridad Referencial: Garantía de que las relaciones entre tablas son válidas y consistentes.

Backups: Copias de seguridad periódicas de la base de datos para evitar la pérdida de datos en caso de fallos.

Unidad: Bases de Datos Relacionales

Base de Datos Relacional: Sistema de almacenamiento de datos organizado en tablas que están interrelacionadas mediante claves.

Tabla: Estructura principal en una base de datos relacional que organiza los datos en filas y columnas.

Fila (Tupla): Registro único en una tabla que representa un conjunto de datos relacionados.

Columna (Atributo): Propiedad o campo de una tabla que almacena un tipo específico de dato.

Clave Primaria (Primary Key): Atributo único que identifica de manera exclusiva cada fila en una tabla.

Clave Foránea (Foreign Key): Atributo en una tabla que referencia la clave primaria de otra tabla, creando relaciones entre ellas.

Relación: Asociación lógica entre dos o más tablas mediante claves primarias y foráneas.



Cardinalidad: Indica la cantidad de instancias que participan en una relación entre tablas (uno a uno, uno a muchos, muchos a muchos).

Modelo Relacional: Modelo de datos basado en las matemáticas de las relaciones que define la estructura de una base de datos relacional.

Esquema Relacional: Diseño lógico que describe las tablas, atributos y relaciones en una base de datos relacional.

Integridad Referencial: Restricción que asegura que las claves foráneas en una tabla siempre hagan referencia a valores válidos en la tabla relacionada.

SQL (Structured Query Language): Lenguaje estándar para consultar, manipular y gestionar bases de datos relacionales.

Consultas (Queries): Instrucciones SQL utilizadas para recuperar datos de una base de datos.

CRUD: Operaciones básicas en bases de datos: Crear (Create), Leer (Read), Actualizar (Update) y Eliminar (Delete).

Índice: Estructura de datos que optimiza la velocidad de recuperación de datos en una tabla.

Vista (View): Consulta predefinida que actúa como una tabla virtual, mostrando datos calculados o filtrados de otras tablas.



Join: Operación SQL que combina datos de dos o más tablas basándose en una condición.

Inner Join: Devuelve filas que tienen coincidencias en ambas tablas.

Left Join: Devuelve todas las filas de la tabla izquierda y las coincidencias de la derecha.

Right Join: Devuelve todas las filas de la tabla derecha y las coincidencias de la izquierda.

Full Join: Devuelve todas las filas cuando hay coincidencias o no en ambas tablas.

Normalización: Proceso de organizar una base de datos para minimizar redundancia y mejorar la integridad.

Primera Forma Normal (1NF): Asegura que cada celda tenga un único valor y que todas las filas sean únicas.

Segunda Forma Normal (2NF): Garantiza que todos los atributos dependen completamente de la clave primaria.

Tercera Forma Normal (3NF): Asegura que los atributos no dependan de atributos no clave.

Denormalización: Técnica que introduce redundancia en los datos para mejorar el rendimiento de las consultas.

Transacción: Conjunto de operaciones SQL que se ejecutan como una unidad lógica, cumpliendo las propiedades ACID.



ACID: Propiedades fundamentales de las transacciones en bases de datos relacionales:

Atomicidad: Todas las operaciones se completan o ninguna lo hace.

Consistencia: El estado de la base de datos permanece válido antes y después de la transacción.

Aislamiento: Las transacciones no afectan ni son afectadas por otras concurrentes.

Durabilidad: Los cambios realizados por una transacción se mantienen incluso después de fallos del sistema.

Subconsulta: Consulta anidada dentro de otra consulta para realizar operaciones más complejas.

Agrupación (GROUP BY): Comando SQL que organiza datos en grupos basados en valores de una o más columnas.

Funciones de Agregación: Operaciones que calculan valores resumen (ej. COUNT, AVG, SUM, MIN, MAX).

Procedimientos Almacenados: Conjunto de comandos SQL precompilados almacenados en la base de datos y reutilizables.

Triggers: Procedimientos automáticos que se ejecutan en respuesta a eventos específicos, como inserciones, actualizaciones o eliminaciones.



Relación Muchos a Muchos: Asociación entre dos tablas donde cada fila de una puede estar relacionada con varias filas de la otra.

Entidad-Relación (ER): Modelo gráfico que describe las entidades, sus atributos y relaciones en una base de datos.

Esquema Físico: Implementación técnica del esquema lógico en un sistema de gestión de bases de datos (DBMS).

Constraint (Restricción): Reglas aplicadas a columnas o tablas para garantizar la validez e integridad de los datos (ej. UNIQUE, CHECK).

Cascada (ON DELETE CASCADE): Comando que elimina automáticamente las filas relacionadas cuando se borra una fila referenciada.

Particionado de Tablas: Dividir una tabla grande en partes más pequeñas para mejorar el rendimiento y la administración.

Backup: Copia de seguridad de una base de datos para evitar la pérdida de datos en caso de fallos.

Replica: Copias exactas de una base de datos en diferentes ubicaciones para mejorar la disponibilidad y redundancia.

Data Integrity: Garantía de que los datos son precisos, consistentes y libres de errores.



Sistema de Gestión de Bases de Datos Relacionales (RDBMS): Software que facilita la creación, gestión y consulta de bases de datos relacionales (ej. MySQL, PostgreSQL, Oracle).

Optimización de Consultas: Proceso de mejorar el rendimiento de las consultas SQL mediante índices, estadísticas y reestructuración.



Unidad: Bases de Datos No Relacionales (NoSQL)

Base de Datos NoSQL: Tipo de base de datos que no utiliza un modelo relacional tradicional y está diseñado para manejar grandes volúmenes de datos no estructurados o semi-estructurados.

Escalabilidad Horizontal: Capacidad de añadir más servidores para manejar el aumento de datos y tráfico, característica clave de las bases de datos NoSQL.

Desnormalización: Estrategia utilizada en NoSQL que almacena datos redundantes para optimizar consultas y mejorar el rendimiento.

Modelo Clave-Valor: Base de datos donde los datos se almacenan como pares clave-valor, ideal para búsquedas rápidas (ej. Redis, DynamoDB).

Modelo Orientado a Documentos: Base de datos que almacena datos en formato JSON, BSON o XML, permitiendo estructuras flexibles (ej. MongoDB, CouchDB).

Modelo Columnar: Base de datos donde los datos se organizan y almacenan en columnas en lugar de filas, optimizado para consultas analíticas (ej. Cassandra, HBase).

Modelo de Grafos: Base de datos que utiliza nodos, aristas y propiedades para representar y almacenar relaciones entre datos (ej. Neo4j, ArangoDB).



JSON (JavaScript Object Notation): Formato ligero y legible utilizado para almacenar datos en bases de datos orientadas a documentos.

BSON (Binary JSON): Variante binaria de JSON utilizada en bases de datos NoSQL para mejorar el rendimiento.

Replica: Copia exacta de los datos en múltiples nodos para garantizar la disponibilidad y tolerancia a fallos.

Sharding: Técnica que divide los datos en fragmentos más pequeños que se distribuyen entre varios servidores para optimizar el rendimiento.

Consistencia Eventual: Modelo en el que las actualizaciones de datos se propagan a través de todos los nodos con el tiempo, garantizando disponibilidad en sistemas distribuidos.

CAP Theorem (Teorema CAP): Principio que establece que una base de datos distribuida no puede garantizar simultáneamente Consistencia, Disponibilidad y Tolerancia a Particiones.

Indexación: Técnica para acelerar la recuperación de datos en bases de datos NoSQL mediante estructuras adicionales.

Query: Consulta o instrucción para recuperar o manipular datos almacenados en una base de datos.

CQL (Cassandra Query Language): Lenguaje de consulta similar a SQL utilizado en bases de datos columnar como Cassandra.



API (Interfaz de Programación de Aplicaciones): Conjunto de funciones que permite a las aplicaciones interactuar con bases de datos NoSQL.

Base de Datos Distribuida: Sistema en el que los datos se almacenan en varios nodos o servidores interconectados.

Alta Disponibilidad: Capacidad de una base de datos para estar accesible en todo momento, incluso ante fallos del sistema.

Tolerancia a Fallos: Habilidad de un sistema para continuar operando incluso cuando uno o más componentes fallan.

TTL (Time to Live): Tiempo de vida de los datos antes de ser eliminados automáticamente, común en bases de datos clave-valor.

Agregación: Operación en bases de datos NoSQL para realizar cálculos o combinaciones de datos, similar a las funciones de agregación en SQL.

MapReduce: Técnica para procesar y generar grandes conjuntos de datos de manera distribuida, común en sistemas NoSQL como Hadoop.

Base de Datos Multimodelo: Sistema que soporta múltiples tipos de modelos de datos, como documentos, grafos y clave-valor (ej. ArangoDB).



Particionamiento: División lógica de los datos en bases de datos NoSQL para mejorar el rendimiento y la escalabilidad.

Carga en Lotes: Proceso de insertar o actualizar grandes cantidades de datos en una base de datos en una sola operación.

Latencia Baja: Característica que asegura tiempos de respuesta rápidos en operaciones de lectura y escritura.

Transacciones ACID: Aunque menos comunes en NoSQL, algunas bases soportan propiedades ACID para garantizar integridad en operaciones críticas (ej. MongoDB).

Consistency Levels (Niveles de Consistencia): Opciones para equilibrar consistencia y disponibilidad en bases de datos NoSQL (ej. fuerte, eventual).

NoSQL vs SQL: Comparación entre bases de datos no relacionales y relacionales, considerando escalabilidad, flexibilidad y modelos de datos.

MongoDB: Base de datos NoSQL orientada a documentos, popular por su facilidad de uso y flexibilidad.

Cassandra: Base de datos NoSQL columnar diseñada para manejar grandes volúmenes de datos distribuidos.

Neo4j: Base de datos orientada a grafos, ideal para analizar relaciones complejas entre datos.

Redis: Base de datos NoSQL clave-valor en memoria, optimizada para velocidad y almacenamiento temporal.



Eventual Consistency: Modelo donde las actualizaciones se propagan gradualmente a los nodos, logrando consistencia final.

Cluster: Conjunto de servidores interconectados que funcionan como una sola base de datos distribuida.

Inmutabilidad: Característica donde los datos una vez almacenados no se modifican, común en sistemas analíticos.

Elasticidad: Capacidad de una base de datos NoSQL para ajustar dinámicamente los recursos en función de la carga.

Base de Datos en Tiempo Real: Sistema diseñado para manejar y procesar datos en tiempo real, como en aplicaciones de mensajería o monitoreo.



DUODÉCIMO

Subárea Desarrollo de componentes de software

Unidad: Programación Orientada a Objetos (POO) avanzada

Componente de Software: Unidad modular, reutilizable e independiente que realiza una función específica en una aplicación.

Polimorfismo Dinámico: Capacidad de resolver llamadas a métodos en tiempo de ejecución, utilizando referencias a clases base para invocar métodos de clases derivadas.

Métodos Virtuales: Métodos que pueden ser sobrescritos en clases derivadas y que se resuelven en tiempo de ejecución.

Clases Internas: Clases definidas dentro de otra clase, utilizadas para organizar mejor el código o para encapsular comportamientos específicos.

Interfaces Funcionales: Interfaces con un solo método abstracto, comúnmente utilizadas en programación funcional y expresiones lambda.



Expresiones Lambda: Representaciones concisas de funciones anónimas, utilizadas para simplificar el código y manejar eventos o datos.

Anotaciones (Annotations): Metadatos utilizados para proporcionar información adicional sobre el código y facilitar procesos automáticos como inyección de dependencias o pruebas.

Decoradores: Patrón de diseño utilizado para agregar funcionalidades adicionales a clases o métodos sin modificar su estructura original.

Inyección de Dependencias: Técnica en la que las dependencias de una clase se proporcionan desde fuera, facilitando pruebas y flexibilidad.

Reflexión (Reflection): Capacidad de un programa para inspeccionar y modificar su propia estructura en tiempo de ejecución, incluyendo clases, métodos y atributos.

Serialización Avanzada: Conversión de objetos complejos en un formato transportable, con soporte para herencia, colecciones y datos anidados.

Deserialización Avanzada: Reconstrucción de objetos desde datos serializados, con soporte para validaciones y compatibilidad entre versiones.

Clases Abstractas Avanzadas: Clases que combinan métodos abstractos y concretos para definir plantillas de comportamiento más sofisticadas.



Interfaz Avanzada: Uso de múltiples interfaces para definir un comportamiento complejo en una clase, evitando problemas de herencia múltiple.

Mixin: Técnica que permite agregar funcionalidad a clases sin usar herencia múltiple, comúnmente mediante la implementación de interfaces o funciones adicionales.

Sobrecarga de Operadores: Modificación de los operadores estándar para que funcionen de manera personalizada con objetos de una clase (común en C++).

Métodos de Clase y de Instancia: Métodos que pueden ser llamados directamente desde la clase o desde un objeto de la clase, según su propósito y definición.

Clases Meta (Metaclasses): Clases que definen el comportamiento de otras clases, permitiendo modificar su creación y propiedades en tiempo de ejecución.

Clases Inmutables: Clases cuyos objetos no pueden ser modificados después de su creación, comúnmente utilizadas para manejar datos sensibles.

Programación Concurrente en POO: Uso de hilos, procesos o tareas asíncronas en objetos para manejar múltiples operaciones simultáneamente.

Sólidos Principios de Diseño (SOLID): Conjunto de principios para escribir código limpio y escalable en POO:

S: Principio de Responsabilidad Única (Single Responsibility Principle).



O: Principio de Abierto/Cerrado (Open/Closed Principle).

L: Principio de Sustitución de Liskov (Liskov Substitution Principle).

I: Principio de Segregación de Interfaces (Interface Segregation Principle).

D: Principio de Inversión de Dependencias (Dependency Inversion Principle).

Patrones de Diseño Avanzados: Soluciones arquitectónicas específicas para problemas complejos en POO:

Patrón Proxy: Control de acceso a objetos mediante intermediarios.

Patrón Builder: Construcción de objetos complejos mediante pasos configurables.

Patrón Decorator: Agregación de funcionalidades a objetos de manera dinámica.

Patrón Comando: Encapsulación de una solicitud como un objeto.

Arquitectura de Microservicios: División de aplicaciones en servicios pequeños e independientes que interactúan mediante POO y API.

ORM (Object-Relational Mapping): Herramientas que mapean clases y objetos a bases de datos relacionales para facilitar el manejo de datos (ej. Hibernate, SQLAlchemy).



TDD Avanzado (Test-Driven Development): Uso de pruebas automatizadas para definir y validar funcionalidades de objetos y clases antes de su desarrollo.

Patrón Singleton: Restricción que asegura que una clase tenga una única instancia global en toda la aplicación.

Clases Dinámicas: Creación de clases en tiempo de ejecución basadas en parámetros o configuraciones específicas.

Eventos y Delegados: Mecanismo para manejar la comunicación entre objetos mediante notificaciones y métodos delegados.

Acoplamiento y Cohesión: Medidas de la interdependencia entre clases (acoplamiento) y la relación lógica entre los elementos de una clase (cohesión).

Polimorfismo por Composición: Uso de la composición en lugar de la herencia para lograr flexibilidad en el comportamiento de los objetos.

Control de Versiones de Clases: Gestión de cambios en estructuras de clases para mantener compatibilidad con versiones anteriores.

Desacoplamiento: Diseño de clases que minimiza las dependencias para mejorar la reutilización y mantenibilidad.



POO en Entornos Distribuidos: Aplicación de principios de POO para manejar comunicación y objetos en sistemas distribuidos mediante RPC, gRPC o REST.

Programación Funcional en POO: Uso combinado de paradigmas funcional y orientado a objetos para optimizar el diseño de software.

Proxies Dinámicos: Generación de objetos proxy en tiempo de ejecución para controlar el acceso o modificar el comportamiento de otros objetos.

Estrategias de Pruebas: Métodos avanzados para evaluar componentes orientados a objetos:

Pruebas Unitarias: Validación de métodos individuales dentro de una clase.

Pruebas de Integración: Evaluación de la interacción entre múltiples componentes.

Pruebas de Mock: Simulación de objetos dependientes para aislar pruebas.

Unidad: Desarrollo de Aplicaciones Móviles

Aplicación Móvil: Software diseñado para ejecutarse en dispositivos móviles como smartphones y tablets.

Sistema Operativo Móvil: Software que gestiona los recursos y aplicaciones de un dispositivo móvil (ej. Android, iOS).



SDK (Software Development Kit): Conjunto de herramientas y bibliotecas que permiten el desarrollo de aplicaciones para una plataforma específica (ej. Android SDK, iOS SDK).

IDE (Integrated Development Environment): Entorno de desarrollo integrado que facilita la escritura, depuración y prueba de aplicaciones (ej. Android Studio, Xcode).

Aplicaciones Nativas: Apps desarrolladas específicamente para un sistema operativo móvil, utilizando herramientas y lenguajes propios de la plataforma (ej. Swift para iOS, Kotlin para Android).

Aplicaciones Híbridas: Apps que combinan tecnologías web (HTML, CSS, JavaScript) y se ejecutan en un contenedor nativo (ej. Ionic, Apache Cordova).

Aplicaciones Web Progresivas (PWA): Apps basadas en la web que ofrecen una experiencia similar a la de una app nativa sin necesidad de instalación.

UI/UX (Interfaz de Usuario y Experiencia de Usuario): Diseño visual y funcional de la aplicación, enfocado en la facilidad de uso y satisfacción del usuario.

Responsividad: Capacidad de una aplicación para ajustarse y funcionar adecuadamente en dispositivos con diferentes tamaños de pantalla y resoluciones.

API (Interfaz de Programación de Aplicaciones): Conjunto de funciones que permiten a las aplicaciones interactuar con servicios externos o componentes del sistema operativo.



Backend: Parte de la aplicación que maneja la lógica del negocio, el almacenamiento de datos y la comunicación con el servidor.

Frontend: Parte de la aplicación que interactúa directamente con el usuario, mostrando la interfaz gráfica.

Base de Datos Móvil: Sistema de almacenamiento de datos optimizado para funcionar en dispositivos móviles (ej. SQLite, Room, Core Data).

Firebase: Plataforma de Google que ofrece servicios backend como autenticación, base de datos en tiempo real, notificaciones push y análisis.

Autenticación Biométrica: Uso de características físicas como huellas digitales o reconocimiento facial para validar el acceso a la app.

Geolocalización: Tecnología que permite obtener la ubicación exacta de un dispositivo mediante GPS, Wi-Fi o redes móviles.

Notificaciones Push: Mensajes enviados directamente a los usuarios desde un servidor para mantenerlos informados o interactuar con ellos.

Permisos: Solicitudes que una aplicación realiza para acceder a funciones o datos específicos del dispositivo (ej. cámara, ubicación, contactos).



Gestión de Estados: Técnica para manejar el estado dinámico de una aplicación, asegurando que los datos y la interfaz se mantengan consistentes.

Optimización de Rendimiento: Proceso de mejorar la velocidad, consumo de recursos y tiempo de carga de una aplicación móvil.

Animaciones y Transiciones: Efectos visuales que mejoran la interacción del usuario con la aplicación.

Desarrollo Multiplataforma: Creación de aplicaciones que funcionan en múltiples sistemas operativos desde un único código base (ej. Flutter, React Native).

Arquitecturas Comunes: Patrones para organizar el código y facilitar el mantenimiento de aplicaciones:

MVC (Modelo-Vista-Controlador): Patrón que separa la lógica de la interfaz y el control de eventos.

MVVM (Modelo-Vista-VistaModelo): Patrón que facilita la vinculación de datos entre la lógica y la interfaz.

Clean Architecture: Arquitectura modular y escalable para aplicaciones móviles.

Emulador: Software que simula un dispositivo móvil en una computadora para probar y depurar aplicaciones.

Depuración (Debugging): Proceso de identificar y corregir errores en una aplicación móvil.



Testing: Evaluación de la funcionalidad, rendimiento y usabilidad de una aplicación móvil antes de su lanzamiento:

Pruebas Unitarias: Verificación de componentes individuales de la aplicación.

Pruebas de Interfaz: Validación de la interacción del usuario con la aplicación.

Pruebas de Integración: Comprobación de la interacción entre diferentes módulos de la aplicación.

Publicación: Proceso de subir una aplicación móvil a una tienda de aplicaciones como Google Play Store o Apple App Store.

Versionado: Identificación de las actualizaciones y cambios en una aplicación mediante números de versión.

Actualizaciones OTA (Over-The-Air): Distribución de nuevas versiones de la app sin necesidad de reinstalación manual.

Monetización: Estrategias para generar ingresos con una aplicación móvil:

Freemium: Ofrecer una versión gratuita con funciones premium de pago.

Publicidad: Integración de anuncios dentro de la app.

Compras In-App: Ventas de contenido o funcionalidades adicionales dentro de la aplicación.



Suscripciones: Cobros recurrentes por acceso a características exclusivas.

Analítica Móvil: Seguimiento del comportamiento del usuario dentro de la aplicación para optimizar la experiencia y las estrategias de marketing.

Seguridad Móvil: Prácticas para proteger los datos y la privacidad del usuario en aplicaciones móviles.

Tiempos de Carga: Tiempo que tarda una aplicación en estar lista para su uso después de abrirse.

Caché: Almacenamiento temporal de datos para mejorar el rendimiento y reducir el consumo de recursos.

Compatibilidad: Asegurar que la aplicación funcione correctamente en diferentes versiones de sistemas operativos y dispositivos.

Accesibilidad: Diseño de aplicaciones que sean fáciles de usar por personas con discapacidades.



Unidad: Comunicación entre Componentes de Software

Componente de Software: Módulo independiente que realiza una función específica y puede interactuar con otros componentes para formar una aplicación completa.

Interfaz: Conjunto de métodos o funciones expuestos por un componente que permite a otros interactuar con él.

Acoplamiento: Grado de dependencia entre componentes; se busca bajo acoplamiento para mejorar la modularidad y la reutilización.

Cohesión: Medida de qué tan enfocadas están las responsabilidades de un componente; se busca alta cohesión para mejorar la eficiencia y claridad.

API (Interfaz de Programación de Aplicaciones): Interfaz que define cómo interactúan los componentes de software entre sí.

REST (Representational State Transfer): Estilo arquitectónico para la comunicación entre sistemas mediante HTTP y datos como JSON o XML.

SOAP (Simple Object Access Protocol): Protocolo estándar para la comunicación entre aplicaciones utilizando XML.



RPC (Remote Procedure Call): Protocolo que permite a un componente ejecutar funciones en otro componente ubicado en un servidor remoto.

Webhooks: Mecanismo donde un sistema envía notificaciones automáticas a otros sistemas cuando ocurre un evento específico.

Mensajería: Modelo de comunicación asíncrona entre componentes mediante mensajes (ej. RabbitMQ, Kafka).

Colas de Mensajes: Estructura utilizada para gestionar y almacenar mensajes enviados entre componentes en sistemas distribuidos.

Protocolo de Comunicación: Conjunto de reglas que define cómo los datos se transmiten entre componentes (ej. HTTP, gRPC, AMQP).

gRPC (Google Remote Procedure Call): Protocolo de comunicación de alto rendimiento que utiliza HTTP/2 y serialización con Protobuf.

Microservicios: Arquitectura en la que una aplicación se divide en pequeños componentes independientes que se comunican entre sí.

Pub/Sub (Publicación/Suscripción): Modelo de comunicación en el que los componentes publican mensajes que son recibidos por suscriptores interesados.



Serialización: Proceso de convertir un objeto o datos en un formato que pueda transmitirse entre componentes, como JSON o Protobuf.

Deserialización: Reconstrucción de un objeto o datos a partir de su formato serializado.

Middleware: Software intermediario que facilita la comunicación y gestión de solicitudes entre componentes.

Eventos: Notificaciones generadas por un componente que otros componentes pueden escuchar y manejar.

Delegados: Referencias a métodos que permiten la invocación de funciones entre componentes.

Sockets: Interfaz que permite la comunicación en tiempo real entre componentes en una red mediante conexiones bidireccionales.

RPC Bidireccional: Comunicación remota que permite enviar y recibir mensajes simultáneamente entre componentes.

Carga Útil (Payload): Datos reales que se envían dentro de una solicitud o mensaje entre componentes.

Timeout: Límite de tiempo que un componente espera para recibir una respuesta antes de abortar la solicitud.

Balanceo de Carga: Distribución del tráfico entre múltiples instancias de componentes para optimizar el rendimiento.



Escalabilidad Horizontal: Adición de más instancias de un componente para manejar mayores cargas de trabajo.

Protocolos de Transporte: Métodos para transmitir datos entre componentes (ej. TCP, UDP).

Mensajería Asíncrona: Comunicación donde los componentes no necesitan esperar una respuesta inmediata (ej. colas de mensajes).

Mensajería Síncrona: Comunicación donde el emisor espera una respuesta inmediata del receptor.

Broker de Mensajes: Intermediario que gestiona la distribución de mensajes entre componentes (ej. RabbitMQ, ActiveMQ).

Interoperabilidad: Capacidad de diferentes componentes o sistemas para comunicarse y trabajar juntos eficazmente.

Integración Continua: Proceso de combinar y probar regularmente los componentes de software para garantizar que funcionen bien juntos.

Control de Concurrencia: Técnicas para manejar el acceso simultáneo a recursos compartidos por múltiples componentes.

Firmas Digitales: Método para asegurar la autenticidad e integridad de los mensajes transmitidos entre componentes.



Cifrado de Datos: Técnica para proteger la información transmitida entre componentes mediante métodos como SSL/TLS.

Tolerancia a Fallos: Capacidad de un sistema para continuar funcionando incluso cuando uno o más componentes fallan.

Circuit Breaker (Cortacircuitos): Patrón para manejar fallos en la comunicación entre componentes, evitando sobrecargar sistemas defectuosos.

Patrón de Mediador: Componente central que gestiona la comunicación entre otros componentes, reduciendo el acoplamiento directo.

Patrón de Observador: Diseño donde los componentes observadores reciben notificaciones de cambios en un componente observado.

Streaming de Datos: Comunicación continua de datos entre componentes en tiempo real (ej. WebSockets, Server-Sent Events).

Transformación de Datos: Modificación de los datos para adaptarlos al formato esperado por otro componente durante la comunicación.

Seguridad en la Comunicación: Prácticas para garantizar que los datos transmitidos entre componentes sean confidenciales y no manipulados.



Logs de Comunicación: Registro de las interacciones entre componentes para facilitar la depuración y el monitoreo.

APIs Versionadas: Práctica de gestionar múltiples versiones de una API para garantizar la compatibilidad entre componentes.



Subárea Experiencia de usuario y diseño de interacciones para componentes de software

Unidad: Experiencia de Usuario (UX)

Experiencia de Usuario (UX): Percepción y emociones de un usuario al interactuar con un producto, sistema o servicio.

Diseño Centrado en el Usuario: Proceso de diseño que coloca las necesidades, objetivos y comportamientos del usuario en el centro de la toma de decisiones.

Interfaz de Usuario (UI): Elementos visuales y funcionales con los que los usuarios interactúan directamente en un sistema.

Usabilidad: Facilidad con la que los usuarios pueden aprender, usar y alcanzar sus objetivos con un sistema.

Arquitectura de la Información: Organización y estructuración del contenido para facilitar la navegación y comprensión.

Wireframe: Representación básica y esquemática de la disposición de los elementos en una página o pantalla.

Prototipo: Modelo funcional de un diseño que permite probar interacciones y flujos antes de la implementación completa.



Mapa de Sitio: Diagrama que muestra la estructura y jerarquía de las páginas en un sitio web o aplicación.

Flujo de Usuario: Camino que sigue un usuario al interactuar con un producto para completar tareas específicas.

Persona: Representación ficticia de un usuario típico, basada en datos reales, que ayuda a guiar el diseño.

Pruebas de Usabilidad: Evaluaciones realizadas con usuarios reales para identificar problemas y mejorar la interacción.

Feedback del Usuario: Opiniones y sugerencias proporcionadas por los usuarios sobre su experiencia con un producto.

Diseño Responsivo: Adaptación automática de la interfaz para que funcione bien en diferentes tamaños de pantalla y dispositivos.

Accesibilidad: Diseño de productos digitales para que sean utilizables por personas con discapacidades.

Criterios de Diseño: Reglas y principios que guían las decisiones de diseño para cumplir con los objetivos del usuario y del negocio.

Card Sorting: Técnica para organizar y categorizar contenido basada en cómo los usuarios agrupan y etiquetan información.



Tarea Principal: Acción clave que un usuario debe realizar en un sistema para alcanzar su objetivo.

Microinteracciones: Pequeños detalles interactivos que mejoran la experiencia del usuario al dar retroalimentación inmediata (ej. animaciones al hacer clic).

Diseño de Interacción: Creación de flujos y comportamientos que definen cómo los usuarios interactúan con un sistema.

Jerarquía Visual: Organización de los elementos en una interfaz para guiar la atención del usuario hacia las áreas más importantes.

Teoría del Color: Uso estratégico de colores para transmitir emociones, destacar elementos y mejorar la experiencia visual.

Tipografía: Estilo y disposición del texto en una interfaz, clave para la legibilidad y la estética.

Gestos: Interacciones táctiles en dispositivos móviles, como deslizamientos, toques y pellizcos.

Onboarding: Proceso de guiar a los usuarios en su primera interacción con un producto para ayudarlos a comprender su funcionalidad.

Métricas UX: Indicadores clave para medir la experiencia de usuario, como el tiempo para completar una tarea, la tasa de error y la satisfacción del usuario.



Mapas de Calor: Representaciones visuales que muestran las áreas más interactuadas de una interfaz.

A/B Testing: Comparación de dos versiones de un diseño para determinar cuál ofrece una mejor experiencia al usuario.

Journey Map (Mapa del Viaje del Usuario): Representación visual del recorrido del usuario al interactuar con un producto, identificando puntos de fricción y oportunidades de mejora.

Empatía: Habilidad de comprender y diseñar para las emociones, necesidades y motivaciones del usuario.

Diseño Iterativo: Proceso de realizar pruebas, recopilar retroalimentación y mejorar continuamente el diseño.

Error 404: Mensaje que informa al usuario que una página o recurso solicitado no está disponible, cuyo diseño puede minimizar la frustración.

Análisis Heurístico: Evaluación de la usabilidad de un sistema basada en principios establecidos, como las heurísticas de Nielsen.

Sistema de Diseño: Conjunto de componentes reutilizables y guías que garantizan consistencia en el diseño de un producto.

Loading States: Pantallas o animaciones que informan al usuario que un sistema está procesando su solicitud.



Contexto de Uso: Circunstancias y entornos en los que los usuarios interactúan con un producto, influyendo en el diseño.

Retroalimentación Visual: Indicadores inmediatos, como cambios de color o mensajes emergentes, que confirman las acciones del usuario.

Diseño Emocional: Enfoque en cómo un diseño provoca emociones y crea una conexión positiva con los usuarios.

Patrones de Diseño: Soluciones probadas para problemas comunes en interfaces, como menús desplegables o barras de búsqueda.

Antropometría Digital: Consideración de las dimensiones físicas y limitaciones humanas al diseñar interfaces y dispositivos.

Fidelidad del Prototipo: Nivel de detalle y funcionalidad de un prototipo, desde baja fidelidad (esbozos) hasta alta fidelidad (interacciones completas).

Pruebas de Campo: Evaluación de la experiencia de usuario en un entorno real, fuera de un laboratorio controlado.

SEO (Optimización para Motores de Búsqueda): Diseño de sitios web que mejora la visibilidad en motores de búsqueda, relacionado con la experiencia de usuario.



Unidad: Diseño de Interacción (IxD)

Diseño de Interacción (IxD): Disciplina enfocada en la creación de sistemas interactivos que permiten a los usuarios realizar tareas de manera efectiva y agradable.

Interacción Usuario-Sistema: Proceso en el que los usuarios se comunican con un sistema para completar tareas específicas.

Flujo de Interacción: Secuencia de pasos que sigue un usuario al interactuar con un sistema para alcanzar un objetivo.

Microinteracciones: Pequeñas interacciones dentro de un sistema que proporcionan retroalimentación, como animaciones al pulsar un botón.

Elementos de Interfaz: Componentes visuales y funcionales que permiten la interacción, como botones, menús y formularios.

Modelo Mental: Percepción y expectativas del usuario sobre cómo debería funcionar un sistema.

Prototipo Interactivo: Representación funcional de un diseño que permite a los usuarios interactuar y experimentar flujos de trabajo.

Wireframe: Esquema inicial de una interfaz que muestra la disposición básica de los elementos, sin detalles visuales o de interacción.



Persona: Representación ficticia de un usuario típico que ayuda a guiar las decisiones de diseño de interacción.

Mapa de Navegación: Diagrama que muestra cómo las diferentes pantallas o páginas de un sistema están conectadas entre sí.

Animaciones: Efectos visuales que indican cambios de estado, transiciones o interacciones dentro de un sistema.

Retroalimentación: Respuesta inmediata del sistema a las acciones del usuario, como mensajes de confirmación o cambios visuales.

Estados del Sistema: Diferentes condiciones que puede presentar una interfaz según el contexto, como estados de carga, error o éxito.

Gestos: Acciones táctiles o basadas en movimientos que los usuarios realizan para interactuar con un sistema (ej. deslizar, tocar, pellizcar).

Consistencia: Mantenimiento de patrones y comportamientos similares en toda la interfaz para facilitar el aprendizaje del usuario.

Control del Usuario: Permitir que los usuarios tengan el poder de tomar decisiones y revertir acciones en el sistema.



Prevención de Errores: Diseño que minimiza las posibilidades de que los usuarios cometan errores, como validar datos en tiempo real.

Accesibilidad: Diseño inclusivo que asegura que las personas con discapacidades puedan interactuar con el sistema.

Jerarquía de Información: Organización de los elementos en la interfaz para guiar la atención del usuario hacia lo más importante.

Patrones de Diseño: Soluciones reutilizables para problemas comunes en el diseño de interacción (ej. menús desplegables, barras de búsqueda).

Metáforas de Interacción: Uso de conceptos familiares para los usuarios para representar acciones o funciones en el sistema (ej. un ícono de carpeta para archivos).

Antropometría Digital: Consideración de las dimensiones físicas y capacidades humanas al diseñar interacciones.

Test de Usabilidad: Evaluaciones con usuarios reales para identificar problemas en los flujos de interacción y proponer mejoras.

Loading States: Indicadores que muestran al usuario que el sistema está procesando una acción, como una barra de progreso o un spinner.



Fidelidad del Prototipo: Nivel de detalle y funcionalidad de un prototipo, desde baja fidelidad (esbozos básicos) hasta alta fidelidad (interacciones completas).

Diseño Adaptativo: Creación de interfaces que se ajustan a diferentes dispositivos y resoluciones para ofrecer una experiencia consistente.

Interacciones Asistidas: Uso de tecnologías como chatbots o asistentes virtuales para mejorar la interacción usuario-sistema.

Empatía: Habilidad de entender y diseñar desde la perspectiva del usuario, considerando sus emociones y necesidades.

Pruebas A/B: Comparación de dos versiones de una interfaz para determinar cuál ofrece mejor interacción y experiencia al usuario.

Flujo Multidispositivo: Diseño que asegura una experiencia coherente y continua cuando un usuario cambia entre diferentes dispositivos.

Zonas Táctiles: Áreas de la interfaz optimizadas para la interacción mediante toques, considerando el tamaño y alcance de los dedos.

Diseño Basado en Tareas: Enfoque de diseño que prioriza las tareas principales que los usuarios necesitan completar.



Interfaces Conversacionales: Sistemas que permiten a los usuarios interactuar mediante texto o voz, como asistentes virtuales y chatbots.

Principio de Feedback Inmediato: Regla que asegura que cada acción del usuario reciba una respuesta inmediata del sistema.

Diseño Progresivo: Mostrar únicamente los elementos necesarios en cada momento, revelando opciones avanzadas o adicionales según sea necesario.

Cognitive Load (Carga Cognitiva): Esfuerzo mental requerido para interactuar con un sistema; el diseño debe minimizar esta carga.

Primero la Móvil (Mobile-First): Estrategia que prioriza el diseño para dispositivos móviles antes de adaptarlo a pantallas más grandes.

Indicadores de Navegación: Elementos visuales como migas de pan o barras de progreso que ayudan al usuario a entender dónde están y cómo avanzar.

Heurísticas de Nielsen: Principios básicos para evaluar y mejorar la usabilidad en el diseño de interacción.

Interacciones en Tiempo Real: Actualizaciones y respuestas instantáneas en la interfaz, comunes en sistemas de chat o monitoreo.

Simplicidad: Diseño que elimina elementos innecesarios para facilitar la comprensión y uso de la interfaz.



**MINISTERIO DE
EDUCACIÓN PÚBLICA**

**GOBIERNO
DE COSTA RICA**

**Dirección de Educación
Técnica y Capacidades
Emprendedoras**

Diseño Emocional: Incorporar elementos que despierten emociones positivas en los usuarios al interactuar con el sistema.



»»» Desarrollo de Aplicaciones de Software



Nivel
Undécimo

Juega para aprender



“Motívate a la salud digital en tu aprendizaje”