



MINISTERIO DE
EDUCACIÓN PÚBLICA

GOBIERNO
DE COSTA RICA

Dirección de Educación
Técnica y Capacidades
Emprendedoras



»»» Desarrollo de
Aplicaciones
de Software

Nivel
Duodécimo

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 Desarrollo de componentes de software	120
Descripción de la Subárea Desarrollo de componentes de software	121
Subárea Experiencia de usuario diseño de interacciones para componentes de software.....	137
Descripción de la Subárea Experiencia de usuario diseño de interacciones para componentes de software	138
Subject Area English Oriented to Software Applications Development XII	149
Description.....	150
Rationale.....	154
Education for Sustainable Development	157
Global Citizenship with National Identity	157
Digital Citizenship with Social Equity	158
Common European Framework of Reference for Languages	160
General Mediation Strategies and Pedagogical Approach	162
The Methodology Used in the Classroom.....	171



Curricular Design Template Elements	174
Curriculum Template.....	177
Planning	179
Annual Learning Plan	179
Pedagogical Practice Plan.....	181
Task-Building Process.....	182
Pedagogical Practice Plan.....	188
Evaluation of the Learning Process	192
Curricular Grid: English-Oriented to Software Applications Development	197
Curricular Design	206
Curricular Design	224
Curricular Design	242
Curricular Design	263
References.....	289
Referencias Bibliográficas.....	294
Referencias Generales	294
Bibliografía complementaria	300



MINISTERIO DE
EDUCACIÓN PÚBLICA

GOBIERNO
DE COSTA RICA

Dirección de Educación
Técnica y Capacidades
Emprendedoras

<i>Apéndices</i>	303
<i>Glosario de Términos</i>	304



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.

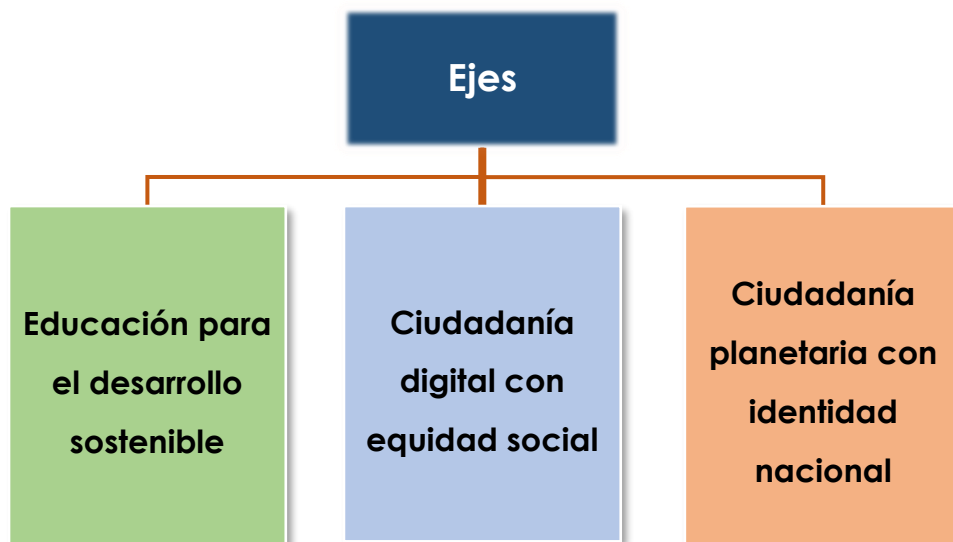
Este acompañamiento, por parte de la persona docente, es trascendental en el proceso educativo de una carrera técnica, ya que, durante la mediación pedagógica y la ejecución de visitas técnicas, giras, pasantías y prácticas profesionales en la empresa, las personas estudiantes pueden utilizar equipos, 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 ETEP 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-ETEP-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 aprehender 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												
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 Desarrollo de componentes de software



Subárea: Desarrollo de componentes de software



Desarrollo de
Aplicaciones
de Software



Nivel

Duodécimo



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 en duodécimo nivel se enfoca en el diseño, desarrollo e integración de soluciones de software complejas, robustas y escalables, que respondan a necesidades reales de usuarios y organizaciones en el contexto de la transformación digital.

Esta subárea cuenta con 300 horas anuales y contempla el desarrollo de tres unidades de estudio: Programación orientada a objetos avanzada, Aplicaciones móviles (APPS) y Comunicación entre componentes de software. Estas unidades de estudio fortalecen en la persona estudiante el pensamiento lógico, la capacidad de análisis, la resolución de problemas técnicos, el liderazgo en el desarrollo de soluciones colaborativas y la responsabilidad social del software.

La unidad de estudio de Programación orientada a objetos avanzada implementa estructuras complejas en entornos asincrónicos, concurrentes y distribuidos, integrando arquitecturas modernas y arquitectura hexagonal. Se profundiza en el uso de patrones de diseño avanzados, pruebas automatizadas y herramientas de calidad, incorporando además principios éticos en el uso de inteligencia artificial y objetos inteligentes, para soluciones sostenibles e inclusivas.

La unidad de Aplicaciones móviles (APPS) permite analizar necesidades reales de usuarios que facilite el desarrollo de soluciones móviles accesibles, funcionales y seguras. Además, se promueve el diseño de interfaces intuitivas y éticas, el uso eficiente de recursos, la publicación responsable de aplicaciones y el



proceso de mejora continua, aplicando criterios de diseño centrado en el ser humano y estándares de accesibilidad y usabilidad.

En la unidad de estudio Comunicación entre componentes de software, se abordan los principios y mecanismos de interacción entre módulos y sistemas heterogéneos, empleando contratos de interfaz, protocolos de comunicación síncronos y asíncronos, así como pruebas de integración y monitoreo. Se promueve la interoperabilidad, seguridad y eficiencia en sistemas distribuidos, fomentando la innovación técnica y la trazabilidad de soluciones.

Esta subárea permite al estudiantado desarrollarse como profesionales técnicos capaces de crear soluciones tecnológicas integradas, de calidad y con visión ética, aportando valor al entorno social y productivo, mediante el desarrollo de software responsable, colaborativo e innovador.



Tabla 17

Distribución de unidades de estudio de la subárea Desarrollo de componentes de software

Unidades de estudio	N° semanas	N° horas anuales
1 Programación orientada a objetos avanzada	9	108
2 Aplicaciones móviles	10	120
3 Comunicación entre componentes de software	6	72
Total	25	300



Tabla 21

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: Duodécimo
Unidad de estudio: Programación orientada a objetos (POO) avanzada	Tiempo estimado: 108 horas
Competencias para el desarrollo humano: Empatía	Eje política educativa: Educación para el desarrollo sostenible

Tabla 22

Planificación curricular de la unidad de estudio

Resultados de aprendizaje	Saberes esenciales	Indicador de logro
1. Aplicar estructuras avanzadas de programación orientada a objetos en entornos concurrentes, asíncronos y distribuidos.	<ul style="list-style-type: none">• Hilos y concurrencia• POO y asincronía• Programación orientada a eventos	<ul style="list-style-type: none">• Desarrolla aplicaciones de POO concurrentes y distribuidas.• Aplica los principios avanzados de la POO.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Comunicación entre objetos distribuidos• Gestión de estados compartidos y sincronización• Condiciones de carrera	<ul style="list-style-type: none">• Utiliza técnicas que mitiguen problemas como las condiciones de carrera.
2. Diseñar sistemas orientados a objetos aplicando arquitecturas modernas y principios de diseño evolutivo.	<ul style="list-style-type: none">• Arquitectura hexagonal• Entidades• Agregados• Servicios• Principios de separación de responsabilidades• Diseño guiado por testeo• Inversión de dependencias avanzadas	<ul style="list-style-type: none">• Modela sistemas orientados a objetos con arquitectura hexagonal.• Aplica procedimientos del diseño guiado por el dominio.• Refactoriza arquitecturas para desacoplar responsabilidades.
3. Implementar patrones de diseño avanzados en entornos de alta variabilidad, aplicando principios de extensión flexible y bajo acoplamiento para el diseño de componentes adaptativos.	<ul style="list-style-type: none">• Patrones avanzados• Patrón de especificación• Aplicación contextual• Antipatrones y cómo evitarlos• Aplicaciones de software adaptativas	<ul style="list-style-type: none">• Utiliza patrones avanzados en el desarrollo de aplicaciones de software• Distingue entre patrones útiles y antipatrones.• Desarrolla soluciones adaptativas usando la Programación Orientada a Objetos.
4. Integrar herramientas de calidad y pruebas automatizadas orientadas a objetos en el ciclo de vida del software que garanticen	<ul style="list-style-type: none">• Pruebas unitarias• Pruebas de integración• Dobles de pruebas• Análisis estático de código	<ul style="list-style-type: none">• Diseña pruebas automatizadas basadas en clases y objetos.• Utiliza herramientas de análisis estático.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
robustez, legibilidad y mantenibilidad del código.	<ul style="list-style-type: none">• Métricas de calidad para diseño<ul style="list-style-type: none">• Cohesión• Acoplamiento• Complejidad	<ul style="list-style-type: none">• Evalúa la calidad de las soluciones con métricas objetivas.
5. Desarrollar soluciones orientadas a objetos que integren el uso de inteligencia artificial (IA) y objetos inteligentes.	<ul style="list-style-type: none">• Agentes orientados a objetos• Objetos inteligentes y autónomos• Ética en la toma de decisiones automatizadas• Riesgos y sesgos en sistemas inteligentes• Derechos digitales• Privacidad desde el diseño	<ul style="list-style-type: none">• Modela objetos inteligentes aplicando principios éticos.• Identifica los riesgos éticos al desarrollar soluciones de aplicaciones de software con IA.• Diseña aplicaciones de software respetando la privacidad y el consentimiento informado.
6. Emplear alternativas para el desarrollo de la empatía que promuevan la sana convivencia.	Empatía <ul style="list-style-type: none">• Concepto• Características• Importancia• Alternativas para desarrollarla<ul style="list-style-type: none">• Mejorar la empatía pensando en el otro• Dejar de creer que somos el centro del universo• Ir más allá de las palabras de la otra persona• Decir adiós a los prejuicios	<ul style="list-style-type: none">• Reconoce el concepto e importancia de la empatía.• Identifica las características y formas de desarrollar la empatía.• Aplica alternativas para el desarrollo de la empatía.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Tener cuidado sobre cómo nos comunicamos• Evitar minimizar los problemas ajenos	
7. 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.	<p>Ahorro energético.</p> <ul style="list-style-type: none">• Concepto y características• Tipos: solar, eólica, biomasa, biogás, geotérmica.• Energías limpias.• Eficiencia energética.	<ul style="list-style-type: none">• Reconoce formas de ahorro energético de manera individual y colectiva.• Explica tipos y formas de uso de energías limpias en su entorno de trabajo.• Emplea eficiencia energética en la ejecución de trabajos eléctricos.



Tabla 23

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: Duodécimo
Unidad de estudio: Aplicaciones móviles (APPS)	Tiempo estimado: 120 horas
Competencias para el desarrollo humano: Innovación y creatividad	Eje política educativa: Educación para el desarrollo sostenible

Tabla 24

Planificación curricular de la unidad de estudio

Resultados de aprendizaje	Saberes esenciales	Indicador de logro
1. Analizar requerimientos funcionales y no funcionales para el desarrollo de aplicaciones móviles, considerando necesidades	<ul style="list-style-type: none">• Recolección de requerimientos• Necesidades del usuario móvil• Funcionalidades comunes en APPS móviles• Diseño centrado en el usuario UX/UI	<ul style="list-style-type: none">• Reconoce la importancia de la recolección de requerimientos claros y relevantes para el desarrollo de software.• Distingue las necesidades del usuario móvil.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
reales del usuario y principios de diseño.	<ul style="list-style-type: none">• Accesibilidad y usabilidad• Responsabilidad en el análisis de necesidades	<ul style="list-style-type: none">• Realiza la toma de decisiones de diseño considerando la experiencia del usuario.• Identifica las características de accesibilidad y usabilidad en el desarrollo de aplicaciones móviles.• Argumenta la importancia del análisis de necesidades en el desarrollo de aplicaciones.
2. Diseñar interfaces de usuario móviles, funcionales y accesibles, considerando los principios del diseño visual.	<ul style="list-style-type: none">• Principios de diseño visual y responsive• Accesibilidad en plataformas móviles• Navegación móvil intuitiva• Responsabilidad en el uso de datos sensibles• Diseño inclusivo	<ul style="list-style-type: none">• Crea interfaces accesibles, estéticas e intuitivas.• Aplica los principios de la ética e inclusividad, en el diseño de interfaces de usuario.• Demuestra responsabilidad en el tratamiento visual de la información.
3. Desarrollar aplicaciones móviles funcionales, seguras y eficientes, integrando componentes comunes.	<ul style="list-style-type: none">• Ciclo de vida de las aplicaciones móviles<ul style="list-style-type: none">• Etapas• Componentes comunes<ul style="list-style-type: none">• Pantallas• Formularios• Menús• Validación de datos• Conectividad	<ul style="list-style-type: none">• Identifica las etapas del ciclo de vida de las aplicaciones móviles.• Diferencia los componentes comunes de las aplicaciones móviles.• Desarrolla aplicaciones aplicando los criterios de



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Almacenamiento• Principios de responsabilidad en el desarrollo seguro	<ul style="list-style-type: none">funcionalidad, eficiencia y seguridad.• Implementa soluciones seguras para la protección de la información del usuario.• Aplica las etapas del ciclo de vida del desarrollo de aplicaciones móviles.
4. Aplicar pruebas funcionales y de usabilidad que optimicen el desarrollo de aplicaciones móviles, garantizando su rendimiento, seguridad y usabilidad.	<ul style="list-style-type: none">• Pruebas funcionales• Pruebas funcionales y de usabilidad• Estrategias de mejora del rendimiento móvil• Evaluación de seguridad• Buenas prácticas de depuración• Responsabilidad en la entrega de productos de calidad	<ul style="list-style-type: none">• Implementa pruebas funcionales y de usabilidad.• Optimiza la experiencia de uso aplicando criterios de eficiencia.• Ejecuta acciones para la evaluación de la seguridad de las aplicaciones.• Aplica buenas prácticas de depuración que garanticen la entrega de productos de calidad.
5. Publicar aplicaciones móviles aplicando la normativa legal y ética vigente; y considerando la retroalimentación de los usuarios para la mejora continua.	<ul style="list-style-type: none">• Procesos de publicación en tiendas de aplicaciones• Políticas de privacidad• Normativas legales sobre apps móviles• Gestión de actualizaciones y mejoras	<ul style="list-style-type: none">• Reconoce los procesos de publicación en las tiendas de aplicaciones.• Aplica políticas de privacidad y la normativa ética y legal vigente para la publicación de aplicaciones móviles.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Atención a comentarios de usuarios• Ética y responsabilidad en la distribución	<ul style="list-style-type: none">• Gestiona actualizaciones, el mantenimiento y mejoras del software.• Integra aspectos que surgen de la retroalimentación del usuario
6. Proponer soluciones creativas e innovadoras a necesidades y problemas cotidianos del contexto.	Innovación y creatividad: <ul style="list-style-type: none">• Concepto.• Precondición de la creatividad.• Métodos y técnicas de creatividad.• Creatividad en proceso de pensamiento.• Fases de la resolución creativa de problemas.• Lugares en donde se generan las ideas creativas.• ¿Qué influye en la creatividad?•	<ul style="list-style-type: none">• Reconoce el concepto de innovación y creatividad.• Diferencia las formas y fases para la resolución de problemas con creatividad e innovación.• Formula soluciones de manera creativa e innovadora a las necesidades o problemas que surgen del desarrollo de aplicaciones móviles.
7. Examinar el concepto, características e importancia de la sostenibilidad en el desarrollo de aplicaciones móviles.	<ul style="list-style-type: none">• Sostenibilidad<ul style="list-style-type: none">• Concepto.• Características.• Importancia.• Proyectos de desarrollo de aplicaciones sostenibles:<ul style="list-style-type: none">• Beneficios.	<ul style="list-style-type: none">• Explica el concepto de sostenibilidad y su importancia.• Distingue las características de los proyectos de desarrollo de aplicaciones móviles sostenibles.• Identifica necesidades o problemas que pueden



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">Necesidades o problemas por resolver	solucionarse con prácticas que se ejecutan con proyectos de desarrollo de aplicaciones móviles sostenibles.



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: Duodécimo
Unidad de estudio: Comunicación entre componentes de software	Tiempo estimado: 72 horas
Competencias para el desarrollo humano: Comunicación asertiva	Eje política educativa: La ciudadanía planetaria con identidad nacional

Tabla 26

Planificación curricular de la unidad de estudio

Resultados de aprendizaje	Saberes esenciales	Indicador de logro
1. Analizar los tipos de componentes de software y sus responsabilidades dentro de los sistemas distribuidos o modulares.	<ul style="list-style-type: none">• Tipos de componentes• Separación de responsabilidades• Cohesión• Acoplamiento	<ul style="list-style-type: none">• Distingue los tipos de componentes de software.• Clasifica los componentes de software, según su funcionalidad.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Escenarios de modularidad• Ética en los roles de funciones	<ul style="list-style-type: none">• Reconoce la importancia de la cohesión y el bajo acoplamiento.• Realiza decisiones de diseño en la arquitectura del software.
2. Aplicar mecanismos de comunicación síncronos y asíncronos entre componentes, evaluando su idoneidad, según los requerimientos funcionales y no funcionales del sistema.	<ul style="list-style-type: none">• Comunicación síncrona• Comunicación asíncrona• Protocolos comunes• Ventajas• Desafíos• Tiempos de respuesta• Tolerancia a fallos	<ul style="list-style-type: none">• Diferencia la comunicación síncrona y asíncrona.• Evalúa ventajas y desafíos de la comunicación síncrona y asíncrona.• Diseña soluciones efectivas con base en requerimientos técnicos y operativos.
3. Diseñar interfaces de comunicación entre componentes, utilizando contratos definidos que garanticen interoperabilidad, mantenibilidad y escalabilidad de aplicaciones de software.	<ul style="list-style-type: none">• Interfaces• Contratos• Servicios• Especificaciones• Versionamiento• Buenas prácticas de diseño de interfaces	<ul style="list-style-type: none">• Reconoce contratos entre componentes.• Utiliza especificaciones que garanticen la interoperabilidad.• Aplica principios de diseño que favorezcan el mantenimiento y escalamiento de aplicaciones de software.
4. Implementar mecanismos de integración entre componentes de software heterogéneos.	<ul style="list-style-type: none">• Integración de componentes con tecnologías diversas• Serialización• Formatos de datos• Control de acceso• Seguridad en la comunicación• Flujo de datos	<ul style="list-style-type: none">• Integra componentes heterogéneos de forma funcional y segura.• Gestiona el flujo de información entre servicios de forma eficiente.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">Transformación	<ul style="list-style-type: none">Aplica mecanismos de seguridad en la transmisión de datos.
5. Validar la comunicación entre componentes mediante pruebas de integración y monitoreo.	<ul style="list-style-type: none">Pruebas de integraciónSimulación de componentesHerramientas de monitoreoHerramientas de trazabilidadGestión de erroresFallos de comunicaciónMétodos innovadores y creativos para la comunicación de componentes de software	<ul style="list-style-type: none">Ejecuta pruebas efectivas para la validación de la comunicación entre módulos.Establece mecanismos de monitoreo y diagnóstico.Aplica métodos para la comunicación de los componentes integrados.
6. Emplear formas de comunicación asertiva en la convivencia con las personas.	Comunicación asertiva: <ul style="list-style-type: none">Concepto.Obstáculos para ser una persona asertiva:<ul style="list-style-type: none">Agresivo y pasivo.Técnicas para la comunicación asertiva.	<ul style="list-style-type: none">Reconoce el concepto de comunicación asertiva.Compara rasgos humanos de la persona asertiva, pasiva y agresiva.Aplica técnicas de comunicación asertiva en contextos de su área de formación técnica.
7. Argumentar sobre cómo la discriminación genera desigualdad.	<ul style="list-style-type: none">Discriminación<ul style="list-style-type: none">¿Qué es?Tipos de discriminaciónEjemplos de cómo se produce	<ul style="list-style-type: none">Reconoce el concepto de discriminación.Identifica los tipos de discriminación.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Acciones para evitar la discriminación	<ul style="list-style-type: none">• Argumenta acerca de cómo se produce la discriminación y qué acciones se deben realizar para evitarla.




MINISTERIO DE
EDUCACIÓN PÚBLICA

GOBIERNO
DE COSTA RICA

Dirección de Educación
Técnica y Capacidades
Emprendedoras

Subárea Experiencia de usuario diseño de interacciones para componentes de software



 **Subárea:**
**Experiencias de
usuario y diseño de
interacciones para
componentes de
software**

»»» **Desarrollo de
Aplicaciones
de Software**



Nivel

Duodécimo



Juega para aprender



"Motívate a la salud digital en tu aprendizaje"





Descripción de la Subárea Experiencia de usuario diseño de interacciones para componentes de software

La Subárea de Experiencia de Usuarios y Diseño de Interacciones para Componentes tiene como objetivo desarrollar en el estudiantado una visión integral centrada en las personas, que les permita diseñar productos digitales accesibles, funcionales, intuitivos y emocionalmente significativos. Esta subárea combina el pensamiento crítico, la empatía y la creatividad con habilidades técnicas para la investigación, el diseño iterativo, la evaluación de prototipos y la comunicación efectiva.

La subárea tiene una duración de 200 horas anuales de formación y contempla el desarrollo de dos unidades de estudio: Experiencia de Usuario (UX) y Diseño de Interacciones (IxD), que en conjunto permiten abordar el diseño centrado en las personas desde la comprensión de sus necesidades, hasta la validación de soluciones digitales que generan valor social, emocional y funcional.

En la unidad de estudio denominada Experiencia de Usuario (UX), el estudiantado desarrollará capacidades para el análisis de los fundamentos y principios de la experiencia de usuario, interpretando el comportamiento, expectativas y emociones de las personas usuarias mediante métodos de observación y recolección de datos. Se promueve la creación de soluciones innovadoras a partir del prototipado y diseño iterativo, así como la evaluación constante de las experiencias digitales, buscando siempre la mejora continua y la sostenibilidad. Además, se desarrollan habilidades comunicativas y visuales para presentar propuestas UX a distintos públicos, incluyendo usuarios finales, desarrolladores y clientes.



La unidad de estudio Diseño de Interacciones (IxD) profundiza en la creación de experiencias digitales coherentes y centradas en el ser humano, aplicando principios de usabilidad, accesibilidad e inclusión. El estudiantado diseña y prototipa interacciones que promuevan la equidad social y representen la diversidad cultural, trabajando de manera colaborativa en equipos multidisciplinarios. Se fomenta el análisis crítico de las interacciones propuestas, evaluando su efectividad y valor social, mediante el uso de herramientas de diseño y técnicas de prueba con usuarios.

Esta subárea promueve la formación de personas técnicas creativas, empáticas y comprometidas con el desarrollo de productos digitales que respondan a necesidades reales, fomentando la inclusión, la participación activa y el bienestar de todas las personas usuarias en entornos digitales diversos.



Tabla 18

Distribución de unidades de estudio de la subárea Experiencia de usuario diseño de interacciones para componentes de software

Unidades de estudio	N° semanas	N° horas anuales
① Experiencia de usuario (UX)	13	104
② Diseño de interacciones (IxD)	12	96
Total	25	200



Tabla 28

Información administrativa

Carrera técnica: Desarrollo de Aplicaciones de Software	Campo detallado: 0619 Tecnologías de la información y la comunicación (TIC)
Subárea: Experiencia de usuario y diseño de interacciones para componentes de software	Nivel: Duodécimo
Unidad de estudio: Experiencias de usuario (UX)	Tiempo estimado: 104 horas
Competencias para el desarrollo humano: Capacidad de negociación	Eje política educativa: La ciudadanía digital con equidad social

Tabla 29

Planificación curricular de la unidad de estudio

Resultados de aprendizaje	Saberes esenciales	Indicador de logro
1. Determinar los fundamentos de la experiencia de usuario (UX) y su impacto en el desarrollo de productos digitales funcionales y accesibles.	<ul style="list-style-type: none">• Historia de UX• Evolución UX• Principios del diseño centrado en el usuario• Accesibilidad	<ul style="list-style-type: none">• Identifica los principios de la experiencia de usuario y su aplicabilidad.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
	<ul style="list-style-type: none">• Usabilidad• Diseño emocional• Psicología del usuario digital	<ul style="list-style-type: none">• Explica el impacto del diseño en la experiencia emocional del usuario.• Aplica criterios de accesibilidad y usabilidad en el desarrollo de productos digitales.• Argumenta aspectos de la psicología del usuario digital.
2. Interpretar las necesidades, comportamientos y expectativas de los usuarios reales, mediante métodos innovadores de observación y recolección de datos.	<ul style="list-style-type: none">• Métodos de investigación UX<ul style="list-style-type: none">• Encuestas• Entrevistas• Pruebas de usabilidad• Diseño participativo• Técnicas creativas• Empatía e interpretación de emociones de usuarios• Diseño responsable con el entorno del usuario	<ul style="list-style-type: none">• Reconoce métodos de investigación de la experiencia del usuario (UX)• Aplica técnicas innovadoras de investigación centrada en el usuario.• Analiza patrones de comportamiento relevantes.
3. Diseñar soluciones UX innovadoras, utilizando herramientas de ideación, prototipado y diseño iterativo en la resolución de problemas.	<ul style="list-style-type: none">• Técnicas de ideación<ul style="list-style-type: none">• Design Thinking u otros• Prototipado de baja• Prototipado de alta fidelidad• Iteración• Validación• Creatividad aplicada al diseño de experiencias• Innovación centrada en el usuario	<ul style="list-style-type: none">• Aplica técnicas de ideación.• Genera propuestas creativas con base en necesidades reales.• Crea prototipos funcionales para validar ideas.• Aplica procesos iterativos para la solución de aplicaciones de software.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
4. Verificar la experiencia de usuario, mediante la implementación de pruebas controladas e instrumentos de retroalimentación.	<ul style="list-style-type: none">• Pruebas de usabilidad• Métricas<ul style="list-style-type: none">• Tiempo de tareas• Errores• Otras• Feedback continuo• Análisis de interacción• Mejora continua del diseño• Diseño sostenible	<ul style="list-style-type: none">• Realiza el diseño e implementación de pruebas de usabilidad efectivas.• Interpreta resultados y propone mejoras innovadoras.• Aplica principios de sostenibilidad en el desarrollo de aplicaciones de software.
5. Comunicar propuestas de experiencia de usuario (UX) de manera creativa y visual, adaptando el mensaje a diversos públicos.	<ul style="list-style-type: none">• Storytelling en diseño UX• Mapas de experiencia• Presentación visual de hallazgos• Comunicación empática con equipos multidisciplinarios• Ética en la representación del usuario	<ul style="list-style-type: none">• Presenta resultados de UX.• Comunica UX a diferentes públicos.• Propone soluciones considerando información brindada por el usuario final.
6. Aplicar estrategias de negociación que propicien acuerdos exitosos.	<p>Capacidad de negociación:</p> <ul style="list-style-type: none">• Concepto.• Elementos del proceso de una negociación exitosa.• Habilidades para la negociación.• Estrategias para la negociación	<ul style="list-style-type: none">• Reconoce el concepto de capacidad de negociación.• Explica las habilidades de las personas negociadoras.• Determina los elementos de la negociación exitosa.• Negocia la ejecución de propuestas de acuerdos viables en el contexto de su área de formación técnica.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
7. Argumentar sobre cómo la discriminación genera desigualdad.	<ul style="list-style-type: none">• Discriminación<ul style="list-style-type: none">• ¿Qué es?• Tipos de discriminación• Ejemplos de cómo se produce• Acciones para evitar la discriminación	<ul style="list-style-type: none">• Reconoce el concepto de discriminación.• Identifica los tipos de discriminación.• Argumenta acerca de cómo se produce la discriminación y qué acciones se deben realizar para evitarla.



Tabla 30

Información administrativa

Carrera técnica: Desarrollo de Aplicaciones de Software	Campo detallado: 0619 Tecnologías de la información y la comunicación (TIC)
Subárea: Experiencia de usuario y diseño de interacciones para componentes de software	Nivel: Duodécimo
Unidad de estudio: Diseño de Interacciones (IxD)	Tiempo estimado: 96 horas
Competencias para el desarrollo humano: Discernimiento y responsabilidad	Eje política educativa: Educación para el desarrollo sostenible

Tabla 31

Planificación curricular de la unidad de estudio

Resultados de aprendizaje	Saberes esenciales	Indicador de logro
1. Aplicar los principios fundamentales del diseño de interacciones para la creación de experiencias digitales,	<ul style="list-style-type: none">Fundamentos del diseño de interacciones (IxD)Elementos clave<ul style="list-style-type: none">AccionesRetroalimentación	<ul style="list-style-type: none">Explica los fundamentos del diseño de interacciones.Aplica elementos clave y principios del diseño de interacciones.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
coherentes, intuitivas y centradas en las personas.	<ul style="list-style-type: none">• Control• Consistencia• Principios de diseño centrado en el usuario• Diferencias entre IxD, UX y UI• Experiencias digitales accesibles e inclusivas.	<ul style="list-style-type: none">• Diferencia entre IxD, UX y UI.
2. Diseñar interacciones digitales que promuevan la equidad social, la accesibilidad y la participación activa de distintos perfiles de usuarios.	<ul style="list-style-type: none">• Diseño con accesibilidad universal e inclusiva• Brecha digital en el acceso a la información• Interacción para la ciudadanía digital• Diseño ético y responsable• Casos de estudio de soluciones de aplicaciones de software	<ul style="list-style-type: none">• Diseña propuestas que promueven la equidad social.• Evalúa interfaces desde la perspectiva de inclusión.• Aplica principios de ciudadanía digital en las decisiones de diseño.
3. Elaborar el diseño y desarrollo de prototipos interactivos aplicando herramientas colaborativas.	<ul style="list-style-type: none">• Roles en equipos de diseño• Herramientas colaborativas digitales• Comunicación efectiva en equipos• Resolución de conflictos• Ética en el trabajo colaborativo	<ul style="list-style-type: none">• Reconoce el rol en equipos de diseño.• Distingue herramientas colaborativas digitales utilizadas en el diseño y desarrollo de prototipos.• Utiliza herramientas colaborativas digitales en el diseño y desarrollo de prototipos.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
4. Desarrollar prototipos funcionales de interacciones digitales utilizando herramientas de diseño.	<ul style="list-style-type: none">• Prototipado interactivo• Diseño visual enfocado según al contexto• Simulación de experiencias interactivas• Narrativa visual y feedback	<ul style="list-style-type: none">• Desarrolla prototipos funcionales de calidad.• Simula experiencias interactivas.• Aplica principios de la narrativa visual y el feedback.
5. Verificar las interacciones digitales propuestas, considerando criterios de usabilidad, accesibilidad, satisfacción del usuario y valor social del producto digital.	<ul style="list-style-type: none">• Pruebas de interacción• Heurística de usabilidad• Métodos de recolección de retroalimentación• Indicadores de impacto social• Mejora continua con enfoque acorde a contexto	<ul style="list-style-type: none">• Realiza evaluaciones basadas en datos reales de uso.• Identifica oportunidades de mejora con base en la experiencia del usuario.• Demuestra responsabilidad y compromiso con el impacto social de las soluciones.
6. Aplicar los principios de discernimiento y responsabilidad en el manejo y protección de los datos.	Discernimiento y responsabilidad: <ul style="list-style-type: none">• Concepto.• Importancia.• Responsabilidad:<ul style="list-style-type: none">• Condiciones.• Tipos.	<ul style="list-style-type: none">• Explica la importancia de la ejecución de acciones con discernimiento y responsabilidad en el uso de los datos.• Relaciona características de las personas que actúan con responsabilidad y discernimiento.• Ejecuta procedimientos orientados a la protección y la integridad de los datos.



Resultados de aprendizaje	Saberes esenciales	Indicador de logro
		<ul style="list-style-type: none">• Aplica el discernimiento y la responsabilidad como parte importante del proceso de transmisión y análisis de la información.
7. Interpretar los Objetivos para el Desarrollo sostenible según lo establecido por la Asamblea General de las Naciones Unidas en la agenda 2030.	Objetivos y metas del Desarrollo Sostenible. <ul style="list-style-type: none">• ¿Qué son?• ¿Cuándo se crearon los objetivos de desarrollo sostenible?• ¿Cuál es su función?• ¿Cuáles son los ODS?	<ul style="list-style-type: none">• Reconoce la importancia de los Objetivos para el Desarrollo Sostenible (ODS) para la sociedad.• Distingue los 17 ODS.• Identifica las metas de los 17 ODS.



MINISTERIO DE
EDUCACIÓN PÚBLICA

GOBIERNO
DE COSTA RICA

Dirección de Educación
Técnica y Capacidades
Emprendedoras

Subject Area English Oriented to Software Applications Development XII





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 **Twelfth 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

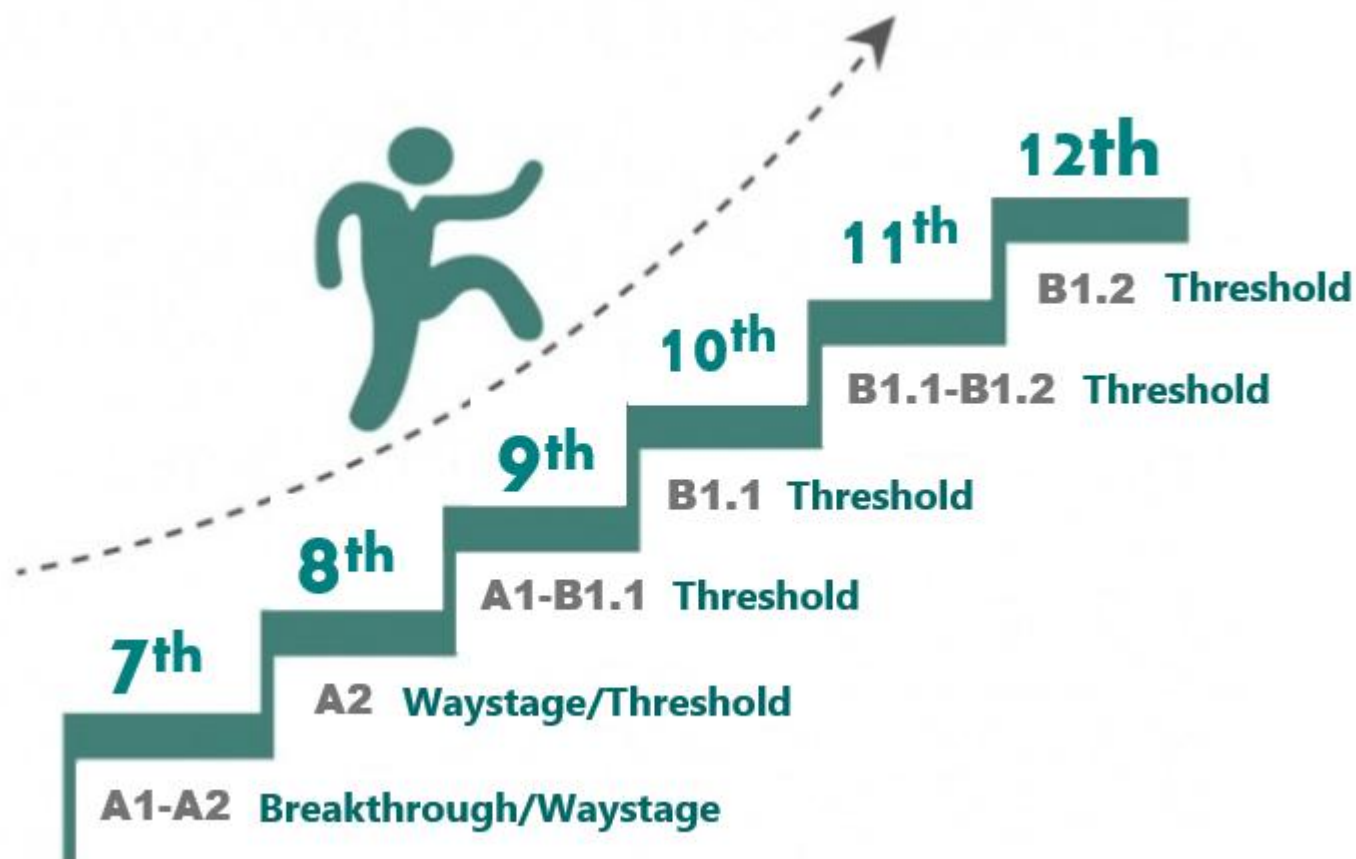




Table 1

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 capacities and interests of the students, 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 learners of foreign languages. This guideline contains standards for grading an individual's language proficiency. The Council of Europe established it as part of the "Language Learning for European Citizenship" project between 1989 and 1996. 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 and pragmatic skills 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 the 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 **Twelfth 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 understand 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:		



Planning

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



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

Curricular Structure English Oriented to Software Applications Development

Level: Twelfth Grade

Table 8

Themes per scenarios

Scenarios	Weekly Hours	Yearly Hours
Programming	4	52
Network and Virtualization	4	48
Total		100



Curricular Grid: English-Oriented to Software Applications Development

Level: Tenth

Table 7

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: Tenth

Scenario 1: Programming

Theme 1: Class Design and Exception Models

Goals:

Essential competence: Show initiative by independently designing a class with the right attributes and methods, recognizing possible errors, and adding exception handling to improve program reliability.

New citizenship axis: Apply digital citizenship with social equity by designing inclusive class structures and exception-handling models that consider diverse user needs, ensuring fair and accessible software solutions.

Listening:

Understand and follow detailed explanations about class design and exception models in programming, identifying key concepts and relationships from spoken instructions.

Reading: Read and understand moderately complex texts about class design and exception models in programming, identifying key concepts, relationships, and practical applications.



Spoken interaction: Engage in structured discussions about class design and exception models in programming, expressing their ideas clearly, asking relevant questions, and responding appropriately to peers' contributions.

Spoken production:

- Explain class design and exception models in programming, using appropriate terminology and structured reasoning to describe concepts, examples, and solutions.

Writing: Write clear and structured explanations about class design and exception models in programming, using appropriate terminology and examples to communicate their understanding effectively.

Theme 2: Mobile Applications

Goals:

Essential competence: Demonstrate creativity by designing and developing an innovative mobile application that solves a real-world problem, incorporating user-friendly features and intuitive functionality.

New citizenship axis: Strengthen their planetary citizenship by designing mobile applications that promote global awareness, social responsibility, and inclusivity, while reflecting a strong sense of personal and cultural identity.

Listening:



- Listen to and understand explanations about mobile application programming, identifying key concepts such as user interface design and functionality from spoken instructions.
- Follow spoken discussions on coding practices and app development, demonstrating their understanding by summarizing key points and asking relevant questions for clarification.

Reading: Read and comprehend texts about mobile application programming, identifying key concepts, coding techniques, and design principles, and applying this understanding to their development projects.

Spoken interaction:

- Participate in discussions about mobile application programming, sharing their ideas on app design and development while responding appropriately to peers' suggestions and feedback.
- Engage in problem-solving conversations, explaining coding challenges and solutions related to mobile app development and collaborating with others to find effective approaches.

Spoken production:

- Explain the process of designing and developing a mobile application, using clear language and appropriate technical terms to describe their project from concept to execution.
- Present their mobile app ideas, demonstrating the functionality and features of the app, and justifying design choices through structured, coherent speech.



Writing: Write clear and organized descriptions of their mobile application development process, explaining design choices, functionality, and coding techniques while using appropriate terminology and providing relevant examples.

Scenario 2: User Experience

Theme 1: User Experience

Goals:

Essential competence: Develop problem-solving skills by identifying and addressing key user experience challenges, designing intuitive interfaces, and creating solutions that enhance the overall usability of digital products.

New citizenship axis: Apply principles of digital citizenship with social equity by designing user experiences that are inclusive, accessible, and considerate of diverse user needs, ensuring fair access to digital tools for all.

Listening: Listen to and understand discussions about user experience principles, identifying key ideas related to usability, accessibility, and design from spoken explanations.

Reading: Read and comprehend texts about user experience design, identifying key concepts such as usability, accessibility, and user-centered design, and applying these principles to evaluate and improve digital interfaces.

Spoken interaction:



- Engage in discussions about user experience design, sharing their opinions on the effectiveness of various interface elements and responding thoughtfully to peers' feedback.
- Collaborate in groups to solve user experience challenges, offering suggestions, asking relevant questions, and providing constructive feedback on design ideas and solutions.

Spoken production:

- Explain key concepts of user experience design, including usability and accessibility, using clear language and appropriate terminology.
- Present their user experience design ideas, describing their thought process, the rationale behind their choices, and how they address user needs and challenges.

Writing: Write clear and organized descriptions of user experience design processes, outlining key decisions, design principles, and how their solutions meet user needs and improve usability.



Goals:

Essential competence: Demonstrate creativity by designing innovative and user-friendly interaction models, and exploring new ways to enhance user experience and engagement through unique interface elements and design features.

New citizenship axis: Incorporate sustainable development principles into their interaction design projects, creating user interfaces that promote environmental awareness, social responsibility, and equitable access for all users.

Listening:

- Listen to and understand explanations about interaction design principles in programming, identifying key ideas related to user interfaces, user experience, and design patterns.
- Actively listen to feedback and discussions on interaction design concepts, summarizing key points and asking clarifying questions to deepen their understanding and improve their design skills.

Reading: Read and comprehend texts about interaction design in programming, identifying key concepts such as usability, user flows, and interface design, and applying these principles to improve their projects.

Spoken interaction:

- Engage in discussions about interaction design in programming, sharing ideas on user interface best practices and responding appropriately to peers' feedback and suggestions.



- Collaborate in groups to brainstorm and problem-solve interaction design challenges, explaining their ideas clearly and providing constructive input to enhance design solutions.

Spoken production:

- Explain the key principles of interaction design in programming, describing how they apply these principles to create intuitive user interfaces and enhance user engagement.
- Present their interaction design projects, discussing design challenges, solutions, and the impact of their choices on the user experience, using clear and relevant terminology.

Writing: Write clear and structured explanations of interaction design principles in programming, detailing their design choices, user interface considerations, and how they enhance the overall user experience in their projects.

Curricular Design

Subject Area: English Oriented to Software Applications Development

Grade: Twelfth

CEFR : B1.2



Scenario 1: Programming

Theme 1: Class Design and Exception Models

Time: 20 hours

Essential Question: How can you take initiative in designing a class and handling errors to make your program work better and be more reliable?

Essential Competences: Initiative

New Citizenship Axis: Digital Citizenship with Social Equity

Table 8

Curriculum Pedagogical Design

Goals	Performance Indicator
The learners can...	The student...
Show initiative by designing a class with the right attributes and methods, recognizing possible errors, and adding exception handling to improve program reliability.	Takes initiative when planning the creation of a simple class and exception model in a program, identifying possible errors, and making improvements on their own.



Goals	Performance Indicator
The learners can...	The student...
Apply digital citizenship with social equity by designing inclusive class structures and exception-handling models that consider diverse user needs, ensuring fair and accessible software solutions.	Practice Digital Citizenship by using inclusive and respectful names in their class designs and creating clear error messages that make programs easier for everyone to understand.

TABLE 9

Oral and Written Comprehension

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



Listening:

- Understand and follow detailed explanations about class design and exception models in programming, identifying key concepts and relationships from spoken instructions.
- Listen to explanations about class design and exception models in programming, understand key concepts such as object attributes and methods, and learn how to handle errors effectively.
- Follow audio discussions on class structures and exception handling, identifying important details and asking relevant questions to clarify their understanding of the programming concepts.

Reading: Read and understand moderately complex texts about class design and exception models in programming,

-Read programming guides and examples on class structures and exception handling, applying the information to solve coding problems and enhance their own programming projects.



identifying key
concepts, relationships,
and practical
applications.

Table 10

Oral and Written Production

Goals The learners can...	Performance Indicator The student...
Spoken Interaction: Engage in structured discussions about class design and exception models in programming, expressing their ideas clearly, asking relevant questions, and responding appropriately to peers'	-Engages in discussions about class design and exception models in programming, sharing their ideas on how to structure classes and handle errors, while responding to feedback from peers. -Collaborates in groups to solve programming challenges related to class design and exception handling, asking relevant questions and offering solutions to improve the code's functionality and reliability.



contributions.	
Spoken Production: <ul style="list-style-type: none">• Explain class design and exception models in programming, using appropriate terminology and structured reasoning to describe concepts, examples, and solutions.	<ul style="list-style-type: none">-Explains their class design and exception-handling choices clearly, using appropriate programming terminology to describe how they structure their classes and manage errors.-Presents their solutions to class design and exception handling problems, justifying their approaches and explaining how they enhance program reliability and efficiency.
Writing: Write clear and structured explanations	<ul style="list-style-type: none">-Writes clear and organized explanations of their class design choices and exception-handling strategies, detailing the structure and functionality of their



about class design and exception models in programming, using appropriate terminology and examples to communicate their understanding effectively.

code.

Table 11

Learnings of the curriculum pedagogical design

Functions and Discourse Markers	Grammar	Vocabulary
------------------------------------	---------	------------



Functions

Learn how to design a class to store and manage language learning data, such as vocabulary words, sentences, and progress tracking. This includes defining attributes like word lists or study time.

Understand how to handle errors in user inputs, such as invalid word entries or incorrect sentence structure, using exception handling. This ensures that the

1. Present Perfect Continuous

The Present Perfect Continuous (also called the Present Perfect Progressive) is used to describe actions that started in the past and are still happening now or have recently stopped but have a visible effect in the present.

Structure:

Subject + have/has + been + verb (-ing)

Uses:

To describe an action that started in the past and is still happening.

I have been studying English for three years.

Class design in

programming refers to the **structuring and organization** of classes in an object-oriented system.

It involves defining

attributes (variables), methods (functions), and relationships

between different classes to ensure modularity, reusability, and maintainability of the code.

Good class design follows principles such as

encapsulation, inheritance, abstraction, and polymorphism (OOP concepts).

1. **Abstraction** – A principle in object-



<p>program can guide the user to provide valid input and continue without crashing.</p> <p>Discourse Markers</p> <p>Managing interaction (resuming or continuing)</p> <p>Connecting words for Illustration</p> <p>for example, for instance, namely, to illustrate, in other words, in particular, specifically, such as.</p>	<p>To talk about an action that recently stopped but has results in the present.</p> <p>She is tired because she has been working all day.</p> <p>To emphasize the duration of an activity.</p> <p>They have been waiting for the bus since 8 a.m.</p> <p>Examples:</p> <p>I have been reading this book for two hours.</p> <p>He has been playing football since this morning.</p> <p>We have been working on the project all week.</p> <p>2. Present Perfect vs. Past Simple</p> <p>Present Perfect: I have finished my homework. (We don't say when.)</p>	<p>oriented</p> <p>programming (OOP)</p> <p>that hides</p> <p>implementation details and only shows relevant features.</p> <p>2. Attribute – A variable within a class that holds data specific to an object.</p> <p>3. Class – A blueprint for creating objects that defines attributes and methods.</p> <p>4. Class Hierarchy – The structure that represents relationships</p>
--	---	--



<p>Managing Interaction (Resuming or Continuing)</p> <p>As I was saying, the new software update will improve efficiency.</p> <p>Anyway, let's get back to our main topic about office automation.</p> <p>To continue, we need to focus on training employees to use the new system.</p> <p>As I mentioned earlier, the use of cloud storage is essential for data security.</p>	<p>Past Simple: I finished my homework an hour ago. (Specific time given.)</p> <p>Present Perfect: She has lived here since 2015. (She still lives here.)</p> <p>Past Simple: She lived here for five years. (She doesn't live here anymore.)</p> <p>Present Perfect: We have never eaten sushi before. (Life experience.)</p> <p>Past Simple: We ate sushi last night. (A completed action.)</p> <p>References you may use:</p> <p>https://learnenglish.britishcouncil.org/grammar/b1-b2-grammar/present-perfect-simple-continuous</p> <p>https://test-english.com/explanation/b1-2/review-verb-tenses-b1/</p>	<p>between parent and child classes.</p> <p>5. Constructor – A special method in a class that is automatically called when an object is created.</p> <p>6. Catch Block – A section of code used to handle exceptions thrown in a try block.</p> <p>7. Data Encapsulation – The process of restricting direct access to class attributes and modifying them only</p>
--	---	--



Moving on, let's discuss the benefits of automation in reducing workload.

Connecting Words for Illustration

For example, many companies use Microsoft Excel for data analysis.

For instance, Google Docs allows multiple users to edit a document in real-time.

The software suite includes various applications, namely,

<https://www.englishradar.com/english-grammar/present-perfect-continuous/>

<https://www.cambridgeenglish.org/learnin-g-english/activities-for-learners/b1g003-past-simple-and-present-perfect>

<https://www.examenglish.com/CEFR/cefr-grammar.htm>

The Present Perfect and Past Simple are often confused. Here's how they differ:

Present Perfect	Past Simple
Describes an action that happened at an unspecified time before now.	Describes an action that happened at a specific time in the past.
Often used with just, already, yet, ever, never, since, for .	Often used with yesterday, last week, in 2010, two days ago .
Focuses on the result or experience .	Focuses on the completed action in the past.
Example: <i>I have visited Paris three times.</i> (We don't say when.)	Example: <i>I visited Paris last year.</i> (Specific time is mentioned.)

through methods.

8. **Dependency** – A relationship where one class relies on another to function.
9. **Destructor** – A special method called when an object is destroyed to free resources.

10. **Encapsulation** – The concept of bundling data and methods into a single unit (class) to restrict access.

11. **Exception** – An error or unexpected event that occurs during program



<p>Word, Excel, and PowerPoint.</p> <p>To illustrate, imagine a business that automates its emails, saving hours of manual work.</p> <p>Office automation enhances productivity; in other words, it helps employees complete tasks faster.</p> <p>The IT department must focus on cybersecurity, in particular, protecting sensitive customer data.</p>		<p>execution.</p> <p>12.Exception Handling – A mechanism to handle runtime errors and prevent program crashes.</p> <p>13.Extensibility – The ability of a class to be modified or extended without changing its existing behavior.</p> <p>14.Helper Method – A private or protected method used within a class to assist main methods.</p> <p>15.Inheritance – A feature in OOP that</p>
---	--	--



<p>This training is designed specifically for administrative professionals.</p> <p>Office tools such as spreadsheets and databases are crucial for managing company records.</p> <p>Idioms and common colloquial expressions</p> <p>"Break the ice"</p> <p>Meaning: To initiate conversation or start a discussion, especially in a way that makes people feel more comfortable.</p>		<p>allows a class (child) to derive attributes and methods from another class (parent).</p> <p>16.Instance – A specific object created from a class.</p> <p>17.Interface – A contract that defines methods a class must implement but does not provide their implementations.</p> <p>18.Instantiation – The process of creating an object from a class.</p>
---	--	--



<p>Example: "Let's break the ice by discussing some basic principles of class design before diving into the exceptions."</p> <p>2. "In the loop"</p> <p>Meaning: Being informed or involved in something.</p> <p>Example: "It's important to keep the whole team in the loop about the exception handling strategy we're adopting."</p> <p>3. "By the book"</p> <p>Meaning: Following the rules or instructions</p>		<p>19.Logic Error – A bug in the program where the code runs but produces incorrect results.</p> <p>20.Method – A function defined inside a class that operates on objects.</p> <p>21.Method Overloading – Defining multiple methods in the same class with the same name but different parameters.</p> <p>22.Method Overriding – A child class redefining a method</p>
--	--	---



exactly as they are laid out.

Example: "We need to design this class by the book to ensure that all exceptions are handled properly."

4. "Back to the drawing board"

Meaning: To start over from scratch because the current attempt has failed.

Example: "Our current class design doesn't handle the exception properly. It's back to the drawing board for us."

from its parent class to provide specific behavior.

23. **Modularity** – The concept of designing a system with separate components or classes for easier maintenance.

24. **Object** – An instance of a class that contains its own data and behaviors.

25. **Object-Oriented Programming (OOP)** – A programming paradigm that organizes code into



5. "A fine-tuned machine"

Meaning: Something that works very efficiently and smoothly.

Example: "With the proper class design and exception handling, our program will be a fine-tuned machine."

6. "Catch-22"

Meaning: A paradoxical situation where an action cannot be taken due to contradictory rules.

Example: "It's a catch-22 situation; the class

reusable classes and objects.

26. Polymorphism –

The ability of different objects to respond to the same method call in different ways.

27. Private Access

Modifier – Restricts access to class members only within the class itself.

28. Protected Access

Modifier – Allows access to class members within the same class and subclasses.

29. Public Access



design has to be perfect for exceptions to work, but we can't test it without handling those exceptions."

7. "Throw in the towel"

Meaning: To give up or admit defeat.

Example: "After several failed attempts at exception handling, I was about to throw in the towel, but then I found a better approach."

8. "Hit the nail on the head"

Modifier – Allows access to class members from anywhere in the program.

30. Runtime Error –

An error that occurs while the program is running, often leading to exceptions.

31. Static Method – A method that belongs to a class rather than an instance and can be called without creating an object.

32. Subclass – A class



Meaning: To describe exactly what is causing a situation or problem.

Example: "When the developer suggested restructuring the class, they hit the nail on the head for solving the exception handling issue."

that inherits from another class (also called a derived class or child class).

33. Superclass – A class from which other classes inherit attributes and methods (also called a parent class).

34. Throw Statement
– A statement used to signal that an exception has occurred.

35. Try Block – A section of code that is monitored for exceptions, followed by a catch block to



handle errors.

Curricular Design

Subject Area: English Oriented to Software Applications Development

Grade: Twelfth

CEFR : B1.2

Scenario 1: Programming

Theme 2: Mobile Applications

Time: 32 hours

Essential Question: How can creativity be applied to the design and development of mobile applications to enhance user experience and solve real-world problems?

Essential Competences: Creativity

New Citizenship Axis: Strengthening of Planetary Citizenship with Identify



Table 16

Curriculum Pedagogical Design

Goals	Performance Indicator
The learners can...	The student...
Demonstrate creativity when designing and developing an innovative mobile application that solves a real-world problem, incorporating user-friendly features and intuitive functionality.	Uses the correct language competencies to address the goal when designing and developing a basic mobile application that addresses a real-world problem, integrating user-friendly features and intuitive navigation , while making minor improvements based on feedback
Strengthen their planetary citizenship by thinking about designing mobile applications that promote global awareness, social responsibility, and inclusivity, while reflecting a strong sense of personal and cultural identity.	Uses the correct language competencies to address the goal when designing and developing a basic mobile application that raises global awareness and promotes social responsibility , incorporating elements that reflect the cultural identity and inclusive values in its content and design.

TABLE 17

Oral and Written Comprehension



Goals	Performance Indicator
The learners can...	The student...
<p>Listening:</p> <p>-Listen to and understand explanations about mobile application programming, identifying key concepts such as user interface design and functionality from spoken instructions.</p> <p>-Follow spoken discussions on coding practices and app development, demonstrating their understanding by summarizing key points</p>	<ul style="list-style-type: none">• Understands clear, structured explanations about mobile application programming, identifying key concepts such as user interface design and functionality when presented in standard speech at a moderate pace.• Learners can follow spoken discussions on coding practices and app development, accurately summarizing key points and formulating relevant questions to clarify technical concepts.



and asking relevant questions for clarification.	
Reading: Read and comprehend texts about mobile application programming, identifying key concepts, coding techniques, and design principles, and applying this understanding to their development projects.	Reads and comprehends texts about mobile application programming, identifying key concepts, coding techniques, and design principles, and applying this understanding to their development projects.

Table 18

Oral and Written Production

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



Spoken Interaction:

-Participate in discussions about mobile application programming, sharing their ideas on app design and development while responding appropriately to peers' suggestions and feedback.

-Engage in problem-solving conversations, explaining coding challenges and solutions related to mobile app development and collaborating with others to find effective

- Engages in discussions about mobile application programming, exchanging ideas on coding techniques and design principles, while asking and responding to questions to clarify concepts and solve problems collaboratively.
- Participates in problem-solving conversations about mobile app development, clearly explaining coding challenges, proposing solutions, and collaborating with others to find effective approaches.



approaches.

Spoken Production:

- | | |
|--|--|
| <ul style="list-style-type: none">• Explain the process of designing and developing a mobile application, using clear language and appropriate technical terms to describe their project from concept to execution.• Present their mobile app ideas, demonstrating the functionality and features of the app, and justifying design choices through structured, | <ul style="list-style-type: none">• Clearly explains the process of designing and developing a mobile application, using appropriate technical terms to describe their project from concept to execution in a structured and coherent manner.• Explains their mobile app ideas clearly, demonstrating the functionality and features of the app while justifying their design choices through structured and coherent speech. |
|--|--|



coherent speech.	
Writing: Write clear and organized descriptions of their mobile application development process, explaining design choices, functionality, and coding techniques while using appropriate terminology and providing relevant examples.	<ul style="list-style-type: none">Writes clear and structured texts about mobile application development, explaining coding processes, design choices, and app functionality using appropriate technical terminology.

Table 19

Learnings of the curriculum pedagogical design



Functions and Discourse Markers	Grammar	Vocabulary
<p>Functions</p> <p>-Explain how a mobile app works, what features it has, and its purpose.</p> <p>-Give instructions and explain processes to guide users on how to use an app effectively.</p> <p>-Express opinions about mobile applications, comparing their usefulness and usability.</p> <p>Discourse Markers: Managing interaction (resuming or</p>	<p>Reported Speech (Range of Tenses) (Based on CEFR B1.2 Level – Intermediate)</p> <p>1. What is Reported Speech?</p> <p>Reported speech (also called indirect speech) is used to tell what someone else has said without quoting their exact words. Instead of using quotation marks, we change the tense of the original statement and adjust pronouns, time expressions, and modal verbs accordingly.</p> <p>2. Changing Tenses in Reported Speech</p>	<p>Accessibility: Designing applications that are easy to use for people with disabilities.</p> <p>Over-The-Air Updates (OTA Updates): Distribution of new app versions without the need for manual reinstallation.</p> <p>Mobile Analytics: Tracking user behavior within the app to optimize the experience and marketing strategies.</p> <p>Animations and Transitions: Visual effects that enhance the user interaction with the application.</p>



Functions and Discourse

Markers

continuing)

Connecting words for

Illustration

for example, for instance,
namely, to illustrate, in
other words, in particular,
specifically, such as.

Managing Interaction

(Resuming or Continuing)

As I was saying, the new
software update will
improve efficiency.
Anyway, let's get back to
our main topic about
office automation.
To continue, we need to
focus on training

Grammar

When reporting speech, the verb in direct
speech often moves one step back in tense
(backshifting).

Direct Speech	Reported Speech
Present Simple: "I like coffee."	Past Simple: She said she liked coffee.
Present Continuous: "I am studying."	Past Continuous: He said he was studying.
Past Simple: "I visited Rome."	Past Perfect: She said she had visited Rome.
Present Perfect: "I have finished my work."	Past Perfect: He said he had finished his work.
Past Continuous: "I was watching TV."	Past Perfect Continuous: She said she had been watching TV.
Future (will): "I will call you later."	Would: He said he would call me later.

3. No Change in Tense (When Not to Backshift)

- If the reporting verb is in the present
tense, the original tense remains the same.
He says, "I am happy." → He says he is
happy.

Vocabulary

API (Application

Programming Interface):

A
set of functions that allow
applications to interact with
external services or operating
system components.

Mobile Application: Software
designed to run on mobile
devices such as smartphones
and tablets.

Hybrid Applications: Apps
that combine web
technologies (HTML, CSS,
JavaScript) and run in a
native container (e.g., Ionic,
Apache Cordova).

Native Applications: Apps



Functions and Discourse

Markers

employees to use the new system.

As I was mentioning earlier, the use of cloud storage is essential for data security. Moving on, let's discuss the benefits of automation in reducing workload.

Connecting Words for Illustration

For example, many companies use Microsoft Excel for data analysis. For instance, Google Docs allows multiple users to edit a document in real-time.

Grammar

- If the original statement refers to a general truth or fact, the tense remains the same.

She said, "The sun rises in the east." → She said the sun rises in the east.

- If the past perfect is used in direct speech, no change is necessary.

He said, "I had already eaten." → He said he had already eaten.

4. Changing Time and Place Expressions

Certain time and place expressions change in reported speech:

Direct Speech	Reported Speech
today → that day	now → then
yesterday → the day before	tomorrow → the next day
next week → the following week	last week → the previous week
here → there	this → that

Vocabulary

developed specifically for a mobile operating system, using platform-specific tools and languages (e.g., Swift for iOS, Kotlin for Android).

Progressive Web Applications

(PWA): Web-based apps that offer a native app-like experience without needing installation.

Common Architectures:

Patterns to organize code and facilitate app maintenance.

MVC (Model-View-

Controller): A pattern that separates logic from



Functions and Discourse Markers	Grammar	Vocabulary
<p>The software suite includes various applications, namely, Word, Excel, and PowerPoint.</p> <p>To illustrate, imagine a business that automates its emails, saving hours of manual work.</p> <p>Office automation enhances productivity; in other words, it helps employees complete tasks faster.</p> <p>The IT department must focus on cybersecurity, in particular, protecting sensitive customer data.</p>	<p>5. Examples of Reported Speech</p> <p>Direct Speech: "I am learning English."</p> <p>Reported Speech: She said she was learning English.</p> <p>Direct Speech: "We went to the park yesterday."</p> <p>Reported Speech: They said they had gone to the park the day before.</p> <p>Direct Speech: "I will help you tomorrow."</p> <p>Reported Speech: He said he would help me the next day.</p>	<p>interface and event handling.</p> <p>MVVM (Model-View-ViewModel): A pattern that facilitates data binding between logic and interface.</p> <p>Clean Architecture: A modular and scalable architecture for mobile applications.</p> <p>Biometric Authentication: The use of physical features such as fingerprints or facial recognition to validate access to the app.</p> <p>Backend: The part of the application that handles</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>This training is designed specifically for administrative professionals. Office tools such as spreadsheets and databases are crucial for managing company records.</p> <p>Idioms and common colloquial expressions</p> <p>"At Your Fingertips" Meaning: Easily accessible on a mobile device.</p>	<p>Direct Speech: "I have finished my homework."</p> <p>Reported Speech: She said she had finished her homework.</p> <p>References you may use:</p> <p>British Council – Reported Speech https://learnenglish.britishcouncil.org/grammar/english-grammar-reference/reported-speech</p> <p>Exam English – CEFR Grammar Levels https://www.examenglish.com/CEFR/cefr_grammar.htm</p> <p>All Things Grammar – CEFR Levels</p>	<p>business logic, data storage, and communication with the server.</p> <p>Mobile Database: A data storage system optimized to work on mobile devices (e.g., SQLite, Room, Core Data).</p> <p>Cache: Temporary data storage to improve performance and reduce resource consumption.</p> <p>Compatibility: Ensuring that the application works correctly across different operating system versions and devices.</p> <p>Debugging: The process of</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>Example: "With this app, all your banking needs are at your fingertips!"</p> <p>"Tap Into"</p> <p>Meaning: To make use of something, often digital resources or services.</p> <p>Example: "This fitness app helps users tap into personalized workout plans."</p> <p>"There's an App for That"</p> <p>Meaning: Used humorously to suggest that an app exists for almost any task.</p>	<p>https://www.allthingsgrammar.com/cefr-levels.html</p> <p>Grade University – Key Grammar Structures for Every Proficiency Level</p> <p>https://grade-university.com/blog/typical-grammar-structures-for-each-level</p> <p>Scribd – Reported Speech I (A2 & B1.2)</p> <p>https://www.scribd.com/document/714916111/Reported-Speech-I-A2-B1-2</p> <p>Simple passive1.</p> <p>What is the Passive Voice?</p> <p>The passive voice is used when the focus is on the action rather than who is performing</p>	<p>identifying and fixing errors in a mobile application.</p> <p>Cross-Platform Development:</p> <p>Creating apps that work on multiple operating systems from a single codebase (e.g., Flutter, React Native).</p> <p>Emulator: Software that simulates a mobile device on a computer for testing and debugging applications.</p> <p>Firebase: Google's platform offering backend services like authentication, real-time database, push notifications, and analytics.</p> <p>Frontend: The part of the</p>



Functions and Discourse

Markers

Example: "You need help tracking your sleep?

There's an app for that!"

"Swipe Right"

Meaning: Originally from dating apps, now used to mean approving or liking something.

Example: "If you love online shopping, you'll definitely swipe right on this app!"

"Crash and Burn"

Meaning: When an app or system fails completely.

Example: "The new game launched yesterday, but it

Grammar

it. The subject of the sentence receives the action instead of doing it.

2. Structure of the Passive Voice

Subject + be (in the correct tense) + past participle (+ by + agent, if necessary)

3. Forming the Simple Passive

The passive voice can be used in different simple tenses as follows:

Active Voice	Passive Voice
Present Simple: <i>They clean the house every day.</i>	Present Simple Passive: <i>The house is cleaned every day.</i>
Past Simple: <i>She wrote the book in 2020.</i>	Past Simple Passive: <i>The book was written in 2020.</i>
Future Simple: <i>They will deliver the package tomorrow.</i>	Future Simple Passive: <i>The package will be delivered tomorrow.</i>

4. When to Use the Passive Voice

Vocabulary

application that interacts directly with the user, displaying the graphical interface.

Geolocation: Technology that determines the exact location of a device via GPS, Wi-Fi, or mobile networks.

State Management: A technique for managing the dynamic state of an app, ensuring that data and interfaces remain consistent. Integrated Development

Environment (IDE): An environment that facilitates writing, debugging, and



Functions and Discourse Markers	Grammar	Vocabulary
<p>crashed and burned due to too many bugs."</p> <p>"In Real Time"</p> <p>Meaning: Instantly or live, as it happens.</p> <p>Example: "This app lets you track your package in real time."</p> <p>"A Game-Changer"</p> <p>Meaning: A revolutionary idea or technology that transforms how things work.</p> <p>Example: "Mobile payment apps have been a game-</p>	<p>When the person doing the action (the agent) is unknown, unimportant, or obvious.</p> <p>A new bridge was built last year. (We don't know or need to mention who built it.)</p> <p>When the focus is on the action or result rather than the doer.</p> <p>The emails are sent every morning. (Focus on the emails, not who sends them.)</p> <p>In formal writing (e.g., scientific or official reports).</p> <p>The experiment was conducted by researchers.</p> <p>5. Including or Omitting the Agent</p>	<p>testing applications (e.g., Android Studio, Xcode).</p> <p>Monetization: Strategies to generate revenue from a mobile application.</p> <p>Freemium: Offering a free version with premium paid features.</p> <p>Advertising: Integrating ads within the app.</p> <p>In-App Purchases: Selling additional content or features within the app.</p> <p>Subscriptions: Recurring charges for access to exclusive features.</p> <p>Push Notifications: Messages</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>changer for small businesses."</p> <p>"Plugged In"</p> <p>Meaning: Being connected and up-to-date with technology.</p> <p>Example: "Developers need to stay plugged in to the latest trends in mobile app design."</p>	<p>If we want to mention who performed the action, we use "by + agent".</p> <p>The Mona Lisa was painted by Leonardo da Vinci.</p> <p>If the agent is obvious or unimportant, it is often omitted.</p> <p>A decision was made yesterday. (We don't need to say "by the manager.")</p> <p>6. Examples of Simple Passive Sentences</p> <p>Present Simple:</p> <p>Active: The chef cooks the meals.</p> <p>Passive: The meals are cooked by the chef.</p> <p>Past Simple:</p> <p>Active: Someone stole my phone.</p>	<p>sent directly to users from a server to keep them informed or interact with them.</p> <p>Performance Optimization:</p> <p>The process of improving an app's speed, resource consumption, and load time.</p> <p>Permissions: Requests made by an app to access specific device functions or data (e.g., camera, location, contacts).</p> <p>Publishing: The process of uploading a mobile app to an app store like Google Play Store or Apple App Store.</p> <p>Responsiveness: The ability of</p>



Functions and Discourse Markers	Grammar	Vocabulary
	<p>Passive: My phone was stolen.</p> <p>Future Simple:</p> <p>Active: They will announce the results soon.</p> <p>Passive: The results will be announced soon.</p> <p>References you may use:</p> <p>Simple Passive:</p> <p>Perfect English Grammar: Explains the formation and usage of the passive voice, including its structure across various tenses.</p> <p>https://www.perfect-english-grammar.com/passive.html</p> <p>British Council - LearnEnglish: Offers insights into the passive voice, including how it's</p>	<p>an app to adjust and function properly on devices with different screen sizes and resolutions.</p> <p>Software Development Kit (SDK): A set of tools and libraries that enable the development of apps for a specific platform (e.g., Android SDK, iOS SDK).</p> <p>Mobile Security: Practices to protect user data and privacy in mobile applications.</p> <p>Mobile Operating System: Software that manages resources and apps on a</p>



Functions and Discourse Markers	Grammar	Vocabulary
	<p>formed and when to use it, along with practice exercises.</p> <p>https://learnenglish.britishcouncil.org/grammar/b1-b2-grammar/passives</p> <p>Ginger Software - English Grammar Rules: Provides rules and examples for using the passive voice, highlighting its structure and appropriate contexts.</p> <p>https://www.gingersoftware.com/content/grammar-rules/verbs/passive-voice</p>	<p>mobile device (e.g., Android, iOS).</p> <p>Testing: Evaluating the functionality, performance, and usability of a mobile app before its release.</p> <p>Unit Testing: Verifying individual components of the app.</p> <p>UI Testing: Validating user interaction with the application.</p> <p>Integration Testing: Checking the interaction between different app modules.</p> <p>Load Times: The time it takes for an app to be ready for</p>



Functions and Discourse Markers	Grammar	Vocabulary
		use after opening. UI/UX (User Interface & User Experience): Visual and functional design of the application, focused on ease of use and user satisfaction. Versioning: Identifying app updates and changes through version numbers.

Curricular Design

Subject Area: English Oriented to Software Applications Development

Grade: Tenth

CEFR : B1.1



Scenario 2: User Experience

Theme 1: User Experience

Time: 28 hours

Essential Question: How can understanding user experience (UX) principles help in identifying and solving usability issues to create more effective and intuitive solutions?

Essential Competences: Problem solving

New Citizenship Axis: Digital Citizenship with Social Equity

Table 20

Curriculum Pedagogical Design

Goals	Performance Indicator
The learners can...	The student...
Develop problem-solving skills by identifying and addressing key user experience challenges, designing intuitive interfaces, and creating solutions that enhance the overall usability of digital products.	<ul style="list-style-type: none">Identifies key user experience challenges in digital products, proposes solutions to improve usability, and describes the design process for creating intuitive interfaces, demonstrating their



Goals	Performance Indicator
The learners can...	The student...
	understanding through clear explanations and examples
Apply principles of digital citizenship with social equity by designing user experiences that are inclusive, accessible, and considerate of diverse user needs, ensuring fair access to digital tools for all.	<ul style="list-style-type: none">Implements principles of digital citizenship by designing inclusive and accessible user experiences that address diverse user needs, ensuring fair access to digital tools, and explaining their design choices to promote social equity.

TABLE 21

Oral and Written Comprehension

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



<p>Listening: Listen to and understand discussions about user experience principles, identifying key ideas related to usability, accessibility, and design from spoken explanations.</p>	<p>Understands discussions about user experience principles, identifying key ideas related to usability, accessibility, and design, and summarizes these ideas in their own words with clarity.</p>
<p>Reading: Read and comprehend texts about user experience design, identifying key concepts such as usability, accessibility, and user-centered design, and applying these principles to evaluate and improve digital interfaces.</p>	<p>Comprehends texts about user experience principles, identifying key concepts related to usability, accessibility, and design, and demonstrates understanding by applying these concepts to real-world examples.</p>



Table 22

Oral and Written Production

Goals		Performance Indicator
The learners can...	The student...	
<p>Spoken Interaction: --</p> <p>Engage in discussions about user experience design, sharing their opinions on the effectiveness of various interface elements and responding thoughtfully to peers' feedback.</p> <p>-Collaborate in groups to solve user experience challenges, offering suggestions, asking relevant questions, and providing constructive</p>	<ul style="list-style-type: none">• Engages in discussions about user experience design, effectively sharing their opinions on the effectiveness of various interface elements and responding to peers' feedback with relevant suggestions and insights.• Participates in groups to solve user experience challenges, offering relevant suggestions, asking clarifying questions, and providing constructive feedback on design ideas and solutions clearly and respectfully.	



feedback on design ideas and solutions.	
Spoken Production: -Explain key concepts of user experience design, including usability and accessibility, using clear language and appropriate terminology. -Present their user experience design ideas, describing their thought process, the rationale behind their choices, and how they address user needs and challenges.	<ul style="list-style-type: none">• Explains key concepts of user experience design, including usability and accessibility, using clear language and appropriate terminology, and provides examples to support their explanations.• Present their user experience design ideas, clearly describing their thought process, the rationale behind their design choices, and how their solutions address user needs and challenges, using appropriate terminology and supporting explanations.
Writing: Write clear and	<ul style="list-style-type: none">• Writes clear and detailed explanations of their user experience design



organized descriptions of user experience design processes, outlining key decisions, design principles, and how their solutions meet user needs and improve usability.	ideas, describing the steps they took, the rationale behind their choices, and how their designs meet user needs and improve usability.
--	---

Table 23

Learnings of the curriculum pedagogical design

Functions and Discourse Markers	Grammar	Vocabulary
Functions Develop the ability to describe user needs and behaviors clearly in written and spoken forms, using appropriate language to analyze and interpret feedback	Grammar Topic: Wh- Questions in the Past (Based on the European Framework B1.2 Band) Overview:	404 Error: Message informing the user that a requested page or resource is unavailable, with design considerations to minimize frustration. A/B Testing: Comparison of two



Functions and Discourse Markers	Grammar	Vocabulary
<p>from users in a way that informs the design process.</p> <p>Articulate their design choices effectively, explaining the rationale behind their decisions and presenting their prototypes with clear, structured language that highlights key features and user benefits.</p> <p>Give and receive constructive feedback on user experience designs, using appropriate technical and professional language to suggest improvements, ask relevant questions, and refine their designs based on peer input.</p>	<p>Wh- questions in the past are used to ask for specific information about actions or events that happened in the past. These questions start with a "wh-" word (e.g., what, where, when, why, who, how) and are followed by the past form of the verb.</p> <p>At the B1.2 level, students are expected to use these questions accurately in different contexts to inquire about past events or situations.</p> <p>Key Structure:</p> <p>Wh- word + past tense verb + subject + (optional object)?</p> <p>Examples:</p>	<p>design versions to determine which provides a better user experience.</p> <p>Accessibility: Designing digital products to be usable by people with disabilities.</p> <p>Anthropometry Digital: Consideration of physical dimensions and human limitations when designing interfaces and devices.</p> <p>Card Sorting: Technique for organizing and categorizing content based on how users group and label information.</p> <p>Color Theory: Strategic use of colors to convey emotions, highlight elements, and</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>Discourse Markers</p> <p>Describing hopes and plans</p> <p>Making predictions</p> <p>Giving presentations</p> <p>Connecting words Examples</p> <p>For example, for instance, to illustrate, in other words, as an illustration, in particular.</p> <p>Connecting words</p> <p>Details</p> <p>Specifically, especially, in particular, to explain, to list, to enumerate, in detail, namely, including.</p> <p>Emphasis</p> <p>Above all, indeed, truly, of course, certainly, surely, in fact,</p>	<p>What did you do yesterday?</p> <p>(Asking about a specific action in the past.)</p> <p>Where did she go on vacation last summer?</p> <p>(Inquiring about a past destination.)</p> <p>When did they arrive at the party?</p> <p>(Asking about the time of an event in the past.)</p> <p>Why did you choose that course last year?</p> <p>(Inquiring about a reason in the past.)</p> <p>Who did you meet at the conference?</p>	<p>enhance the visual experience.</p> <p>Context of Use: Circumstances and environments in which users interact with a product, influencing design.</p> <p>Design Centred on the User: A design process that places the needs, goals, and behaviors of the user at the center of decision-making.</p> <p>Design Emotional: Focus on how a design evokes emotions and creates a positive connection with users.</p> <p>Design Iterative: Process of conducting tests, gathering feedback, and continually improving the design.</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>really, in truth, Connecting words</p> <p>Summary</p> <p>Finally, thus, in short, in conclusion, in brief, as a result, accordingly.</p> <p>Describing Hopes and Plans</p> <p>I hope to complete my certification in office automation by the end of this year.</p> <p>Our company plans to introduce a new automated payroll system next month.</p> <p>We aim to train all employees on the latest office software within six months.</p> <p>I wish to improve my skills in Excel and database management for better efficiency.</p>	<p>(Asking about the person someone met in the past.)</p> <p>How did you learn to play the guitar?</p> <p>(Inquiring about the method or process used in the past.)</p> <p>Key Points:</p> <p>The verb in the question is in the past tense (e.g., did).</p> <p>The subject follows the verb (e.g., did you, did she).</p> <p>The wh- question word determines the type of information being asked.</p> <p>These types of questions help students gather detailed information</p>	<p>Design Interaction: Creation of flows and behaviors that define how users interact with a system.</p> <p>Design of Prototyping: Functional model of a design allowing testing of interactions and flows before full implementation.</p> <p>Design of Responsive: Automatic adaptation of the interface to work well across different screen sizes and devices.</p> <p>Design Patterns: Proven solutions to common interface problems, such as dropdown menus or search bars.</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>The school intends to integrate automation tools into the curriculum soon.</p> <p>Making Predictions</p> <p>I predict that office automation will replace most manual data entry tasks within five years.</p> <p>It is likely that businesses will adopt AI-driven tools for office management.</p> <p>In the future, employees will rely more on cloud-based collaboration tools.</p> <p>Experts foresee a rise in remote work due to automation and digital communication.</p> <p>Chances are that traditional filing systems will become obsolete.</p>	<p>about past experiences, actions, and events.</p> <p>References you may use:</p> <p>Here are the links to the sources related to the topic of Wh- questions in the past:</p> <p>Simple Past: Wh- Questions</p> <p>https://continuingstudies.uvic.ca/elc/studyzone/330/grammar/pastnq3</p> <p>Wh- Questions in the Past Simple Tense</p> <p>https://englishacademy101.com/wh-questions-in-the-past-simple-tense</p> <p>Past Tense - Questions - Who/Why - Learn English Grammar</p>	<p>Design Criteria: Rules and principles that guide design decisions to meet user and business goals.</p> <p>Empathy: The ability to understand and design for the emotions, needs, and motivations of the user.</p> <p>Feedback from the User:</p> <p>Opinions and suggestions provided by users about their experience with a product.</p> <p>Fidelity of Prototypes: Level of detail and functionality of a prototype, from low fidelity (sketches) to high fidelity (full interactions).</p> <p>Field Testing: Evaluation of the</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>Giving Presentations</p> <p>Let me start by saying that office automation has transformed the workplace.</p> <p>Now, let's move on to the benefits of using automation tools.</p> <p>As you can see, automation reduces errors and increases productivity.</p> <p>To summarize, implementing automation leads to time and cost savings.</p> <p>I'd like to conclude by emphasizing the importance of adapting to new technologies.</p> <p>Connecting Words - Examples</p>	<p>https://www.talkenglish.com/video/lessondetails.aspx?ALID=4320</p> <p>Wh- Questions in the Past Simple Tense</p> <p>https://englishacademy101.com/wh-questions-in-the-past-simple-tense</p> <p>What Are Wh- Questions in English?</p> <p>https://ellii.com/blog/wh-questions</p> <p>Grammar Topic: Will and Going to for Prediction in the Past (Based on the European Framework B1.2 Band)</p> <p>Overview:</p> <p>At the B1.2 level, students learn how to express predictions about the past</p>	<p>user experience in a real-world environment, outside of a controlled lab.</p> <p>Heat Maps: Visual representations showing the most interacted areas of an interface.</p> <p>Heuristic Analysis: Usability evaluation of a system based on established principles, such as Nielsen's heuristics.</p> <p>Hierarchy Visual: Organization of elements in an interface to guide user attention to the most important areas.</p> <p>Main Task: Key action a user must perform in a system to achieve their goal.</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>For example, many companies use CRM software to manage customer data efficiently. For instance, Microsoft Teams allows remote teams to collaborate seamlessly. To illustrate, consider a small business that automates invoicing, reducing human error. Spreadsheets help in organizing financial records; in other words, they simplify accounting. Cloud storage services, as an illustration, provide secure access to files from anywhere. Employees must develop digital skills; in particular, proficiency in office applications.</p>	<p>using will and going to. While both forms are typically used for future predictions, they can also be used to talk about past situations or events where something was expected or predicted.</p> <p>Key Concepts:</p> <p>Will for Prediction (Past Context)</p> <p>Will is used to express what someone thought would happen in the past, often with the idea of a future expectation that didn't come to pass as expected. In past contexts, it's usually used with a past time reference.</p> <p>Going to for Prediction (Past Context)</p>	<p>Journey Map (User Journey Map): Visual representation of the user's journey when interacting with a product, identifying friction points and opportunities for improvement.</p> <p>Loading States: Screens or animations that inform the user that a system is processing their request.</p> <p>Microinteractions: Small interactive details that enhance the user experience by providing immediate feedback (e.g., animations when clicking).</p> <p>Metrics UX: Key indicators to measure user experience, such</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>Connecting Words - Details</p> <p>This course covers office automation tools, specifically Word, Excel, and Outlook. Automation is beneficial for all businesses, especially those handling large amounts of data. Security measures, in particular, data encryption and backup, are crucial in automation. To explain, automation reduces repetitive tasks, allowing employees to focus on strategic work. Let's list the key features of office automation: email management, cloud storage, and workflow automation.</p>	<p>Going to is used to talk about predictions or plans that were made before a past event or situation. It reflects the idea of an intention or prediction that was expected to happen at a certain time in the past.</p> <p>Key Structure:</p> <p>Will + subject + base verb (for prediction about the past)</p> <p>Be going to + base verb (for plans or predictions about the past)</p> <p>Examples:</p> <p>Will for Past Prediction:</p> <p>She will probably finish the project last night. (Indicating what was</p>	<p>as task completion time, error rates, and user satisfaction.</p> <p>Onboarding: Process of guiding users through their first interaction with a product to help them understand its functionality.</p> <p>Persona: Fictitious representation of a typical user based on real data, helping guide design decisions.</p> <p>Prototyping: Functional model of a design allowing testing of interactions and flows before full implementation.</p> <p>Responsive Design: Adaptation of an interface to work effectively on various screen</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>Office automation includes various tools, namely, document management systems, scheduling software, and virtual assistants.</p> <p>Employees should learn to use automation tools, including AI-powered chatbots.</p> <p>Emphasis</p> <p>Above all, automation improves efficiency and reduces workload.</p> <p>Indeed, businesses that embrace automation see significant productivity gains.</p> <p>Using the latest software is essential, truly making work easier.</p>	<p>expected, but it may not have happened as predicted.)</p> <p>They will arrive at the airport by 9 p.m. yesterday. (Referring to a past prediction about their arrival.)</p> <p>Going to for Past Prediction:</p> <p>I was going to call you last night, but I forgot. (Indicating a planned action in the past that didn't happen.)</p> <p>He was going to tell her the news yesterday, but he didn't. (Referring to an intended prediction or action in the past.)</p> <p>Key Points:</p>	<p>sizes and devices.</p> <p>SEO (Search Engine Optimization): Design of websites to improve visibility in search engines, related to user experience.</p> <p>Site Map: Diagram showing the structure and hierarchy of pages on a website or application.</p> <p>System of Design: Set of reusable components and guidelines that ensure consistency in product design.</p> <p>Task Flow: Path a user follows when interacting with a product to complete specific tasks.</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>Office automation is necessary, of course, for a competitive business environment.</p> <p>Certainly, digital tools streamline office operations.</p> <p>Surely, automated workflows can save employees a lot of time.</p> <p>In fact, many companies have already replaced manual processes with automation.</p> <p>Really, office automation is transforming the way businesses operate.</p> <p>In truth, technology is shaping the future of work.</p> <p>Connecting Words - Summary</p>	<p>Will is often used in statements where someone made a prediction or expectation in the past about what would happen.</p> <p>Going to refers to past plans or predictions that were made before the event occurred but possibly didn't happen as intended.</p> <p>Both are used when you reflect on or talk about things that were anticipated or planned, but did not necessarily occur as expected in the past.</p> <p>Additional Notes: This structure helps learners to express ideas of missed or unrealized predictions in the past and can be useful in recounting</p>	<p>Usability: The ease with which users can learn, use, and achieve their goals with a system.</p> <p>Usability Testing: Evaluations conducted with real users to identify problems and improve interactions.</p> <p>User Feedback: Opinions and suggestions provided by users about their experience with a product.</p> <p>User Interface (UI): Visual and functional elements that users interact with directly in a system.</p> <p>User Experience (UX): The perception and emotions of a</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>Finally, implementing automation reduces costs and increases productivity.</p> <p>Thus, businesses should invest in training employees on automation tools.</p> <p>In short, automation simplifies office tasks and improves efficiency.</p> <p>In conclusion, adopting technology is key to staying competitive.</p> <p>In brief, automation helps businesses save time and resources.</p> <p>As a result, companies that use automation gain a strategic advantage.</p>	<p>personal experiences or narrating events that did not happen as originally expected.</p> <p>References you may use:</p> <p>Future - Will vs. Going to - Woodward English Grammar https://www.grammar.cl/Notes/Future_Will_vs_Going.htm</p> <p>'Will' and 'Going To' Futures: For Making Predictions - Central School https://centralschool.ie/english-grammar/will-and-going-to-predictions/</p> <p>'Will' vs 'Be Going To' – Future Forms - Test-English</p>	<p>user when interacting with a product, system, or service.</p> <p>Visual Feedback: Immediate indicators, such as color changes or pop-up messages, that confirm a user's actions.</p> <p>Wireframe: Basic, schematic representation of the layout of elements on a page or screen.</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>Accordingly, organizations should integrate automation into their daily operations.</p> <p>Idioms and common colloquial expressions</p> <p>"Hit the nail on the head" – To describe something that is exactly right, often used when discussing a design decision or user feedback that perfectly addresses the issue.</p> <p>"The new layout really hits the nail on the head; it's much easier to navigate."</p> <p>"Back to the drawing board" – To start over with a new idea or</p>	<p>https://test-english.com/explanation/a2/will-vs-be-going-to/</p> <p>Will vs Going To: Difference & Explanation Promova</p> <p>https://promova.com/english-grammar/will-vs-going-to</p> <p>Going to Future - Wikipedia</p> <p>https://en.wikipedia.org/wiki/Going-to_future</p>	



Functions and Discourse Markers	Grammar	Vocabulary
<p>approach after something doesn't work as planned.</p> <p>"The prototype didn't pass the usability test, so it's back to the drawing board for a better design."</p> <p>"Put yourself in someone's shoes" – To consider a situation from another person's perspective, important when thinking about the user experience.</p> <p>"To improve this app, we need to put ourselves in the user's shoes and think about how they would interact with it."</p>		



Functions and Discourse Markers	Grammar	Vocabulary
<p>"On the same page" – To ensure that everyone involved in the project has the same understanding or agreement, especially when discussing design goals.</p> <p>"Before we begin, let's make sure we're all on the same page regarding the app's user flow."</p> <p>"Bite the bullet" – To accept or face something difficult or challenging, such as a design problem or user feedback that needs to be addressed.</p>		



Functions and Discourse Markers	Grammar	Vocabulary
<p>"We've received a lot of negative feedback, but we need to bite the bullet and make those changes."</p> <p>"Out of the box" – Refers to creative, innovative ideas or solutions that are not conventional.</p> <p>"We need an out-of-the-box solution to improve the app's accessibility for visually impaired users."</p> <p>"In the driver's seat" – To be in control or to make the important decisions.</p>		



Functions and Discourse Markers	Grammar	Vocabulary
<p>"With the latest updates, the UX team is in the driver's seat for determining the app's interface design."</p> <p>"Drop the ball" – To make a mistake or fail to do something important, often used when discussing missed opportunities or oversights in the design process.</p> <p>"We dropped the ball on user testing, and now we need to fix the issues before launch."</p>		

Curricular Design

Subject Area: English Oriented to Software Applications Development



Grade: Twelfth

CEFR : B1.2

Scenario 2: User Experience

Theme 2: Interaction Design

Time: 20 hours

Essential Question: How can creativity improve the way users interact with digital products in interaction design?

Essential Competences: Creativity

New Citizenship Axis: Sustainable Development Education

Table 24

Curriculum Pedagogical Design

Goals	Performance Indicator
The learners can...	The student...
Demonstrate creativity by designing innovative and user-friendly interaction models, and exploring new	Describe and present their creative interaction models, using appropriate language to explain how



Goals	Performance Indicator
The learners can...	The student...
ways to enhance user experience and engagement through unique interface elements and design features.	innovative interface elements and design features improve user engagement and enhance the user experience.
Incorporate sustainable development principles into their interaction design projects, creating user interfaces that promote environmental awareness, social responsibility, and equitable access for all users.	Explains how they incorporate sustainable development principles into their interaction design projects, using clear language to describe how their user interfaces promote environmental awareness, social responsibility, and equitable access for all users.

TABLE 25

Oral and Written Comprehension

Goals	Performance Indicator
The learners can...	The student...
Listening: • Listen to and understand explanations about interaction design principles in programming,	• Listens to explanations about interaction design principles, identify and summarize key ideas related to user interfaces, user experience, and design patterns, and articulate these concepts using appropriate language.



identifying key ideas related to user interfaces, user experience, and design patterns. <ul style="list-style-type: none">• Actively listen to feedback and discussions on interaction design concepts, summarizing key points and asking clarifying questions to deepen their understanding and improve their design skills.	<ul style="list-style-type: none">• Actively listens to feedback and discussions on interaction design, summarizing key points and asking relevant questions to clarify concepts, demonstrating their understanding and willingness to improve their design skills.
Reading: Interpret Read and comprehend texts about interaction design in programming, identifying key concepts	Reads and understands texts about interaction design in programming, identifying key concepts such as usability, user flows, and interface design, and explaining how they apply these principles to enhance their design projects.



such as usability, user flows, and interface design, and applying these principles to improve their projects.

Table 26

Oral and Written Production

Goals		Performance Indicator
The learners can...	The student...	
Spoken Interaction: <ul style="list-style-type: none">Engage in discussions about interaction design in programming, sharing ideas on user interface best practices and responding appropriately to peers' feedback and	<ul style="list-style-type: none">Engages in discussions about interaction design in programming, shares ideas on user interface best practices, and responds appropriately to peers' feedback and suggestions, using clear and relevant language.	



suggestions.

- Collaborate in groups to brainstorm and problem-solve interaction design challenges, explaining their ideas clearly and providing constructive input to enhance design solutions.

- Participates in groups to brainstorm and problem-solve interaction design challenges, explaining their ideas clearly and providing constructive feedback to improve design solutions, using appropriate language and terminology.

Spoken Production:

- Explain the key principles of interaction design in programming, describing how they apply these principles to create intuitive user interfaces and enhance user engagement.

Presents the key principles of interaction design in programming, describing how they apply these principles to create intuitive user interfaces and improve user engagement, using clear and appropriate language.



<ul style="list-style-type: none">• Present their interaction design projects, discussing design challenges, solutions, and the impact of their choices on the user experience, using clear and relevant terminology.	Describes different interaction design projects, discusses design challenges, solutions, and the impact of their choices on the user experience, using clear and relevant terminology to convey their ideas effectively.
<p>Writing: Write clear and structured explanations of interaction design principles in programming, detailing their design choices, user interface considerations, and how they enhance the overall user experience in their</p>	Writes clear and structured explanations of interaction design principles in programming, detailing their design choices, user interface considerations, and how these elements enhance the overall user experience in their projects, using appropriate language and organization.



projects.

Table 27

Learnings of the curriculum pedagogical design

Functions and Discourse Markers	Grammar	Vocabulary
Functions Explain how interaction design focuses on understanding user needs, behaviors, and goals, and how this information guides the design of intuitive and user-friendly interfaces. Demonstrate the ability to articulate the	1. Future Continuous Purpose: The Future Continuous tense is used to describe actions or events that will be in progress at a specific time in the future. It emphasizes the duration or the ongoing nature of an action at a particular moment in the future. Structure:	Interaction Design (IxD) is a field of design that focuses on creating meaningful and effective interactions between users and digital systems, such as websites, apps, or other technological devices. It involves designing the behavior and structure of systems to ensure that users can easily and intuitively achieve their goals. Key aspects of Interaction Design include: User-Centered Design:



Functions and Discourse Markers	Grammar	Vocabulary
<p>process of creating design prototypes, how they test these prototypes with users, and how they analyze feedback to improve the design based on user input.</p> <p>Explain how designs can be adapted to different screen sizes and devices, while also addressing the importance of ensuring accessibility for users</p>	<p>Affirmative: Subject + will + be + verb-ing</p> <p>Negative: Subject + will + not + be + verb-ing</p> <p>Question: Will + subject + be + verb-ing?</p> <p>Examples:</p> <p>Affirmative:</p> <p>At 8 p.m. tomorrow, I will be working on my project.</p> <p>They will be traveling to Paris next week.</p> <p>Negative:</p> <p>I won't be studying at 7 p.m. tomorrow.</p> <p>She won't be attending the meeting tomorrow morning.</p> <p>Question:</p>	<p>Understanding users' needs, behaviors, and pain points to create solutions that are practical and easy to use.</p> <p>Usability: Ensuring that users can interact with a system in a way that is simple, efficient, and enjoyable.</p> <p>Feedback: Providing users with clear responses to their actions, such as button presses or completing tasks, to guide their experience.</p> <p>Consistency: Creating predictable patterns in interactions to help users understand how to use the system with minimal effort.</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>with various physical or cognitive needs.</p> <p>Discourse Markers</p> <p>Describing hopes and plans</p> <p>Making predictions</p> <p>Giving presentations</p> <p>Connecting words</p> <p>Examples</p> <p>For example, for instance, to illustrate, in other words, as an illustration, in particular.</p> <p>Connecting words</p> <p>Details</p>	<p>Will you be joining us for dinner tonight?</p> <p>Will he be staying at the hotel for long?</p> <p>2. Future Perfect</p> <p>Purpose:</p> <p>The Future Perfect tense is used to describe actions that will be completed before a specific point in the future. It focuses on the completion of an action or event.</p> <p>Structure:</p> <p>Affirmative: Subject + will + have + past participle</p> <p>Negative: Subject + will + not + have + past participle</p>	<p>Prototyping: Creating models or mockups to test and iterate designs, ensuring that the final product meets the users' needs.</p> <p>Visual Design: Crafting interfaces that are not only functional but also aesthetically pleasing, ensuring that design enhances the user experience.</p> <p>User Flow: Planning how users move through a system, ensuring they can accomplish their tasks in a logical and efficient way.</p> <p>In short, Interaction Design is all about making sure that users can interact with</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>Specifically, especially, in particular, to explain, to list, to enumerate, in detail, namely, including.</p> <p>Emphasis</p> <p>Above all, indeed, truly, of course, certainly, surely, in fact, really, in truth,</p> <p>Connecting words</p> <p>Summary</p> <p>Finally, thus, in short, in conclusion, in brief, as a result, accordingly.</p> <p>Describing Hopes and Plans</p>	<p>Question: Will + subject + have + past participle?</p> <p>Examples:</p> <p>Affirmative:</p> <p>By next year, I will have graduated from university.</p> <p>They will have completed the project by Friday.</p> <p>Negative:</p> <p>I won't have finished my work by the time you arrive.</p> <p>She won't have visited that museum before the end of the week.</p> <p>Question:</p>	<p>digital products in ways that feel intuitive, satisfying, and effective. It combines aspects of psychology, design, and technology to create seamless and user-friendly digital experiences.</p> <p>Related vocabulary</p> <p>Accessibility: Inclusive design ensuring that people with disabilities can interact with the system.</p> <p>Adaptative Design:</p> <p>Creating interfaces that adjust to different devices and resolutions to provide a consistent experience.</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>I hope to complete my certification in office automation by the end of this year.</p> <p>Our company plans to introduce a new automated payroll system next month.</p> <p>We aim to train all employees on the latest office software within six months.</p> <p>I wish to improve my skills in Excel and database management for better efficiency.</p>	<p>Will you have read the book by tomorrow?</p> <p>Will they have repaired the car by next weekend?</p> <p>3. Future Perfect Continuous</p> <p>Purpose:</p> <p>The Future Perfect Continuous tense is used to describe an action that will be ongoing until a specific time in the future. It focuses on the duration of the action before a certain point in the future.</p> <p>Structure:</p> <p>Affirmative: Subject + will + have + been + verb-ing</p>	<p>Anthropometry Digital: Considering human physical dimensions and capabilities when designing interactions.</p> <p>Animation: Visual effects that indicate state changes, transitions, or interactions within a system.</p> <p>Assistive Interactions: Use of technologies like chatbots or virtual assistants to enhance user-system interaction.</p> <p>Cognitive Load: The mental effort required to</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>The school intends to integrate automation tools into the curriculum soon.</p> <p>Making Predictions</p> <p>I predict that office automation will replace most manual data entry tasks within five years. It is likely that businesses will adopt AI-driven tools for office management. In the future, employees will rely more on cloud-based collaboration tools.</p>	<p>Negative: Subject + will + not + have + been + verb-ing</p> <p>Question: Will + subject + have + been + verb-ing?</p> <p>Examples:</p> <p>Affirmative:</p> <p>By next month, I will have been working at this company for five years.</p> <p>At 10 a.m. tomorrow, they will have been traveling for 12 hours.</p> <p>Negative:</p> <p>I won't have been studying for long by the time you call me.</p>	<p>interact with a system; design should minimize this load.</p> <p>Consistency: Maintaining similar patterns and behaviors across the interface to facilitate user learning.</p> <p>Control of the User: Allowing users to make decisions and reverse actions in the system.</p> <p>Design Based on Tasks: A design approach that prioritizes the main tasks users need to complete.</p> <p>Design Emotional:</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>Experts foresee a rise in remote work due to automation and digital communication. Chances are that traditional filing systems will become obsolete.</p> <p>Giving Presentations</p> <p>Let me start by saying that office automation has transformed the workplace.</p> <p>Now, let's move on to the benefits of using automation tools.</p> <p>As you can see, automation reduces</p>	<p>She won't have been working here for more than a year by next summer.</p> <p>Question:</p> <p>Will you have been reading the book for an hour by the time I get home?</p> <p>Will they have been waiting for a long time when the train arrives?</p> <p>Summary:</p> <p>Future Continuous: Focuses on actions in progress at a specific future time.</p> <p>Future Perfect: Focuses on actions completed before a specific future time.</p>	<p>Incorporating elements that evoke positive emotions in users when interacting with the system.</p> <p>Design Progressively:</p> <p>Showing only the necessary elements at each moment, revealing advanced or additional options as needed.</p> <p>Design Patterns: Reusable solutions for common problems in interaction design (e.g., drop-down menus, search bars).</p> <p>Feedback: Immediate response from the system</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>errors and increases productivity.</p> <p>To summarize, implementing automation leads to time and cost savings. I'd like to conclude by emphasizing the importance of adapting to new technologies.</p> <p>Connecting Words -</p> <p>Examples</p> <p>For example, many companies use CRM software to manage</p>	<p>Future Perfect Continuous: Focuses on the duration of an action that will continue up to a specific point in the future.</p> <p>References you may use:</p> <p>British Council - Learn English Grammar: Future Continuous and Future Perfect</p> <p>https://learnenglish.britishcouncil.org/grammar/b1-b2-grammar/future-continuous-future-perfect</p> <p>Grammarly Blog - Future Perfect Continuous Tense</p> <p>https://www.grammarly.com/blog/grammar/future-perfect-continuous-tense/</p> <p>Oxford English Grammar - Future Continuous and Future Perfect</p>	<p>to the user's actions, such as confirmation messages or visual changes.</p> <p>First Mobile (Mobile-First): Strategy prioritizing mobile design before adapting it to larger screens.</p> <p>Flow of Interaction: The sequence of steps a user follows when interacting with a system to achieve a goal.</p> <p>Heat Maps: Visual representations showing the most interacted areas of an interface.</p> <p>Heuristics of Nielsen: Basic</p>



Functions and Discourse Markers	Grammar	Vocabulary
customer data efficiently. For instance, Microsoft Teams allows remote teams to collaborate seamlessly. To illustrate, consider a small business that automates invoicing, reducing human error. Spreadsheets help in organizing financial records; in other words, they simplify accounting. Cloud storage services, as an illustration,	https://www.oxfordlearnersdictionaries.com/grammar/english-grammar-reference/future-continuous-and-future-perfect English Page - Future Continuous Tense https://www.englishpage.com/verbpage/futurecontinuous.html English Club - Future Perfect and Future Perfect Continuous https://www.englishclub.com/grammar/verb-tenses-future-perfect-continuous.htm Using English - Future Continuous, Perfect, and Perfect Continuous Tenses	principles for evaluating and improving usability in interaction design. Interaction Assisted: Using technologies like chatbots or virtual assistants to enhance user-system interaction. Interaction Model: The framework for how users interact with a system or product. Interaction Flow: The sequence of actions a user takes to achieve a goal in a system. Interaction User-System:



Functions and Discourse Markers	Grammar	Vocabulary
<p>provide secure access to files from anywhere. Employees must develop digital skills; in particular, proficiency in office applications. Connecting Words - Details</p> <p>This course covers office automation tools, specifically Word, Excel, and Outlook. Automation is beneficial for all businesses, especially those handling large amounts of data.</p>	<p>https://www.usingenglish.com/articles/future-continuous-perfect-and-perfect-continuous.html</p> <p>Cambridge Dictionary - Grammar: Future Continuous, Perfect, and Perfect Continuous</p> <p>https://dictionary.cambridge.org/grammar/british-grammar/future-continuous-future-perfect</p> <p>Perfect English Grammar - Future Perfect and Future Continuous</p> <p>https://www.perfect-english-grammar.com/future-perfect-continuous.html</p>	<p>The process in which users communicate with a system to complete specific tasks.</p> <p>Interactive Prototype: A functional representation of a design that allows users to interact and experience workflows.</p> <p>Interactive Elements: Visual and functional components that allow interaction, such as buttons, menus, and forms.</p> <p>Mental Model: The user's perception and expectations of how a</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>Security measures, in particular, data encryption and backup, are crucial in automation.</p> <p>To explain, automation reduces repetitive tasks, allowing employees to focus on strategic work. Let's list the key features of office automation: email management, cloud storage, and workflow automation. Office automation includes various tools, namely, document</p>		<p>system should function.</p> <p>Microinteractions: Small interactions within a system that provide feedback, such as animations when pressing a button.</p> <p>Mobile-First Principle: Strategy that prioritizes designing for mobile devices before adapting to larger screens.</p> <p>Navigation Map: A diagram showing how different screens or pages of a system are connected.</p> <p>Navigation Indicators:</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>management systems, scheduling software, and virtual assistants. Employees should learn to use automation tools, including AI-powered chatbots.</p> <p>Emphasis</p> <p>Above all, automation improves efficiency and reduces workload. Indeed, businesses that embrace automation see significant productivity gains.</p>		<p>Visual elements such as breadcrumbs or progress bars that help users understand where they are and how to proceed.</p> <p>Persona: A fictitious representation of a typical user that helps guide interaction design decisions.</p> <p>Prevention of Errors: Design that minimizes the chances of user errors, such as real-time data validation.</p> <p>Progressive Disclosure: Revealing only necessary elements at each moment</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>Using the latest software is essential, truly making work easier.</p> <p>Office automation is necessary, of course, for a competitive business environment.</p> <p>Certainly, digital tools streamline office operations.</p> <p>Surely, automated workflows can save employees a lot of time.</p> <p>In fact, many companies have already replaced</p>		<p>and providing advanced options as needed.</p> <p>Prototype Fidelity: The level of detail and functionality of a prototype, ranging from low fidelity (basic sketches) to high fidelity (complete interactions).</p> <p>Real-Time Interactions: Instant updates and responses in the interface, common in chat systems or monitoring tools.</p> <p>System States: Different conditions an interface can display based on the context, such as loading,</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>manual processes with automation.</p> <p>Really, office automation is transforming the way businesses operate.</p> <p>In truth, technology is shaping the future of work.</p> <p>Connecting Words - Summary</p> <p>Finally, implementing automation reduces costs and increases productivity.</p> <p>Thus, businesses should invest in training</p>		<p>error, or success states.</p> <p>Simplicity: A design that eliminates unnecessary elements to make the interface easier to understand and use.</p> <p>Task-Based Design: A design approach that focuses on the main tasks users need to perform.</p> <p>Touch Zones: Areas of the interface optimized for touch interactions, considering finger size and reach.</p> <p>Usability Test: Evaluations with real users to identify</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>employees on automation tools.</p> <p>In short, automation simplifies office tasks and improves efficiency.</p> <p>In conclusion, adopting technology is key to staying competitive.</p> <p>In brief, automation helps businesses save time and resources.</p> <p>As a result, companies that use automation gain a strategic advantage.</p>		<p>interaction flow issues and propose improvements.</p> <p>Wireframe: The initial blueprint of an interface showing the basic layout of elements, without visual or interaction details.</p> <p>Zones Tactiles: Interface areas optimized for touch interactions, considering finger size and reach.</p>



Functions and Discourse Markers	Grammar	Vocabulary
<p>Accordingly, organizations should integrate automation into their daily operations.</p> <p>Idioms and common colloquial expressions Get in touch" : "Hit the nail on the head" – To describe a design or solution that is exactly right or perfectly aligned with user needs.</p>		



Functions and Discourse Markers	Grammar	Vocabulary
<p>"Back to the drawing board" – Refers to the need to start over with a new design approach after realizing the current one doesn't work as expected.</p> <p>"Break the ice" – To ease into a conversation or situation, particularly when discussing new ideas or brainstorming design concepts with users or team members.</p>		



Functions and Discourse Markers	Grammar	Vocabulary
<p>"On the same page" – Refers to being aligned with team members on design goals, user needs, and project objectives.</p> <p>"Put yourself in the user's shoes" – A common phrase used to emphasize the importance of empathizing with the user's perspective when designing interfaces.</p>		



Functions and Discourse Markers	Grammar	Vocabulary
<p>"Smooth sailing" – Refers to a phase of the design process where things are going well, and there are no major issues or obstacles.</p> <p>"Think outside the box" – Encourages creativity in designing unique, unconventional solutions to user experience challenges.</p> <p>"Dot the i's and cross the t's" – Refers to paying attention to</p>		



Functions and Discourse Markers	Grammar	Vocabulary
every detail in the design process, ensuring no element of the user interface is overlooked.		

References

Class Design and Exception Models

University of Michigan. (n.d.). Basic Class Design. Retrieved from <https://websites.umich.edu/~eecs381/handouts/BasicClassDesign.pdf>



ArjanCodes. (2023, February 19). Class Design Best Practices in OOP Explained. Retrieved from <https://arjancodes.com/blog/best-practices-for-class-design-in-object-oriented-programming/>

Microsoft. (2021, September 15). Exception Handling in C#. Retrieved from <https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/exceptions/exception-handling>

Baeldung. (2023, May 10). Exception Handling in Java. Retrieved from <https://www.baeldung.com/java-exceptions>

Kirkwood Community College. (2017). Exception Handling – Programming Fundamentals. Retrieved from <https://kirkwood.pressbooks.pub/programmingfundamentals/chapter/exception-handling/>

Mobile Applications

4. IBM. (n.d.). What is mobile application development? Retrieved from <https://www.ibm.com/think/topics/mobile-application-development>
5. Microsoft Azure. (n.d.). What is mobile app development? Retrieved from <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-mobile-app-development>
6. BuildFire. (2024, September 15). A complete guide to mobile app development. Retrieved from <https://buildfire.com/understanding-mobile-app-development-lifecycle/>



7. OutSystems. (n.d.). Mobile app development: A complete guide. Retrieved from <https://www.outsystems.com/application-development/mobile-apps-overview/mobile-app-development-fundamentals/>
8. Upwork. (2024, July 10). What is mobile application development? A basic guide. Retrieved from <https://www.upwork.com/resources/mobile-application-development-to-networking-terminology-interfaces-and-protocols>. DigitalOcean. Retrieved from <https://www.digitalocean.com/community/tutorials/an-introduction-to-networking-terminology-interfaces-and-protocols>
9. Clutch.co. (n.d.). App Development Glossary: 64 Essential Terms. Retrieved from <https://clutch.co/resources/app-development-glossary-64-essential-terms>
10. WillowTree Apps. (n.d.). A Glossary of Mobile App and Web Terminology. Retrieved from <https://www.willowtreeapps.com/craft/a-glossary-of-mobile-app-and-web-terminology>
11. Webiotic. (n.d.). Mobile App Development Terminology. Retrieved from <https://www.webiotic.com/mobile-app-development-terminology-you-need-to-know/>
12. Flipabit. (n.d.). Mobile App Glossary. Retrieved from <https://flipabit.dev/glossary/>
13. Builder.ai. (n.d.). Mobile App Terminology - App Glossary. Retrieved from <https://www.builder.ai/glossary/app-terms>



1. UX Design Institute. (n.d.). A complete UX glossary: 101 UX terms all designers should know. Retrieved from <https://www.uxdesigninstitute.com/blog/glossary-ux-terms/>
2. UX Playbook. (n.d.). User Experience Glossary. Retrieved from <https://uxplaybook.org/ux-glossary>
3. UXTweak. (n.d.). Glossary of UX terms. Retrieved from <https://www.uxtweak.com/ux-glossary/>
4. CareerFoundry. (n.d.). The Ultimate UX Design Glossary: 91+ Terms You Should Know. Retrieved from <https://careerfoundry.com/blog/ux-design/ux-design-glossary/>
5. Coursera. (n.d.). User Experience (UX) Terms: A to Z Glossary. Retrieved from <https://www.coursera.org/resources/user-experience-terms>
6. Nielsen Norman Group. (n.d.). UX Basics: Study Guide. Retrieved from <https://www.nngroup.com/articles/ux-basics-study-guide/>
7. Nielsen Norman Group. (1998, August 8). The Definition of User Experience (UX). Retrieved from <https://www.nngroup.com/articles/definition-user-experience/>
8. Baymard Institute. (n.d.). What Is UX? The Basics of User Experience Design. Retrieved from <https://baymard.com/learn/what-is-ux>
9. CareerFoundry. (2023, January 1). What Is UX Design? A Definition & Guide. Retrieved from <https://careerfoundry.com/en/blog/ux-design/what-is-user-experience-ux-design-everything-you-need-to-know-to-get-started>



Interaction Design

1. CareerFoundry's UX Design Glossary: https://careerfoundry.com/blog/ux-design/ux-design-glossary/?utm_source=chatgpt.com
2. UX Design Institute's Complete UX Glossary: https://www.uxdesigninstitute.com/blog/glossary-ux-terms/?utm_source=chatgpt.com
3. City Tech OpenLab's Interaction Design Glossary: https://openlab.citytech.cuny.edu/marzecmttec2120/glossary/?utm_source=chatgpt.com
4. Interaction Design Foundation's Glossary of Human-Computer Interaction: https://www.interaction-design.org/literature/book/the-glossary-of-human-computer-interaction/interaction-design-patterns?utm_source=chatgpt.com
5. Usability First's Interaction Design Glossary: https://www.usabilityfirst.com/glossary/interaction-design-term/index.html?utm_source=chatgpt.com



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%ADculo-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_eds_esp01.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.



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.



Diseño Emocional: Incorporar elementos que despierten emociones positivas en los usuarios al interactuar con el sistema.



»»» Desarrollo de Aplicaciones de Software



Nivel

Duodécimo



Juega para aprender



“Motívate a la salud digital en tu aprendizaje”