

“DESARROLLO DE APLICACIONES WEB”

Nivel de Cualificación: 3 (tres)

Competencia general	Desarrollar, implantar, y mantener aplicaciones web, con independencia del modelo empleado y utilizando tecnologías específicas, garantizando el acceso a los datos de forma segura y cumpliendo los criterios de accesibilidad, usabilidad y calidad exigidas en los estándares establecidos.
----------------------------	--

UNIDADES DE COMPETENCIA	MÓDULOS FORMATIVOS	Carga Horaria
1. Desarrollar elementos software en el entorno cliente.	1. Desarrollo de elementos software en el entorno cliente.	80
2. Desarrollar elementos software en el entorno servidor.	2. Desarrollo de elementos software en el entorno servidor.	100
3. Administrar la comunicación cliente -servidor.	3. Administración de la comunicación cliente-servidor.	70
4. Gestionar el ciclo de vida del software.	4. Gestión del ciclo de vida del software.	50
5. Realizar pruebas de calidad y automatización.	5. Realización de pruebas de calidad y automatización.	70
Carga Horaria Total del Perfil Profesional		370

Ministerio de Trabajo, Empleo y Seguridad Social (MTESS)
Sistema Nacional de Formación y Capacitación Laboral (SINAFOCAL)
Formato Estándar de Diseño Curricular de Módulos Formativos

Nombre del módulo	Desarrollo de elementos software en el entorno cliente.	Código	MF0258_3
Familia profesional	Tecnologías de la Información y Comunicación (TICS).		
Perfil profesional	Desarrollo de aplicaciones web.		
Código del perfil profesional	Tics 0108_3		
Nivel de competencia (CNPP)	3 (tres)		
Duración del módulo	80 Horas	Modalidad	Presencial
Perfil de entrada del participante	<p>Requisitos mínimos del participante para el acceso a la capacitación Cumplir uno de los siguientes requisitos:</p> <ul style="list-style-type: none"> - Educación Media, concluida - Educación Media abierta, concluida. - Educación Media para Personas Jóvenes y Adultas, concluida. - Certificación de competencias ligada a perfiles del CNPP de nivel 2 de cualificación. 		
Perfil del profesional formador	Perfil académico (competencias técnicas)	Experiencia profesional requerida	
	<ul style="list-style-type: none"> - Título de profesor / docente en la especialidad de la familia profesional correspondiente o, - Título de Técnico Superior o, - Título de grado universitario o postgrado relacionado con el módulo formativo. 	Con titulación	Sin titulación
		1 año de experiencia laboral	3 años de experiencia laboral
<p>Competencias pedagógicas</p> <ul style="list-style-type: none"> - Título de profesor / docente en cualquier especialidad o, - Habilitación en docencia en EFTP requerida por cada Administración responsable, o - Título de profesor profesionalizante o, Certificado de especialización en el área con formación metodológica o pedagógica. 			
Perfil de salida del participante	<p>Una vez concluido los "5" módulos, el participante tendrá el perfil de: Desarrollo de aplicaciones web</p> <ul style="list-style-type: none"> - Otras ocupaciones y posibles puestos de trabajo: Programador informático - Programador, programación técnica - Programador, banco de datos - Programador, informática - Verificador, programas - Otros programadores informáticos no especificados - Desarrollador Web - Desarrollador <i>Front End</i> - Desarrollador <i>Back End</i> - Técnicos en Programación Informática - Técnico, análisis informático - Programador, informática 		

Objetivo General	Desarrollar elementos software en el entorno cliente.
Resultados de aprendizaje (RA) y criterios de evaluación (CE) *	
RA1. Crear wireframes de la Interfaz de Usuario según las características específicas de desarrollo, previa maquetación de acuerdo con los requerimientos para cumplan con los estándares y expectativas del proyecto.	
<p>C.E.1.1. Identificar la estructura de la interfaz de usuario y sus elementos de diseño (botones, campos de entrada, secciones de contenido) y los elementos gráficos en el diseño (colores, tipografía, iconos, entre otros) de acuerdo a los lineamientos de diseño visual.</p> <p>C.E.1.2. Desarrollar <i>wireframes</i> que representen la estructura y el diseño básico de la interfaz web de usuario (responsivo y adaptativo), mostrando la disposición general de elementos, utilizando herramientas de <i>wireframing</i> (<i>Sketch, Figma, Adobe XD, o similares</i>) según las características específicas de desarrollo.</p> <p>C.E.1.3. Crear prototipos que integren diseños visuales para la interfaz web del usuario, seleccionando colores, tipografías, iconos y otros elementos gráficos que reflejen la identidad de la marca y expectativas del proyecto, asegurando coherencia visual y estética.</p> <p>C.E.1.4. Adaptar los <i>wireframes</i> para mostrar cómo la interfaz de usuario se comportará en diferentes tamaños de pantalla y dispositivos, asegurando que la experiencia del usuario sea consistente y efectiva en todos los contextos de uso.</p> <p>C.E.1.5. Realizar iteraciones en los <i>wireframes</i>, ajustando la disposición de los elementos según sea necesario para anticiparse y resolver cualquier problema potencial en la estructura y diseño de la interfaz web de usuario.</p> <p>C.E.1.6. Definir colores, tipografías, imágenes y otros elementos visuales utilizando maquetas que añadan detalles visuales y de diseño a los <i>wireframes</i>, siguiendo los lineamientos establecidos por la organización para asegurar coherencia y calidad en la presentación final de la interfaz web de usuario.</p> <p>C.E.1.7. Resolver de forma proactiva problemas de adaptabilidad identificadas, asegurando que la aplicación funcione perfectamente en diversos entornos y dispositivos.</p> <p>C.E.1.8. Mostrar apertura para recibir retroalimentación y está dispuesto a iterar y mejorar el diseño basado en la misma.</p>	
RA.2 Programar componentes software en el entorno cliente que cumplan con los objetivos del diseño según las especificaciones dadas.	
<p>C.E.2.1. Comprender los requisitos y funcionalidades que deben tener los componentes del software en el entorno cliente identificando las técnicas de programación y seguridad, las especificaciones de diseño (visualización, funcionalidad, comunicación, rendimiento, optimización y seguridad) según los estándares de desarrollo de software.</p> <p>C.E.2.2. Desarrollar componentes de Interfaz de Usuario (UI) y otros componentes funcionales, asegurando que cumplan con las especificaciones del diseño, según las especificaciones dadas.</p> <p>C.E.2.3. Traducir la lógica de la aplicación en código, implementando los componentes según las especificaciones proporcionadas, asegurando coherencia y precisión en la interpretación de los requisitos funcionales y técnicos.</p> <p>C.E.2.4. Aplicar técnicas de programación según los estándares de desarrollo de software definidos en el diseño para garantizar la calidad, legibilidad y mantenibilidad del código desarrollado.</p> <p>C.E.2.5. Implementar el código del componente software, asegurando que la interfaz sea usable, accesible y ergonómica, cumpliendo con las especificaciones de diseño y las normativas de usabilidad establecidas.</p> <p>C.E.2.6. Emplear herramientas de depuración y técnicas de identificación y corrección de errores de programación en los componentes desarrollados, asegurando que el código cumpla con los estándares de calidad.</p> <p>C.E.2.7. Integrar funcionalidades específicas en los documentos de desarrollo utilizando componentes software desarrollados para el entorno cliente, siguiendo las normativas y estándares aplicables para asegurar la interoperabilidad y coherencia del sistema.</p> <p>C.E.2.8. Seleccionar herramientas de control de versiones para gestionar eficazmente los cambios en el software, asegurando la trazabilidad, revisión y seguimiento de las modificaciones realizadas de acuerdo a las especificaciones dadas.</p>	
RA.3 Codificar componentes multimedia con herramientas y lenguajes específicos que permita el aumento de la funcionalidad de los elementos del entorno cliente, según especificaciones de diseño.	

C.E.3.1. Identificar las funcionalidades específicas que se desean agregar a los componentes multimedia, (reproducción controlada, interactividad con el usuario, edición en tiempo real, etc.,) así como las especificaciones de rendimiento para el despliegue.

C.E.3.2. Organizar los componentes de audio y vídeo considerando el entorno de despliegue, asegurando que se ajusten a los formatos digitales estándares y cumplan con las especificaciones de rendimiento establecidas.

C.E.3.3. Utilizar herramientas específicas para configurar elementos gráficos, ilustraciones o fotografías, asegurando que la salida sea en formato estándar y compatible con los dispositivos del entorno cliente, cumpliendo con las normativas de seguridad.

C.E.3.4. Aplicar herramientas de animación o *scripts* para configurar elementos multimedia de acuerdo con las especificaciones de diseño establecidas, asegurando que cumplan con los requisitos funcionales y visuales definidos.

C.E.3.5. Ajustar el componente desarrollado para cumplir con los criterios de accesibilidad, usabilidad y ergonomía según las normas y especificaciones dadas, respetando la propiedad intelectual y los derechos de autor aplicables.

C.E.3.6. Verificar la integración de los elementos multimedia en el entorno cliente, asegurando que cumplan con los estándares de calidad del producto según las normas de seguridad establecidas.

C.E.3.7. Implementar herramientas de control de versiones que se integren con el sistema de integración continua, facilitando la automatización de pruebas y verificaciones con cada cambio en el código, así como la calidad del software durante todo el ciclo de desarrollo.

C.E.3.8. Mantener un compromiso constante con la mejora continua de la interfaz de usuario, buscando siempre optimizar la experiencia del usuario.

RA.4 Utilizar patrones de diseño reconocidos en la industria para estructurar y optimizar el código en el desarrollo de aplicaciones web, asegurando la escalabilidad, mantenibilidad y reutilización del código.

C.E.4.1. Clasificar los patrones de diseño en el desarrollo de software (creacionales, estructurales y de comportamiento) según los requisitos del proyecto.

C.E.4.2. Seleccionar patrones de diseño después de la evaluación de las características específicas de la aplicación, asegurando la elección que mejor se adapten a las necesidades técnicas y funcionales del proyecto.

C.E.4.3. Implementar patrones de diseño, asegurando su integración adecuada en la estructura del proyecto de acuerdo con las directrices de desarrollo y las mejores prácticas establecidas.

C.E.4.4. Analizar el código existente para mejorar la calidad y eficiencia del mismo y oportunidades de refactorización mediante el uso de patrones de diseño, realizando las modificaciones necesarias.

C.E.4.5. Documentar componentes claves de la implementación de los patrones de diseño (descripción, diagramas, ejemplos de código) proporcionando información detallada para facilitar el mantenimiento continuo y la comprensión del código por parte del equipo de desarrollo.

C.E.4.6. Utilizar herramientas de control de versiones de manera responsable, asegurando un seguimiento preciso y organizado de los cambios en el código.

C.E.4.7. Demostrar un alto nivel de compromiso en la entrega de una experiencia de usuario consistente y de alta calidad a través de todas las plataformas y dispositivos.

C.E.4.8. Asegurar que el diseño visual sea de alta calidad y coherente en todas las partes de la aplicación.

C.E.4.9. Implementa validaciones y controles de integridad de manera responsable, asegurando la precisión y seguridad de los datos.

RA.5 Desarrollar aplicaciones web con soporte de base de datos que gestione los datos internos siguiendo las especificaciones del usuario.

C.E.5.1. Identificar los siguientes elementos:

- los tipos de datos (estructurados y no estructurados) necesarios para la gestión,
- los requisitos de almacenamiento (espacio, velocidad de acceso, seguridad)
- la estructura de la base de datos (tablas, relaciones, índices)
- y las operaciones de consulta y manipulación (*CRUD: Create, Read, Update, Delete*).

C.E. 5.2. Desarrollar la capa de acceso a datos (*SQL, ORM, APIS, etc.*) de la aplicación web, implementando funciones y métodos para acceder y manipular eficientemente los datos almacenados en la base de datos.

C.E. 5.3. Utilizar tecnologías y lenguajes de programación apropiados que permitan la interacción efectiva con el gestor de base de datos (*MySQL, PostgreSQL, MongoDB y otros*) y que cumplan con los requerimientos específicos.

C.E. 5.4. Integrar la capa de acceso a datos en la lógica de la aplicación web (controladores, servicios), asegurando que las funciones de gestión y acceso a datos internos sean eficientes, coherentes y compatibles con la interfaz de usuario para garantizar el correcto funcionamiento de la aplicación.

C.E. 5.5. Desarrollar las funcionalidades de gestión de datos (*CRUD: Create, Read, Update, Delete*) internos de la aplicación web, incluyendo operaciones como la creación, lectura, actualización y eliminación de registros según las especificaciones técnicas.

C.E.5.6. Implementar las técnicas de validaciones y controles de integridad de datos (*constraints, triggers, transacciones*) para asegurar la consistencia y precisión de la información almacenada en la base de datos.

C.E. 5.7. Realizar pruebas de la funcionalidad de gestión de datos (*Selenium, Appium, manual testing y otros*) de la aplicación web y de compatibilidad (*crossbrowser, cross-platform*) en diferentes dispositivos y sistemas operativos identificando y corrigiendo errores de funcionamiento por diversos motivos.

C.E. 5.8. Desplegar la aplicación web según los procedimientos (CI/CD, entornos de producción) en los dispositivos de los usuarios finales, cumpliendo los requerimientos específicos (accesibilidad, rendimiento, compatibilidad con distintos dispositivos) para una implementación exitosa y funcional de la aplicación.

C.E. 5.9. Realizar la documentación del proceso de las especificaciones utilizadas en:

- los requisitos y la estructura de la base de datos (versiones).
- de los métodos y funciones implementados.
- de la integración realizada y su funcionamiento.
- de las funcionalidades desarrolladas y los controles implementados.
- despliegue y de resolución de cualquier problema surgido según las indicaciones recibidas.

RA.6. Redactar la documentación técnica y del usuario, incluyendo manuales, guías y especificaciones, de acuerdo con los patrones, normas y procedimientos establecidos.

C.E.6.1. Realizar la documentación técnica (de código, de configuración e instalación incluido el diccionario de datos); siguiendo estándares del sector (IEEE, ISO) y especificando el proceso del desarrollo implementado.

C.E.6.2. Elaborar el manual de la aplicación según sus componentes (introducción, características generales, instrucciones específicas y otros) partiendo de lo general a lo específico que refleje las prestaciones del sistema web.

C.E.6.3. Elaborar el manual de usuario teniendo en cuenta el nivel de conocimiento y experiencia del usuario incluyendo ejemplos, capturas de pantalla, resolución de problemas comunes y preguntas frecuentes siguiendo las normativas y estándares establecidos.

C.E.6.4. Elaborar un diccionario de datos que describa las estructuras de datos utilizadas, incluyendo tablas, campos y relaciones.

C.E.6.5. Documentar los cambios de código (Git, SVN, etc) manteniendo un historial claro y accesible (control de versiones, *changelogs*) que describan los cambios realizados, incluyendo fecha, autor, descripción del cambio.

C.E.6.6. Documenta todos los cambios de código de manera precisa y completa, facilitando el mantenimiento y la colaboración en el proyecto.

Contenidos Formativos

Conceptuales:

Características del diseño la UI

- Principios de diseño de interfaces de usuario (UI).
- Función y propósito de los *wireframes* en el desarrollo web.
- Elementos básicos de la interfaz de usuario (botones, campos de entrada, secciones de contenido).
- Principios de diseño visual y estética.
- La identidad de la marca y cómo reflejarla en la interfaz de usuario.
- Elementos gráficos en el diseño: colores, tipografías, iconos y otros elementos.
- Diseño responsivo y adaptativo.
- Tamaños de pantalla: como afectan la disposición y funcionalidad de la interfaz.
- Importancia de una experiencia de usuario consistente y efectiva en todos los dispositivos.
- Problemas potenciales en la estructura y diseño de la interfaz.
- Importancia de los detalles visuales y de diseño en la creación de maquetas.
- Lineamientos establecidos por la organización para el diseño visual.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

Componentes software en el entorno cliente

- Especificaciones de diseño en términos de visualización, funcionalidad, comunicación, rendimiento, optimización y seguridad.
- Políticas de la organización en cuanto a desarrollo de componentes UI.
- Lógica de la aplicación y cómo traducirla en código.
- Especificaciones funcionales y técnicas proporcionadas.
- Coherencia y precisión en la implementación del código.
- Técnicas de programación y estándares de desarrollo de software.
- Importancia de la calidad, legibilidad y mantenibilidad del código.
- Prácticas recomendadas en el desarrollo de software.
- Especificaciones de diseño y normativas de usabilidad.
- Usabilidad, accesibilidad y ergonomía en la interfaz.
- Principios de diseño centrado en el usuario.
- Herramientas de depuración y técnicas de identificación de errores.
- Proceso de corrección de errores en el desarrollo de software.
- Normativas y estándares aplicables para el desarrollo de software.
- Interoperabilidad y coherencia del sistema.
- Proceso de integración de funcionalidades específicas en los documentos de desarrollo.
- Herramientas de control de versiones y su importancia en el desarrollo de software.
- Trazabilidad, revisión y seguimiento de modificaciones.
- Normativas y políticas establecidas por la organización.

Componentes multimedia

- Formatos digitales estándares para audio y vídeo.
- Especificaciones de rendimiento para el despliegue de componentes multimedia.
- Herramientas específicas para configurar elementos gráficos, ilustraciones o fotografías.
- Formatos estándar y la compatibilidad con dispositivos del entorno cliente.
- Normativas de seguridad de la organización.
- Herramientas de animación y scripts para configurar elementos multimedia.
- Especificaciones de diseño establecidas.
- Importancia de cumplir con los requisitos funcionales y visuales definidos.
- Criterios de accesibilidad, usabilidad y ergonomía.
- Normas y especificaciones de la organización.
- Propiedad intelectual y derechos de autor aplicables.
- Estándares de calidad del producto y normas de seguridad establecidas por la organización.
- Proceso de verificación de la calidad y seguridad de los elementos multimedia.
- Herramientas de control de versiones y sistemas de integración continua.
- Importancia de la automatización de pruebas y verificaciones.
- Ciclo de desarrollo del software y la garantía de calidad.

Patrones de diseño

- Propósito de los patrones de diseño en el desarrollo de software.
- Patrones de diseño. Clasificación de: creacionales, estructurales y de comportamiento.
- Los requisitos del proyecto y las características específicas de la aplicación antes de seleccionar patrones de diseño.
- Principios de integración de patrones de diseño en la estructura del proyecto.
- Mejores prácticas para la implementación de patrones de diseño.
- Propósito de la refactorización de código.
- Oportunidades de refactorización mediante el uso de patrones de diseño.
- Relevancias de la documentación en la implementación de patrones de diseño.
- Componentes clave de la documentación de patrones de diseño: descripción, diagramas, ejemplos de código.
- Beneficios de una documentación exhaustiva para el mantenimiento y comprensión del código.

Aplicaciones WEB con soporte a Base de datos

- Tipos de datos (estructurados, no estructurados).
- Requisitos de almacenamiento (espacio, velocidad de acceso, seguridad).
- Estructura de bases de datos (relacionales, NoSQL).
- Operaciones de consulta y manipulación (CRUD: Create, Read, Update, Delete).
- Propósito de la capa de acceso a datos.
- Tecnologías y lenguajes de programación comunes para la capa de acceso a datos (SQL, ORM, APIs).
- Principios de interacción con gestores de bases de datos (MySQL, PostgreSQL, MongoDB).
- Importancia de la integración de la capa de acceso a datos en la lógica de la aplicación web.
- Principios de cohesión entre la interfaz de usuario y la capa de datos.
- Arquitecturas comunes para integración (MVC, MVVM).
- Funcionalidades de gestión de datos (CRUD).
- Importancia de las validaciones y controles de integridad de datos.
- Técnicas de validación y control de integridad (constraints, triggers, transacciones).
- Importancia de las pruebas de funcionalidad en el desarrollo de software.
- Tipos de pruebas de compatibilidad (cross-browser, cross-platform).
- Herramientas y técnicas para pruebas de funcionalidad (Selenium, Appium, manual testing).
- Propósito del despliegue de aplicaciones web.
- Procedimientos y políticas comunes de despliegue (CI/CD, entornos de producción).
- Requerimientos típicos de los usuarios para el despliegue (accesibilidad, rendimiento).

Documentación técnica del usuario

- Importancia de la documentación técnica en el desarrollo de software.
- Tipos de documentación técnica: de código, de configuración, de instalación y diccionario de datos.
- Estándares del sector para la documentación técnica (IEEE, ISO).
- Propósito y estructura de un manual de aplicación.
- Componentes de un manual de aplicación: introducción, características generales, instrucciones específicas.
- Importancia de la claridad y precisión en los manuales de aplicación.
- Importancia de un manual de usuario adaptado al nivel de conocimiento y experiencia del usuario.
- Estructura típica de un manual de usuario.
- Normativas y estándares organizacionales para la creación de manuales de usuario.
- Importancia de la documentación de cambios de código en el desarrollo de software.
- Métodos para mantener un historial de cambios claro y accesible (control de versiones, changelogs).
- Herramientas para la gestión de cambios de código (Git, SVN).

Contenidos procedimentales

Creación de *wireframes* de la UI

- Desarrollo de *wireframes* que representen la estructura y diseño básico de la interfaz web de usuario.
- Disposición de elementos de la interfaz (botones, campos de entrada, secciones de contenido) de acuerdo con las indicaciones recibidas.
- Uso de herramientas de *wireframing* como Sketch, Figma, Adobe XD, o similares.
- Creación de prototipos que integren diseños visuales para la interfaz web del usuario.
- Selección de colores, tipografías, iconos y otros elementos gráficos que reflejen la identidad de la marca.
- Iteración y mejora continua en el diseño de interfaces.
- Ajustes en la disposición pueden mejorar la usabilidad y la experiencia del usuario.
- Asegurarse de coherencia visual y estética en el diseño de la interfaz.
- Adaptación de los *wireframes* para mostrar cómo la interfaz de usuario se comportará en diferentes tamaños de pantalla y dispositivos.
- Asegurarse que la experiencia del usuario sea consistente y efectiva en todos los contextos de uso.
- Realización de los *wireframes* en diferentes dispositivos y tamaños de pantalla.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

- Realización de iteraciones en los *wireframes*, ajustando la disposición de los elementos según sea necesario.
- Asegurarse que el código cumpla con los estándares de calidad y funcionamiento.
- Anticipación y resolución de problemas identificados en la estructura y diseño de la interfaz web de usuario.
- Incorporación del *feedback* de usuarios y *stakeholders* en el diseño.
- Definición de colores, tipografías, imágenes y otros elementos visuales.
- Adición de detalles visuales y de diseño a los *wireframes*.
- Integración de colores, tipografías, imágenes y otros elementos visuales en el diseño.
- Asegurarse de la coherencia y calidad en la presentación final de la interfaz web de usuario.

Programación de componentes software en el entorno cliente

- Desarrollo de componentes de UI y otros componentes funcionales.
- Asegurarse que los componentes desarrollados cumplan con los aspectos de visualización, funcionalidad, comunicación, rendimiento, optimización y seguridad.
- Implementación de políticas de la organización en el desarrollo de componentes UI.
- Traducción de la lógica de la aplicación en código.
- Asegurarse de la coherencia y precisión en la interpretación de los requisitos funcionales y técnicos.
- Selección de patrones de diseño.
- Implementación de patrones de diseño en la arquitectura de un proyecto.
- Verificación del código desarrollado.
- Aplicación de técnicas de programación.
- Revisión de la calidad, legibilidad y mantenibilidad del código desarrollado.
- Revisión y refactorización del código para asegurar que cumpla con los estándares.
- Implementación del código del componente software.
- Integración de elementos multimedia en el entorno cliente.
- Cumplimiento de las especificaciones de diseño y normativas de usabilidad.
- Realización de pruebas de usabilidad.
- Uso de herramientas de depuración de programación en los componentes desarrollados.
- Asegurarse que el código cumpla con los estándares de calidad y funcionamiento definidos.
- Registración de los errores identificados y las correcciones realizadas.
- Integración de funcionalidades específicas en los documentos de desarrollo.
- Asegurarse de la interoperabilidad y coherencia del sistema.
- Seguimiento de las normativas y estándares aplicables en la integración de funcionalidades.
- Selección de herramientas de control de versiones.
- Asegurarse de la trazabilidad, revisión y seguimiento de las modificaciones realizadas.
- Seguimiento de las normativas y políticas establecidas en el uso de herramientas de control de versiones.

Codificación de componentes multimedia

- Organización de los componentes de audio y vídeo.
- Optimización de audio y vídeo según el entorno de despliegue.
- Asegurarse que los componentes multimedia se ajusten a los formatos digitales estándares.
- Verificación de los componentes multimedia cumplan con las especificaciones de rendimiento establecidas.
- Utilización de herramientas específicas para configurar elementos gráficos, ilustraciones o fotografías.
- Asegurarse que la salida sea en formato estándar y compatible con los dispositivos del entorno cliente.
- Cumplimiento de las normativas de seguridad de la organización al configurar los elementos gráficos.
- Aplicación de herramientas de animación o scripts.
- Asegurarse que los elementos multimedia cumplan con las especificaciones de diseño establecidas.
- Verificación que los elementos multimedia cumplan con los requisitos funcionales y visuales.
- Integración de los patrones de diseño en la estructura del proyecto.
- Ajuste del componente desarrollado con los criterios de accesibilidad, usabilidad y ergonomía.
- Asegurarse que el componente desarrollado cumpla con las normas y especificaciones de la organización.
- Cumplimiento de las normativas de la propiedad intelectual y los derechos de autor aplicables en el desarrollo de componentes.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

- Verificación de la integración de los elementos multimedia en el entorno cliente.
- Asegurarse que los elementos multimedia cumplan con los estándares de calidad del producto.
- Seguimiento de las normas de seguridad establecidas por la organización en la integración de elementos multimedia.
- Implementación de herramientas de control de versiones que se integren con el sistema de integración continua.
- Automatización de pruebas y verificaciones con cada cambio en el código.
- Asegurarse la calidad del software durante todo el ciclo de desarrollo.

Uso de patrones de diseño reconocidos en la industria

- La implementación de patrones de diseño en la arquitectura de software.
- Análisis exhaustivo de los requisitos técnicos y funcionales del proyecto.
- Evaluación de los patrones de diseño disponibles en relación con los requisitos del proyecto.
- La refactorización para mejorar la calidad y eficiencia del código.
- Selección de los patrones de diseño más adecuados para resolver los problemas específicos del proyecto.
- Planificación de la implementación de los patrones de diseño seleccionados en la arquitectura del proyecto.
- Integración de los patrones de diseño en la estructura del proyecto siguiendo las directrices de desarrollo.
- Validación de la implementación para asegurar que cumple con las mejores prácticas establecidas.

Gestión de los datos en el desarrollo de aplicaciones WEB

- Análisis de las necesidades de la organización para identificar los tipos de datos a gestionar.
- Diseño de la estructura de la base de datos adecuada (tablas, relaciones, índices).
- Definición de las operaciones de consulta y manipulación requeridas.
- Documentación de los requisitos y la estructura de la base de datos.
- Implementación de funciones y métodos para acceder y manipular datos (consultas SQL, uso de ORM).
- Configuración de la capa de acceso a datos para interactuar con el gestor de la base de datos.
- Pruebas y validación de las funciones de acceso a datos.
- Documentación de los métodos y funciones implementados.
- Integración de la capa de acceso a datos en la lógica de la aplicación (controladores, servicios).
- Implementación de funciones de gestión de datos que interactúen con la interfaz de usuario.
- Pruebas para asegurar la cohesión y correcto funcionamiento de la integración.
- Documentación de la integración realizada y su funcionamiento.
- Desarrollo de funcionalidades CRUD en la aplicación web.
- Implementación de validaciones y controles de integridad en la capa de datos.
- Pruebas de funcionalidades para asegurar consistencia y precisión de los datos.
- Documentación de las funcionalidades desarrolladas y los controles implementados.
- Planificación y ejecución de pruebas de funcionalidad en diferentes dispositivos y sistemas operativos.
- Identificación y documentación de errores de funcionamiento.
- Corrección de errores y re-ejecución de pruebas para validar las soluciones.
- Generación de informes de pruebas detallados.
- Preparación del entorno de producción para el despliegue de la aplicación.
- Ejecución del proceso de despliegue siguiendo los procedimientos y políticas establecidos.
- Verificación del funcionamiento de la aplicación en los dispositivos de los usuarios finales.
- Documentación del proceso de despliegue y resolución de cualquier problema surgido.

Redacción de la documentación técnica y del usuario

- Redacción de documentación de código que explique la estructura, lógica y funciones implementadas.
- Creación de guías de configuración e instalación que detallen los pasos necesarios para preparar y desplegar la aplicación.
- Elaboración de un diccionario de datos que describa las estructuras de datos utilizadas, incluyendo tablas, campos y relaciones.
- Documentación del proceso de desarrollo implementado, desde la concepción hasta la finalización del proyecto.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

- Redacción de una introducción que presente la aplicación, sus objetivos y su público objetivo.
- Descripción de las características generales de la aplicación y sus beneficios.
- Elaboración de instrucciones detalladas para el uso de cada funcionalidad específica de la aplicación.
- Revisión y validación del manual para asegurar que sea comprensible y útil para los usuarios.
- Identificación del perfil de usuario objetivo y sus necesidades específicas.
- Redacción de un manual de usuario que explique claramente cómo utilizar la aplicación, incluyendo ejemplos y capturas de pantalla.
- Inclusión de secciones de resolución de problemas comunes y preguntas frecuentes.
- Revisión y validación del manual con usuarios reales para asegurar su eficacia y claridad.
- Registro de cambios de código detallados en un sistema de control de versiones.
- Creación de changelogs que describan claramente los cambios realizados, incluyendo fecha, autor y descripción del cambio.
- Uso de comentarios en el código para explicar modificaciones importantes.
- Revisión periódica del historial de cambios para asegurar su claridad y accesibilidad.

Contenidos actitudinales

- Compromiso con la entrega de una experiencia de usuario uniforme y de alta calidad.
- Proactividad en la identificación y resolución de problemas de adaptabilidad.
- Apertura a la retroalimentación y disposición para iterar y mejorar el diseño.
- Compromiso con la mejora continua de la interfaz de usuario.
- Compromiso con la calidad y coherencia en el diseño visual.
- Precisión en la implementación de patrones de diseño.
- Compromiso con la elección de patrones de diseño que mejor se adapten a las necesidades del proyecto.
- Compromiso con las mejores prácticas y directrices de desarrollo.
- Responsabilidad en el despliegue de la aplicación web.
- Meticulosidad en la implementación de operaciones CRUD.
- Responsabilidad en el uso adecuado de herramientas de control de versiones.
- Responsabilidad en la implementación de validaciones y controles de integridad.
- Compromiso con la documentación precisa y completa de todos los cambios de código.

Requisitos básicos de infraestructura, espacio y apoyo a la impartición de clases

Infraestructura:

- Aula polivalente hasta 40 estudiantes

Instalaciones:

- Suministro eléctrico con tomas de corrientes.
- Iluminación adecuada.
- Sistema de ventilación.
- Sistema de aire acondicionado (aconsejable pero no excluyente).
- Accesibilidad universal a los espacios.
- Extintores y sistemas de seguridad.
- Conexión a Internet, por cable o red wifi.

Apoyo:

- Notebook o PC (docente).
- Proyector multimedia.
- Pizarra acrílica, pincel y borrador.
- Aplicaciones informáticas de uso general instaladas en los equipos.
- Sillas (una para cada estudiante).
- Mesas de trabajo (acorde a la cantidad de participantes), para trabajo con o sin computador.
- Escritorio y silla para el docente o instructor.

Equipamientos, herramientas manuales y elementos auxiliares para el desarrollo de las clases

Local/Infraestructura:

Espacio	<i>Hasta 20 estudiantes</i>	<i>Hasta 40 estudiantes</i>
<i>Aula polivalente</i>	1,5 m ² por estudiante	1,5 m ² por estudiante
<i>Laboratorio de informática</i>	40 m ²	80 m ²

Equipamientos / Herramientas/Materiales didácticos:

Aula polivalente

- Notebook o PC (docente).
- Proyector multimedia.
- Aplicaciones informáticas de uso general instaladas en los equipos.
- Sillas (una para cada estudiante).
- Mesas de trabajo (acorde a la cantidad de participantes), para trabajo con o sin computador.
- Escritorio y silla para el docente o instructor.
- Computadores necesarios en función del número de estudiantes con conexión a Internet (puede ubicarse en el laboratorio de informática o en el aula polivalente en el caso de disponer de carro portátil)
- Aplicaciones informáticas de uso general instaladas en los equipos.
- Pizarra acrílica, pincel y borrador.
- Otros materiales: regla T, escuadra (45°), cartabón (30 y 60°), compás, escalímetro, portaminas 0.3, 0.5 y 0.7, borrador, estilógrafo punta fina 0.1, 0.3 y 0.5, plantilla de letras, plantilla de símbolos, reglas curvas, hojas tamaño A2, A1 y A0 en distintos gramajes y tipos.
- Programas de diseño asistido por computador instalados en todos los equipos.
- Normativas:
- Normas de calidad.
- Reglamento de Baja Tensión de ANDE.
- Norma Paraguaya de instalaciones de BT (NP 2028 96).
- Ley N° 5668/2016 “De verificación y control de la calidad y seguridad de los productos y servicios en las instalaciones eléctricas” y Decreto N° 9265/2018 “Por el cual se Reglamenta la Ley N° 5668/2016...”
- Documentos de dibujo técnico.

Elementos necesarios, propios de la clase impartida:

- Software de base y de aplicaciones.
- Equipo informático.
- Equipos audiovisuales.
- Internet e intranet.
- Instrumentos de comunicaciones.
- Dispositivos multimedia.
- Materiales de oficina.
- Diccionario de datos.
- Lenguajes de programación
- Entornos integrados de desarrollo.

Elaborado por: Equipo A y DF-CIMA

Ministerio de Trabajo, Empleo y Seguridad Social (MTESS)
Sistema Nacional de Formación y Capacitación Laboral (SINAFOCAL)
Formato Estándar de Diseño Curricular de Módulos Formativos

Nombre del módulo	Desarrollo de elementos software en el entorno servidor	Código	MF0259_3
Familia profesional	Tecnologías de la Información y Comunicación (TICS).		
Perfil profesional	Desarrollo de aplicaciones web.		
Código del perfil profesional	Tics 0108_3		
Nivel de competencia (CNPP)	3 (tres)		
Duración del módulo	100 horas	Modalidad	Presencial
Perfil de entrada del participante	<p>Requisitos mínimos del participante para el acceso a la capacitación Cumplir uno de los siguientes requisitos:</p> <ul style="list-style-type: none"> - Educación Media, concluida - Educación Media abierta, concluida. - Educación Media para Personas Jóvenes y Adultas, concluida. - Certificación de competencias ligada a perfiles del CNPP de nivel 2 de cualificación. 		
Perfil del profesional formador	Perfil académico (competencias técnicas)	Experiencia profesional requerida	
	<ul style="list-style-type: none"> - Título de profesor / docente en la especialidad de la familia profesional correspondiente o, - Título de Técnico Superior o, - Título de grado universitario o postgrado relacionado con el módulo formativo. 	Con titulación	Sin titulación
		1 año de experiencia laboral	3 años de experiencia laboral
<p>Competencias pedagógicas</p> <ul style="list-style-type: none"> - Título de profesor / docente en cualquier especialidad o, - Habilitación en docencia en EFTP requerida por cada Administración responsable, o - Título de profesor profesionalizante o, Certificado de especialización en el área con formación metodológica o pedagógica. 			
Perfil de salida del participante	<p>Una vez concluido los "5" módulos, el participante tendrá el perfil de: Desarrollo de aplicaciones web</p> <ul style="list-style-type: none"> - Otras ocupaciones y posibles puestos de trabajo: Programador informático - Programador, programación técnica - Programador, banco de datos - Programador, informática - Verificador, programas - Otros programadores informáticos no especificados - Desarrollador Web - Desarrollador <i>Front End</i> - Desarrollador <i>Back End</i> - Técnicos en Programación Informática - Técnico, análisis informático - Programador, informática. 		

Objetivo General	Desarrollar elementos software en el entorno servidor.
Resultados de aprendizaje (RA) y criterios de evaluación (CE) *	
RA.1. Desarrollar componentes software en el entorno servidor cumpliendo los objetivos del diseño según las especificaciones dadas.	
<p>C.E.1.1. Desarrollar el componente software utilizando lenguajes de programación (<i>Java, Python, C#, JavaScript, etc.</i>) que permitan implementar todas las funcionalidades según las especificaciones de desarrollo del proyecto.</p> <p>C.E.1.2. Codificar el componente utilizando técnicas de desarrollo estándares (TDD, BDD, CI/CD, etc) y aplicando patrones de diseño (<i>MVC, Singleton, Factory, Observer, etc</i>) que permitan la reutilización de elementos y aseguren la máxima portabilidad del código.</p> <p>C.E.1.3. Aplicar las normas de programación, utilizando las herramientas y formatos especificados para la codificación (<i>IDEs, linters, formatters, etc</i>), asegurando una documentación clara del código desarrollado.</p> <p>C.E.1.4. Definir la interfaz del componente de manera clara y la documentación según las especificaciones para su funcionamiento asegurando una integración efectiva en el sistema global del proyecto.</p> <p>C.E.1.5. Realizar pruebas al componente desarrollado en un entorno controlado y estandarizado, utilizando herramientas de depuración (<i>JUnit, Selenium, Postman, etc.</i>)- siguiendo las especificaciones de seguridad y calidad establecidas.</p> <p>C.E.1.6. Crear roles y perfiles de usuario que permitan gestionar las peticiones relacionadas con el acceso al componente software en el servidor, siguiendo las especificaciones recibidas y las necesidades específicas del proyecto.</p> <p>C.E.1.7. Configurar los componentes software del servidor a los que se accede desde el entorno cliente, asegurando que solo los usuarios con roles y perfiles tengan acceso respetando la política de seguridad vigente.</p> <p>C.E.1.8. Demostrar una actitud proactiva en la implementación de medidas de seguridad para proteger los datos del usuario, garantizando la confidencialidad, integridad y disponibilidad de la información.</p> <p>C.E.1.9. Demostrar adaptabilidad y flexibilidad ante cambios en los requisitos de documentación o en las herramientas utilizadas, ajustándose rápidamente a nuevas condiciones.</p>	
RA.2. Administrar la gestión de bases de datos en el desarrollo de aplicaciones web, garantizando la integridad, seguridad y disponibilidad de los datos, según los requisitos del proyecto.	
<p>C.E.2.1. Identificar los requisitos del proyecto revisando la documentación existente para identificar las necesidades específicas de almacenamiento y-acceso a datos que deben ser cumplidas por el proyecto.</p> <p>C.E.2.2. Seleccionar el sistema de gestión de bases de datos (DBMS) y el lenguaje <i>SQL</i> y <i>NoSQL</i> después de evaluar varias opciones, teniendo en cuenta los requisitos del proyecto, según las necesidades de (rendimiento, escalabilidad, flexibilidad, compatibilidad, etc.)</p> <p>C.E.2.3. Diseñar el esquema de la base de datos, definiendo las tablas, relaciones, claves primarias y foráneas, índices y vistas necesarios respetando el rendimiento de las consultas.</p> <p>C.E.2.4. Instalar el sistema de gestión de bases de datos (DBMS) seleccionado en los entornos de desarrollo y producción, asegurándose de que sea compatible con los sistemas existentes.</p> <p>C.E.2.5. Configurar el sistema de gestión de bases de datos (DBMS) en los entornos de desarrollo y producción, aplicando las mejores prácticas de seguridad y rendimiento para garantizar un funcionamiento óptimo y seguro.</p> <p>C.E.2.6. Implementar políticas de seguridad en la base de datos, configurando controles de acceso y permisos para garantizar que solo los usuarios autorizados puedan acceder y manipular los datos según sea necesario.</p> <p>C.E.2.7. Implementar procedimientos de respaldo y recuperación en la base de datos, estableciendo estrategias que incluyan copias de seguridad (completas, incrementales y diferenciales) para minimizar el tiempo de inactividad en caso de fallo del sistema.</p> <p>C.E.2.8. Realizar ejemplos prácticos de manejo eficientemente utilizando base de datos para un proyecto dado promoviendo la cultura de mejora continua.</p> <p>C.E.2.9. Fomenta la colaboración entre el cliente y el servidor para asegurar una comunicación eficiente y efectiva, optimizando el rendimiento de la aplicación.</p>	
RA.3. Integrar funcionalidades de desarrollo, utilizando servicios distribuidos según los estándares establecidos.	
<p>C.E.3.1. Integrar servicios web (REST, SOAP, GraphQL) en la aplicación web utilizando tecnologías estándar del mercado (HTTP, JSON, XML, etc.), asegurando que el intercambio de información entre los diferentes servicios sea transparente y eficiente.</p>	

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

C.E.3.2. Gestionar efectivamente el intercambio de información entre la aplicación web en el servidor y otros servicios web, utilizando interfaces de acceso (APIs RESTful, SOAP, WebSockets) según la tecnología requerida por el proyecto garantizando el cumplimiento de las normas de seguridad establecidas.

C.E.3.3. Seleccionar servicios frameworks y librerías estándar (Spring Boot, Express.js, Flask, etc.) que proporcionen las funcionalidades necesarias según las especificaciones del diseño, integrándolos adecuadamente en la aplicación web y asegurando la implementación conforme a las normas de seguridad establecidas.

C.E.3.4. Integrar eficazmente los elementos de diseño visual, como colores, tipografía, disposición de elementos y animaciones, asegurándose de que estén alineados con los requisitos del diseño establecidos para la aplicación web.

C.E.3.5. Mantener el diseño de la aplicación web (colores, tipografía, disposición de elementos) conforme a las especificaciones definidas, asegurando que cumpla con los requisitos de apariencia visual, experiencia de usuario y funcionalidad esperados por los usuarios.

C.E.3.6. Mostrar una actitud positiva y colaborativa en el trabajo en equipo, abordando desafíos relacionados con la comunicación cliente-servidor de manera conjunta y eficaz.

C.E.3.7. Mantener una actitud abierta a la exploración y adopción de nuevas tecnologías de comunicación cliente-servidor, evaluando y aplicando las mejores soluciones para el proyecto.

C.E.3.8. Fomentar una cultura de resolución de problemas y mejora continua, incentivando al equipo a identificar y abordar problemas de manera proactiva.

RA.4. Implementar prácticas de seguridad para proteger las aplicaciones web contra vulnerabilidades y ataques manteniendo las buenas prácticas de desarrollo.

C.E.4.1. Identificar vulnerabilidades comunes en aplicaciones web, como inyección de SQL, XSS, CSRF (Cross-Site Scripting), CSRF (Cross-Site Request Forgery), y otras, asegurando que la implementación de medidas esté alineada con las normas de seguridad establecidas por el proyecto.

C.E.4.2. Aplicar herramientas especializadas de escaneo de vulnerabilidades (OWASP ZAP, Burp Suite, Nessus, etc.) para detectar posibles puntos débiles en el código y la configuración de la aplicación web, asegurando una evaluación de la seguridad.

C.E.4.3. Implementar controles robustos de entrada de datos para prevenir ataques como inyección de SQL y XSS, asegurando que los datos ingresados por usuarios sean validados y sanitizados correctamente.

C.E.4.4. Monitorear el uso de fuentes confiables de información sobre vulnerabilidades y actualizaciones de seguridad relevantes para las tecnologías empleadas en la aplicación web, asegurando que se mantenga al día con las amenazas potenciales.

C.E.4.5. Realizar pruebas de seguridad periódicas, que pueden incluir pruebas de penetración, para identificar posibles brechas de seguridad en la aplicación web, garantizando la mitigación proactiva de riesgos.

C.E.4.6. Demostrar una actitud de aprendizaje continuo, buscando constantemente actualizarse en las mejores prácticas y herramientas disponibles en el campo.

C.E.4.7. Asumir el compromiso en la producción de la documentación que misma sea de alta calidad, asegurando que sea clara, precisa y completa.

RA.5. Emplear herramientas de control de versiones garantizando un correcto control de cambios, que facilite el desarrollo, de las pruebas y modificaciones relacionadas con nuevas funcionalidades o corrección de errores.

C.E.5.1. Seleccionar el repositorio de versionado centralizados (SVN, CVS) o distribuidos (Git, Mercurial) conforme al manual de operaciones establecido por el proyecto, asegurando que se cumplan los procedimientos y políticas para el control de cambios durante el desarrollo de la aplicación web.

C.E.5.2. Configurar la estructura organizacional interna del repositorio de versionado (*branches*, *tags*, *trunk*) según los parámetros establecidos, asegurando la integridad de los datos y facilitando un entorno para el desarrollo colaborativo de la aplicación web.

C.E.5.3. Administrar las versiones de código utilizando operaciones como *check-in*, *checkout*, *merge* y creación de *branches* según los requerimientos del proyecto, asegurando un control de acceso para los usuarios según los roles y permisos establecidos.

C.E.5.4. Gestionar efectivamente los conflictos que puedan surgir en el versionado de código y su documentación, abordando las discrepancias y asegurando que las modificaciones se integren en el repositorio de destino, manteniendo la coherencia y la continuidad del desarrollo.

C.E.5.5. Asumir un compromiso ético en la gestión de datos del usuario, asegurando que todas las prácticas y procesos cumplan con los estándares de privacidad y seguridad.

C.E.5.6. Mostrar una alta conciencia sobre la privacidad y la responsabilidad en el manejo de información sensible, implementando medidas adecuadas para protegerla.

C.E.5.7. Asumir la responsabilidad en la integración correcta de modificaciones, garantizando que todas las actualizaciones mantengan la coherencia y la integridad del desarrollo.

Contenidos Formativos

Conceptuales:

Componentes software en el entorno servidor

- Importancia de seleccionar el lenguaje de programación adecuado.
- Principales lenguajes de programación y sus aplicaciones (Java, Python, C#, JavaScript, etc.).
- Especificaciones de desarrollo de proyectos y su influencia en la elección del lenguaje.
- Definición y propósito de los patrones de diseño en el desarrollo de software.
- Técnicas de desarrollo estándares (TDD, BDD, CI/CD).
- Importancia de la reutilización y portabilidad del código.
- Normas de programación de la organización.
- Herramientas y formatos especificados para la codificación (IDEs, linters, formatters).
- Importancia de la documentación clara y adecuada del código.
- Definición y propósito de la interfaz del componente.
- Importancia de la documentación de la interfaz para la integración.
- Principios de diseño de interfaces claras y efectivas.
- Importancia de las pruebas exhaustivas en el desarrollo de software.
- Tipos de pruebas: unitarias, de integración, de sistema, de aceptación.
- Herramientas de depuración y testing (JUnit, Selenium, Postman, etc.).
- Definición de roles y perfiles de usuario en sistemas de software.
- Importancia de la gestión de accesos y permisos.
- Especificaciones y necesidades organizacionales para la gestión de accesos.
- Importancia de la configuración de componentes de servidor para la seguridad.
- Principios de configuración de acceso según roles y perfiles.
- Políticas de seguridad organizacionales y su aplicación en la configuración de accesos.

Seguridad y disponibilidad de los datos

- Definición de requisitos de almacenamiento y acceso a datos.
- Tipos de datos y su gestión en bases de datos.
- Importancia de la revisión de documentación para la identificación de necesidades específicas.
- Tipos de sistemas de gestión de bases de datos: SQL vs. NoSQL.
- Criterios para la selección de un DBMS adecuado (rendimiento, escalabilidad, flexibilidad, etc.).
- Características y ventajas de las bases de datos SQL y NoSQL.
- Principios de diseño de esquemas de bases de datos.
- Definición de tablas, relaciones, claves primarias y foráneas.
- Importancia de índices y vistas para el rendimiento de las consultas.
- Requisitos de instalación de sistemas de gestión de bases de datos.
- Importancia de la compatibilidad del DBMS con los sistemas existentes.
- Procedimientos estándar para la instalación de un DBMS.
- Mejores prácticas de configuración de DBMS para seguridad y rendimiento.
- Parámetros de configuración críticos para el rendimiento y la seguridad del DBMS.
- Importancia de la configuración adecuada para el funcionamiento óptimo del DBMS.
- Importancia de la seguridad en bases de datos.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

- Políticas de seguridad y control de acceso en DBMS.
- Técnicas para la configuración de permisos y controles de acceso.
- Importancia de los procedimientos de respaldo y recuperación en bases de datos.
- Tipos de copias de seguridad: completas, incrementales y diferenciales.
- Estrategias de respaldo y recuperación para minimizar el tiempo de inactividad.
- Importancia de la capacitación continua en administración de bases de datos.
- Técnicas y prácticas de administración de bases de datos.
- Principios de una cultura de mejora continua en tecnología.

Funcionalidades de desarrollo y servicios distribuidos

- Definición y tipos de servicios web (REST, SOAP, GraphQL).
- Importancia de la integración de servicios web en aplicaciones web.
- Tecnologías estándar para la integración de servicios web (HTTP, JSON, XML, etc.).
- Interfaces de acceso a servicios web (APIs RESTful, SOAP, WebSockets).
- Principios de seguridad en el intercambio de información entre servicios web.
- Normas de seguridad establecidas por la organización.
- Criterios para la selección de servicios web según las necesidades del proyecto.
- Proceso de integración de servicios web en aplicaciones.
- Normas de seguridad en la implementación de servicios web.
- Principios de diseño web y experiencia de usuario (UX).
- Importancia de la apariencia visual en aplicaciones web.
- Requisitos funcionales y no funcionales de una aplicación web.
- Principios de diseño visual (colores, tipografía, disposición de elementos).
- Importancia de la consistencia en el diseño de aplicaciones web.
- Técnicas para la implementación de animaciones en aplicaciones web.

Seguridad y buenas prácticas en el desarrollo de aplicaciones

- Definición y tipos de vulnerabilidades comunes en aplicaciones web: inyección de SQL, XSS (Cross-Site Scripting), CSRF (Cross-Site Request Forgery), etc.
- Principios y prácticas de seguridad en aplicaciones web.
- Normas y estándares de seguridad organizacionales y del sector (OWASP, ISO 27001, etc.).
- Importancia del escaneo de vulnerabilidades en el desarrollo y mantenimiento de aplicaciones web.
- Tipos de herramientas de escaneo de vulnerabilidades (OWASP ZAP, Burp Suite, Nessus, etc.).
- Principios de evaluación exhaustiva de seguridad en aplicaciones web.
- Importancia de los controles de entrada de datos en la seguridad de aplicaciones web.
- Técnicas de validación y sanitización de datos.
- Vulnerabilidades específicas como inyección de SQL y XSS, y cómo prevenirlas.
- Importancia de mantenerse actualizado con información sobre vulnerabilidades y seguridad.
- Fuentes confiables de información sobre seguridad (OWASP, CVE, NVD, etc.).
- Métodos para monitorear y aplicar actualizaciones de seguridad en aplicaciones web.
- Importancia de las pruebas de seguridad periódicas en aplicaciones web.
- Tipos de pruebas de seguridad: pruebas de penetración, auditorías de seguridad, etc.
- Métodos y herramientas para la realización de pruebas de seguridad.

Herramientas de versionado

- Definición y propósito de los sistemas de control de versiones.
- Tipos de sistemas de control de versiones: centralizados (SVN, CVS) y distribuidos (Git, Mercurial).
- Importancia de seleccionar el repositorio de versionado adecuado según las políticas y procedimientos organizacionales.
- Estructura organizacional de un repositorio de versionado (branches, tags, trunk).
- Principios de organización y gestión de un repositorio para mantener la integridad de los datos.
- Importancia de un entorno de desarrollo colaborativo y eficiente.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

- Operaciones básicas de administración de versiones de código: check-in, checkout, merge, creación de branches.
- Importancia del control de acceso y permisos en el repositorio de versionado.
- Principios de administración eficiente de versiones de código.
- Definición de conflictos en el versionado de código y documentación.
- Técnicas y herramientas para la resolución de conflictos (Git merge, rebase, diff tools).
- Importancia de mantener la coherencia y continuidad del desarrollo durante la resolución de conflictos.

Procedimentales:

Desarrollo componentes software en el entorno servidor

- Análisis de las especificaciones del proyecto para determinar el lenguaje de programación más adecuado.
- Implementación de funcionalidades especificadas utilizando el lenguaje seleccionado.
- Documentación y justificación de la elección del lenguaje de programación basado en las necesidades del proyecto.
- Aplicación de patrones de diseño como MVC, Singleton, Factory, Observer, etc.
- Uso de técnicas de desarrollo estándares para codificar el componente.
- Verificación del cumplimiento de las normas organizacionales en el desarrollo del componente.
- Codificación del componente siguiendo las normas y estándares de la organización.
- Uso de herramientas especificadas para garantizar la calidad del código.
- Elaboración de documentación clara y adecuada del código desarrollado.
- Definición de la interfaz del componente siguiendo principios de diseño claro y efectivo.
- Documentación detallada del funcionamiento de la interfaz.
- Validación de la interfaz con el equipo de desarrollo para asegurar una integración efectiva.
- Planificación y ejecución de pruebas exhaustivas en un entorno controlado.
- Uso de herramientas de depuración para identificar y corregir errores.
- Validación del cumplimiento de las especificaciones de seguridad y calidad.
- Creación de roles y perfiles de usuario basados en las especificaciones recibidas.
- Implementación de mecanismos de gestión de accesos y permisos.
- Documentación de roles y perfiles de usuario para referencia futura.
- Configuración de componentes del servidor siguiendo las guías y procedimientos establecidos.
- Implementación de mecanismos de autenticación y autorización basados en roles y perfiles.
- Validación y prueba de los accesos configurados para asegurar el cumplimiento de las políticas de seguridad.

Administración de la base de datos en el desarrollo de aplicaciones web

- Revisión detallada de la documentación del proyecto.
- Identificación de las necesidades específicas de almacenamiento y acceso a datos.
- Elaboración de un informe que resuma los requisitos de almacenamiento y acceso a datos.
- Evaluación de diferentes opciones de DBMS basándose en los requisitos del proyecto.
- Comparación de las características y rendimiento de bases de datos SQL y NoSQL.
- Selección y justificación del DBMS más adecuado para el proyecto.
- Diseño del esquema de la base de datos basándose en los requisitos del proyecto.
- Definición y creación de tablas, relaciones, claves primarias y foráneas.
- Implementación de índices y vistas para optimizar el rendimiento de las consultas.
- Preparación de los entornos de desarrollo y producción para la instalación del DBMS.
- Instalación del DBMS seleccionado en ambos entornos.
- Verificación de la compatibilidad y funcionamiento correcto del DBMS con los sistemas existentes.
- Configuración del DBMS en los entornos de desarrollo y producción según mejores prácticas.
- Aplicación de configuraciones de seguridad (encriptación, control de acceso, etc.).
- Optimización de parámetros de configuración para maximizar el rendimiento.
- Definición de políticas de seguridad basadas en los requisitos del proyecto.
- Configuración de controles de acceso y permisos en la base de datos.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

- Verificación y validación de la implementación de las políticas de seguridad.
- Desarrollo de procedimientos de respaldo y recuperación basados en las necesidades del proyecto.
- Implementación de estrategias de copias de seguridad (completas y diferenciales).
- Pruebas de recuperación de datos para asegurar la efectividad de los procedimientos implementados.
- Desarrollo e implementación de programas de capacitación para el equipo de desarrollo.
- Organización de talleres y seminarios sobre técnicas de administración de bases de datos.
- Evaluación y actualización regular del conocimiento del equipo sobre administración de bases de datos.

Integración de funcionalidades y servicios distribuidos

- Selección de tecnologías adecuadas para la integración de servicios web según las especificaciones del proyecto.
- Implementación de servicios web utilizando frameworks y librerías estándar (Spring Boot, Express.js, Flask, etc.).
- Pruebas de integración para asegurar el intercambio de información transparente y eficiente entre servicios.
- Implementación de interfaces de acceso adecuadas para la comunicación entre la aplicación web y otros servicios.
- Configuración de autenticación y autorización para asegurar la seguridad en el intercambio de información.
- Monitoreo y auditoría del intercambio de información para garantizar el cumplimiento de las normas de seguridad.
- Evaluación de diferentes servicios web para seleccionar los que mejor se ajusten a las especificaciones del diseño.
- Integración de los servicios seleccionados en la aplicación web.
- Implementación de medidas de seguridad durante la integración de servicios.
- Desarrollo y mantenimiento del diseño de la aplicación web según las especificaciones definidas.
- Realización de pruebas de usabilidad para asegurar una experiencia de usuario óptima.
- Validación del cumplimiento de los requisitos funcionales y de apariencia visual.
- Selección e implementación de colores, tipografía y disposición de elementos según las especificaciones del diseño.
- Integración de animaciones para mejorar la experiencia de usuario.
- Revisión y ajuste continuo de los elementos de diseño para asegurar la alineación con los requisitos establecidos.

Protección de aplicaciones web contra vulnerabilidades y ataques

- Análisis de aplicaciones web para identificar posibles vulnerabilidades.
- Implementación de medidas de seguridad alineadas con las normas establecidas por la organización.
- Documentación de las vulnerabilidades identificadas y las medidas de mitigación implementadas.
- Selección e instalación de herramientas especializadas de escaneo de vulnerabilidades.
- Configuración y ejecución de escaneos de vulnerabilidades en la aplicación web.
- Análisis de los resultados del escaneo y documentación de los puntos débiles detectados.
- Implementación de validaciones y sanitización de datos en los formularios y entradas de la aplicación web.
- Uso de bibliotecas y frameworks de seguridad para prevenir inyección de SQL y XSS.
- Pruebas de las entradas de datos para asegurar que los controles funcionan correctamente.
- Suscripción y monitoreo de fuentes confiables de información sobre seguridad.
- Evaluación y aplicación de actualizaciones de seguridad en la aplicación web.
- Documentación y comunicación de las actualizaciones de seguridad implementadas.
- Planificación y ejecución de pruebas de seguridad periódicas.
- Realización de pruebas de penetración para identificar posibles brechas de seguridad.
- Documentación y análisis de los resultados de las pruebas, y aplicación de medidas de mitigación.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

Uso de herramientas de versionado

- Análisis de los requisitos organizacionales y el manual de operaciones.
- Evaluación de diferentes sistemas de control de versiones para determinar el más adecuado.
- Selección y configuración inicial del repositorio de versionado conforme a las políticas establecidas.
- Configuración de la estructura interna del repositorio (creación de branches, tags y trunk).
- Definición de políticas de naming conventions y prácticas de trabajo en el repositorio.
- Implementación de controles de acceso y permisos según los roles de los usuarios.
- Ejecución de operaciones de administración de versiones de código (check-in, checkout, merge, creación de branches).
- Configuración y gestión de roles y permisos para asegurar el control de acceso adecuado.
- Monitoreo y documentación de las versiones y cambios en el código.
- Identificación y análisis de conflictos en el código y documentación.
- Aplicación de técnicas de resolución de conflictos para integrar modificaciones correctamente.
- Verificación y validación de la coherencia y continuidad del código después de resolver los conflictos.

Actitudinales

- Actitud proactiva en la implementación de medidas de seguridad para proteger los datos del usuario.
- Valoración de la colaboración entre el cliente y el servidor para lograr una comunicación eficiente.
- Actitud de trabajo en equipo para resolver desafíos relacionados con la comunicación cliente-servidor.
- Apertura a la exploración y adopción de nuevas tecnologías de comunicación cliente-servidor.
- Actitud de aprendizaje continuo para mantenerse actualizado en las mejores prácticas y herramientas.
- Compromiso ético en la gestión de datos del usuario durante la comunicación cliente-servidor.
- Conciencia sobre la privacidad y la responsabilidad en el manejo de información sensible.
- Compromiso constante con la calidad y la excelencia en la documentación producida.
- Adaptabilidad ante cambios en los requisitos de documentación o en las herramientas utilizadas.
- Mostrar creatividad y atención al detalle en la integración de los elementos de diseño.
- Mantener un enfoque en la calidad y la estética del diseño.
- Fomentar una cultura de resolución de problemas y mejora continua
- Compromiso con la resolución efectiva de conflictos en el código y documentación.
- Responsabilidad en la integración correcta de modificaciones para mantener la coherencia del desarrollo.
- Actitud colaborativa y proactiva para abordar discrepancias y asegurar la continuidad del proyecto.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

Requisitos básicos de infraestructura, espacio y apoyo a la impartición de clases

Infraestructura:

- Aula polivalente hasta 40 estudiantes

Instalaciones:

- Suministro eléctrico con tomas de corrientes.
- Iluminación adecuada.
- Sistema de ventilación.
- Sistema de aire acondicionado (aconsejable pero no excluyente).
- Accesibilidad universal a los espacios.
- Extintores y sistemas de seguridad.
- Conexión a Internet, por cable o red wifi.

Apoyo:

- Notebook o PC (docente).
- Proyector multimedia.
- Pizarra acrílica, pincel y borrador.
- Aplicaciones informáticas de uso general instaladas en los equipos.
- Sillas (una para cada estudiante).
- Mesas de trabajo (acorde a la cantidad de participantes), para trabajo con o sin computador.
- Escritorio y silla para el docente o instructor.

Equipamientos, herramientas manuales y elementos auxiliares para el desarrollo de las clases

Local/Infraestructura:

Espacio	Hasta 20 estudiantes	Hasta 40 estudiantes
Aula polivalente	1,5 m ² por estudiante	1,5 m ² por estudiante
Laboratorio de informática	40 m ²	80 m ²

Equipamientos / Herramientas/Materiales didácticos:

Aula polivalente

- Notebook o PC (docente).
- Proyector multimedia.
- Aplicaciones informáticas de uso general instaladas en los equipos.
- Sillas (una para cada estudiante).
- Mesas de trabajo (acorde a la cantidad de participantes), para trabajo con o sin computador.
- Escritorio y silla para el docente o instructor.
- Computadores necesarios en función del número de estudiantes con conexión a Internet (puede ubicarse en el laboratorio de informática o en el aula polivalente en el caso de disponer de carro portátil)
- Aplicaciones informáticas de uso general instaladas en los equipos.
- Pizarra acrílica, pincel y borrador.
- Otros materiales: regla T, escuadra (45°), cartabón (30 y 60°), compás, escalímetro, portaminas 0.3, 0.5 y 0.7, borrador, estilógrafo punta fina 0.1, 0.3 y 0.5, plantilla de letras, plantilla de símbolos, reglas curvas, hojas tamaño A2, A1 y A0 en distintos gramajes y tipos.
- Programas de diseño asistido por computador instalados en todos los equipos.
- Normativas:
- Normas de calidad.
- Reglamento de Baja Tensión de ANDE.
- Norma Paraguaya de instalaciones de BT (NP 2028 96).
- Ley N° 5668/2016 "De verificación y control de la calidad y seguridad de los productos y servicios en las instalaciones eléctricas" y Decreto N° 9265/2018 "Por el cual se Reglamenta la Ley N° 5668/2016..."
- Documentos de dibujo técnico.

Elementos necesarios, propios de la clase impartida:

- Software de base y de aplicaciones.
- Equipo informático.
- Equipos audiovisuales.
- Internet e intranet.
- Instrumentos de comunicaciones.
- Dispositivos multimedia.
- Materiales de oficina.
- Diccionario de datos.
- Lenguajes de programación
- Entornos integrados de desarrollo.

Elaborado por: Equipo A y DF-CIMA

Ministerio de Trabajo, Empleo y Seguridad Social (MTESS)
Sistema Nacional de Formación y Capacitación Laboral (SINAFOCAL)
Formato Estándar de Diseño Curricular de Módulos Formativos

Nombre del módulo	Administración de la comunicación cliente-servidor.	Código	MF0260_3
Familia profesional	Tecnologías de la Información y Comunicación (TICS).		
Perfil profesional	Desarrollo de aplicaciones web.		
Código del perfil profesional	TICS 0108_3		
Nivel de competencia (CNPP)	3 (tres)		
Duración del módulo	70 Horas	Modalidad	Presencial
Perfil de entrada del participante	<p>Requisitos mínimos del participante para el acceso a la capacitación Cumplir uno de los siguientes requisitos:</p> <ul style="list-style-type: none"> - Educación Media, concluida - Educación Media abierta, concluida. - Educación Media para Personas Jóvenes y Adultas, concluida. - Certificación de competencias ligada a perfiles del CNPP de nivel 2 de cualificación. 		
Perfil del profesional formador	Perfil académico (competencias técnicas)	Experiencia profesional requerida	
	<ul style="list-style-type: none"> - Título de profesor / docente en la especialidad de la familia profesional correspondiente o, - Título de Técnico Superior o, - Título de grado universitario o postgrado relacionado con el módulo formativo. 	Con titulación	Sin titulación
		1 año de experiencia laboral	3 años de experiencia laboral
<p>Competencias pedagógicas</p> <ul style="list-style-type: none"> - Título de profesor / docente en cualquier especialidad o, - Habilitación en docencia en EFTP requerida por cada Administración responsable, o - Título de profesor profesionalizante o, Certificado de especialización en el área con formación metodológica o pedagógica. 			
Perfil de salida del participante	<p>Una vez concluido los "5" módulos, el participante tendrá el perfil de: Desarrollo de aplicaciones web</p> <ul style="list-style-type: none"> - Otras ocupaciones y posibles puestos de trabajo: - Programador informático - Programador, programación técnica - Programador, banco de datos - Programador, informática - Verificador, programas - Otros programadores informáticos no especificados - Desarrollador Web - Desarrollador <i>Front End</i> - Desarrollador <i>Back End</i> - Técnicos en Programación Informática - Técnico, análisis informático - Programador, informática 		

Objetivo General	Administrar la comunicación cliente-servidor.
Resultados de aprendizaje (RA) y criterios de evaluación (CE) *	
RA.1. Configurar el servidor web o de aplicaciones siguiendo las especificaciones del proyecto, implementando protocolos de comunicación que garanticen la conectividad y la seguridad de las comunicaciones.	
<p>C.E.1.1. Instalar el servidor web o de aplicaciones (Apache, Nginx, Tomcat, etc.) utilizando las versiones de software aprobadas de acuerdo con las especificaciones detalladas en el proyecto garantizando la compatibilidad y el funcionamiento del entorno.</p> <p>C.E.1.2. Configurar el servidor web o de aplicaciones según los requisitos específicos del entorno de desarrollo y producción, asegurando que cumpla con las políticas de seguridad establecidas, incluyendo la configuración adecuada de puertos, protocolos y permisos de acceso para optimizar el rendimiento y la seguridad del servidor.</p> <p>C.E.1.3. Implementar los métodos HTTP necesarios (GET, POST, PUT, DELETE) de acuerdo con los requisitos funcionales del proyecto, asegurando que la aplicación web pueda realizar operaciones CRUD (Create, Read, Update, Delete).</p> <p>C.E.1.4. Realizar la autenticación y autorización de las APIs utilizando mecanismos como (<i>OAuth, JWT u otros</i>) según las políticas de seguridad establecidas por el proyecto, para proteger los recursos y datos sensibles de la aplicación web.</p> <p>C.E.1.5. Implementar mecanismos de seguridad para asegurar la comunicación entre clientes y servidores, configurando certificados SSL/TLS según las directrices del proyecto para proteger la integridad y confidencialidad de los datos transmitidos.</p> <p>C.E.1.6. Aplicar técnicas de caché como (<i>HTTP caching y ETag, etc</i>) para mejorar la velocidad de respuesta y reducir la latencia de datos proporcionados por el servidor, optimizando las respuestas para minimizar la carga de datos y mejorar la experiencia del usuario.</p>	
RA.2. Desarrollar los recursos requeridos por el servicio web para el cumplimiento de los objetivos propuestos en el proyecto.	
<p>C.E.2.1. Verificar que todos los elementos y operaciones del servicio web se mantengan de acuerdo con los estándares definidos y la nomenclatura establecida, garantizando consistencia y facilidad de mantenimiento en el desarrollo y evolución del servicio mediante la aplicación de buenas prácticas.</p> <p>C.E.2.2. Mantener actualizados controladores de conexión a la base de datos según los estándares y mecanismos de seguridad (encriptación, autenticación, autorización, etc.) para garantizar la protección continua de los datos y sistemas contra vulnerabilidades y amenazas, conforme a las especificaciones del proyecto.</p> <p>C.E.2.3. Implementar mecanismos eficaces de control de errores para gestionar excepciones y errores en el servicio web optimizando el uso de recursos y asegurando la disponibilidad y fiabilidad del servicio.</p> <p>C.E.2.4. Implementar medidas de seguridad robustas (encriptación, autenticación, autorización, etc.) para proteger los recursos del servicio web contra posibles amenazas y vulnerabilidades, asegurando el cumplimiento de las políticas y estándares de seguridad establecidos.</p> <p>C.E.2.5. Implementar mecanismos de paginación efectivos para gestionar grandes volúmenes de datos devueltos por el servicio web, asegurando una navegación eficiente y una respuesta rápida a las consultas de los usuarios.</p> <p>C.E.2.5. Desarrollar métodos de filtrado para permitir a los usuarios aplicar criterios específicos a las consultas realizadas al servicio web, facilitando la obtención de resultados personalizados y relevantes según las necesidades requeridas por el proyecto.</p> <p>CD.2.6. Utilizar estrategias de almacenamiento en caché que permita la reducción de carga en el servidor y aumentar la velocidad de respuesta según los requerimientos del proyecto</p>	
RA.3. Integrar los servicios WEB nativos o externos, teniendo en cuenta las medidas de seguridad y las técnicas de autenticación establecidas por el cliente.	
<p>C.E.3.1. Integrar técnicas de comunicación <i>asíncrona</i> (<i>AJAX, WebSockets, Server-Sent Events, etc.</i>) en la aplicación web para mejorar la eficiencia y la experiencia del usuario, permitiendo operaciones concurrentes y actualizaciones en tiempo real sin bloquear la interfaz de usuario.</p> <p>C.E.3.2. Implementar protocolos de seguridad y autenticación robustos (<i>HTTPS, OAuth, JWT, etc.</i>) para proteger la comunicación entre el cliente y el servidor, garantizando la confidencialidad, integridad y autenticidad de los datos transmitidos, conforme a las normas de seguridad establecidas.</p>	

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

C.E.3.3. Definir herramientas y procedimientos para monitorear la comunicación cliente-servidor, identificando y resolviendo problemas relacionados con el rendimiento y la seguridad de manera proactiva, asegurando un funcionamiento óptimo y seguro de la aplicación web.

C.E.3.4. Configurar registros de errores detallados y herramientas de seguimiento de rendimiento (Log4j, ELK Stack, Prometheus, etc.) en el servidor, facilitando la identificación y comprensión de problemas permitiendo una resolución rápida y eficiente de incidencias.

C.E.3.5. Colaborar en equipos multidisciplinarios durante el diseño y desarrollo de aplicaciones web, asegurando que las soluciones propuestas cumplan con los requisitos del cliente y las expectativas de los usuarios finales, integrando conocimientos y perspectivas diversas para optimizar el resultado final.

RA.4. Integrar APIs externas que conecte de terceros para que maneje la autenticación y autorización de acceso a recursos externos siguiendo las mejores prácticas para su implementación.

C.E.4.1. Realizar una evaluación exhaustiva de varias APIs considerando sus características, funcionalidades, documentación y reputación que cumpla con los requisitos específicos del proyecto web, asegurando que pueda integrarse de manera efectiva y eficiente en la aplicación.

C.E.4.2. Registrar en el servicio proporcionado por el gestor de la API para obtener las credenciales necesarias o tokens externos requeridos que serán utilizadas para la autenticación y autorización en el desarrollo de la aplicación web, siguiendo las políticas de seguridad establecidas.

C.E.4.3. Integrar la API externa en el código fuente de la aplicación web utilizando las bibliotecas y SDKs proporcionados por el gestor de la API-garantizando que la funcionalidad proporcionada por la API se adapte e integre a los requerimientos del proyecto.

C.E.4.4. Realizar pruebas exhaustivas de la autenticación, el manejo de errores y la autorización de accesos a recursos externos utilizando la API integrada-en diversos escenarios de uso, asegurando que responda a todas las solicitudes y situaciones previstas.

C.E.4.5. Implementar medidas de seguridad robustas conforme a las regulaciones de privacidad y protección de datos de los usuarios durante el proceso de autenticación y autorización en la API externa, garantizando el cumplimiento de las normativas establecidas y la confianza de los usuarios.

C.E. 4.6. Documentar detalladamente las instrucciones de configuración, ejemplos prácticos de uso y posibles problemas conocidos junto con sus soluciones, acerca de la integración de la API externa en la aplicación web facilitando el mantenimiento y la resolución de problemas futuros.

Contenidos Formativos

Conceptuales:

Protocolos de comunicación

- Definición y tipos de servidores web y de aplicaciones (Apache, Nginx, Tomcat, etc.).
- Importancia de seguir las especificaciones del proyecto para la instalación del servidor.
- Compatibilidad de versiones de software aprobadas por la organización.
- Principios de configuración de servidores web y de aplicaciones.
- Requisitos específicos del entorno de desarrollo y producción.
- Políticas de seguridad organizacionales relacionadas con servidores (puertos, protocolos, permisos de acceso).
- Principios de los métodos HTTP y su uso en aplicaciones web (GET, POST, PUT, DELETE).
- Importancia de las operaciones CRUD en el desarrollo de aplicaciones web.
- Buenas prácticas para la implementación de métodos HTTP de manera segura.
- Definición y diferencias entre autenticación y autorización.
- Principios de seguridad en APIs y mecanismos de autenticación y autorización (OAuth, JWT).
- Políticas de seguridad organizacionales relacionadas con la protección de APIs.
- Importancia de la seguridad en la comunicación cliente-servidor.
- Principios de SSL/TLS y su uso en la protección de datos.
- Procedimientos para la configuración de certificados SSL/TLS.
- Definición y tipos de caché en aplicaciones web (HTTP caching, ETag).
- Importancia de la caché para la mejora del rendimiento y la experiencia del usuario.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

- Principios y técnicas para la implementación de caché en aplicaciones web.

Requerimientos para el servicio web

- Importancia de los estándares y nomenclatura en el desarrollo de servicios web.
- Principios de consistencia y mantenibilidad en el desarrollo de software.
- Buenas prácticas para la nomenclatura de elementos y operaciones en servicios web.
- Importancia de la actualización regular de mecanismos de seguridad en servicios web.
- Principios de seguridad en la conexión a bases de datos.
- Políticas de seguridad organizacionales y su aplicación.
- Definición y tipos de errores y excepciones en servicios web.
- Importancia del control de errores en la disponibilidad y fiabilidad del servicio.
- Buenas prácticas para la gestión de errores y excepciones.
- Principios y medidas de seguridad en servicios web.
- Tipos de amenazas y vulnerabilidades comunes en servicios web.
- Políticas y estándares de seguridad organizacionales.
- Importancia de la paginación en la gestión de grandes volúmenes de datos.
- Principios y técnicas de paginación en servicios web.
- Impacto de la paginación en la experiencia del usuario y el rendimiento del servicio.
- Importancia del filtrado en la personalización de consultas en servicios web.
- Principios y técnicas de filtrado de datos.
- Impacto del filtrado en la experiencia del usuario y la relevancia de los resultados.

Autenticación y Autorización de acceso a recursos externos

- Definición y beneficios de la comunicación asíncrona en aplicaciones web.
- Técnicas de comunicación asíncrona (AJAX, WebSockets, Server-Sent Events, etc.).
- Impacto de la comunicación asíncrona en la eficiencia y experiencia del usuario.
- Importancia de la seguridad en la comunicación cliente-servidor.
- Protocolos de seguridad y autenticación (HTTPS, OAuth, JWT, etc.).
- Principios de confidencialidad, integridad y autenticidad de los datos.
- Importancia del monitoreo en la comunicación cliente-servidor.
- Herramientas de monitoreo de rendimiento y seguridad (New Relic, Nagios, Wireshark, etc.).
- Procedimientos para la identificación y resolución de problemas de rendimiento y seguridad.
- Importancia de los registros de errores y el seguimiento de rendimiento en servidores.
- Herramientas para el registro de errores y monitoreo de rendimiento (Log4j, ELK Stack, Prometheus, etc.).
- Buenas prácticas para la configuración y uso de registros de errores y seguimiento de rendimiento.
- Importancia de la colaboración en equipos multidisciplinarios.
- Roles y responsabilidades en un equipo multidisciplinario de desarrollo web.
- Beneficios de integrar diferentes perspectivas y conocimientos en el desarrollo de aplicaciones.

APIs externas y acceso a recursos

- Definición y tipos de APIs (REST, SOAP, GraphQL).
- Criterios para la evaluación de APIs: características, funcionalidades, documentación, reputación, soporte, etc.
- Importancia de la selección adecuada de una API para el éxito del proyecto web.
- Importancia de la autenticación y autorización en el uso de APIs.
- Tipos de credenciales y tokens (API keys, OAuth tokens, JWT).
- Políticas de seguridad relacionadas con el manejo de credenciales en la organización.
- Importancia de una integración eficiente de APIs en aplicaciones web.
- Uso de bibliotecas y SDKs para facilitar la integración de APIs.
- Principios de adaptación de la funcionalidad de la API a los requisitos del proyecto.
- Importancia de las pruebas exhaustivas en la integración de APIs.
- Métodos de prueba de autenticación, manejo de errores y autorización.
- Escenarios de uso y casos de prueba para la verificación de APIs.
- Regulaciones de privacidad y protección de datos (GDPR, CCPA, etc.).
- Principios de seguridad en la autenticación y autorización de APIs.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

- Normativas y mejores prácticas para la protección de datos de usuarios.
- Importancia de la documentación en el desarrollo de software.
- Componentes clave de una buena documentación técnica (instrucciones, ejemplos, solución de problemas).
- Buenas prácticas para la redacción de documentación de APIs.

Procedimentales:

Configuración de servidor y aplicaciones

- Definición y tipos de servidores web y de aplicaciones (Apache, Nginx, Tomcat, etc.).
- Importancia de seguir las especificaciones del proyecto para la instalación del servidor.
- Compatibilidad de versiones de software aprobadas por la organización.
- Principios de configuración de servidores web y de aplicaciones.
- Requisitos específicos del entorno de desarrollo y producción.
- Políticas de seguridad organizacionales relacionadas con servidores (puertos, protocolos, permisos de acceso).
- Implementación de los métodos HTTP necesarios en la aplicación web.
- Pruebas de las operaciones CRUD para asegurar su correcta funcionalidad.
- Documentación de las API y endpoints para facilitar su uso y mantenimiento.
- Implementación de mecanismos de autenticación y autorización en las APIs.
- Configuración y uso de OAuth, JWT u otros métodos según las necesidades del proyecto.
- Pruebas de seguridad para verificar la efectividad de la autenticación y autorización.
- Adquisición e instalación de certificados SSL/TLS.
- Configuración del servidor para utilizar SSL/TLS.
- Pruebas de comunicación segura para verificar la protección de los datos transmitidos.
- Configuración de HTTP caching y uso de ETag en la aplicación web.
- Pruebas de rendimiento para evaluar la efectividad de las técnicas de caché.
- Monitorización y ajuste continuo de las configuraciones de caché para optimizar el rendimiento.

Desarrollo de recursos para el servicio web

- Aplicación de los estándares y nomenclatura establecidos durante el desarrollo del servicio web.
- Revisión periódica del código para asegurar la adherencia a los estándares.
- Documentación de las convenciones y prácticas utilizadas en el desarrollo del servicio.
- Monitoreo de las actualizaciones y parches de seguridad relevantes.
- Actualización de los controladores de conexión a la base de datos según las políticas de seguridad.
- Pruebas de seguridad para asegurar la protección continua de los datos y sistemas.
- Implementación de mecanismos de control de errores y manejo de excepciones en el servicio web.
- Monitoreo y registro de errores para su análisis y resolución.
- Optimización del uso de recursos mediante una gestión eficiente de errores.
- Evaluación de las amenazas y vulnerabilidades potenciales en el servicio web.
- Implementación de medidas de seguridad (encriptación, autenticación, autorización, etc.).
- Pruebas de seguridad y auditorías para asegurar el cumplimiento de las políticas y estándares.
- Diseño e implementación de mecanismos de paginación en el servicio web.
- Pruebas de rendimiento para evaluar la eficiencia de la paginación.
- Optimización de los mecanismos de paginación para mejorar la respuesta y la navegación.
- Diseño e implementación de métodos de filtrado en el servicio web.
- Pruebas de funcionalidad para asegurar la precisión y relevancia de los resultados filtrados.
- Documentación de los métodos de filtrado para facilitar su uso por parte de los usuarios.

Integración se los servicios WEB

- Implementación de AJAX para realizar peticiones asíncronas sin recargar la página.
- Configuración de WebSockets para permitir la comunicación en tiempo real entre el cliente y el servidor.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

- Utilización de Server-Sent Events para actualizar automáticamente la interfaz de usuario con datos del servidor.
- Configuración de HTTPS para asegurar la comunicación entre cliente y servidor.
- Implementación de OAuth para la autenticación segura de usuarios.
- Uso de JWT para la transmisión segura de información de autenticación.
- Configuración de herramientas de monitoreo para supervisar la comunicación cliente-servidor.
- Establecimiento de procedimientos para la identificación y análisis de problemas de rendimiento y seguridad.
- Resolución proactiva de problemas detectados mediante el monitoreo.
- Configuración de herramientas de registro de errores para capturar y almacenar incidencias.
- Implementación de herramientas de seguimiento de rendimiento para monitorear el comportamiento del servidor.
- Análisis de los registros y datos de rendimiento para identificar y resolver problemas.
- Participación activa en reuniones y sesiones de trabajo en equipo.
- Integración de conocimientos y habilidades diversas en el proceso de diseño y desarrollo.
- Aseguramiento del cumplimiento de los requisitos del cliente y las expectativas de los usuarios finales.

Integración de APIs externas

- Investigación y comparación de varias APIs en base a los criterios establecidos.
- Revisión de la documentación y ejemplos de uso de cada API.
- Selección de la API que mejor se ajuste a los requisitos específicos del proyecto.
- Registro en el servicio proporcionado por el gestor de la API.
- Obtención de las credenciales necesarias (API keys, tokens) para la autenticación y autorización.
- Almacenamiento seguro y uso adecuado de las credenciales según las políticas de seguridad.
- Descarga e instalación de bibliotecas y SDKs proporcionados por el gestor de la API.
- Implementación de la API en el código fuente de la aplicación web.
- Verificación de que la funcionalidad proporcionada por la API se adapta a los requerimientos del proyecto.
- Desarrollo de casos de prueba para la autenticación, manejo de errores y autorización.
- Ejecución de pruebas exhaustivas en diversos escenarios de uso.
- Documentación y análisis de los resultados de las pruebas para asegurar el correcto funcionamiento de la API.
- Implementación de medidas de seguridad robustas para la protección de datos.
- Configuración de mecanismos de autenticación y autorización seguros (OAuth, JWT, etc.).
- Verificación del cumplimiento de las regulaciones y normativas de privacidad.
- Elaboración de una guía detallada de integración de la API externa.
- Inclusión de instrucciones de configuración, ejemplos prácticos de uso y problemas conocidos con sus soluciones.
- Revisión y actualización periódica de la documentación para asegurar su precisión y utilidad.

Conceptuales:

Protocolos de comunicación

- Definición y tipos de servidores web y de aplicaciones (Apache, Nginx, Tomcat, etc.).
- Importancia de seguir las especificaciones del proyecto para la instalación del servidor.
- Compatibilidad de versiones de software aprobadas por la organización.
- Principios de configuración de servidores web y de aplicaciones.
- Requisitos específicos del entorno de desarrollo y producción.
- Políticas de seguridad organizacionales relacionadas con servidores (puertos, protocolos, permisos de acceso).
- Principios de los métodos HTTP y su uso en aplicaciones web (GET, POST, PUT, DELETE).
- Importancia de las operaciones CRUD en el desarrollo de aplicaciones web.
- Buenas prácticas para la implementación de métodos HTTP de manera segura.
- Definición y diferencias entre autenticación y autorización.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

- Principios de seguridad en APIs y mecanismos de autenticación y autorización (OAuth, JWT).
- Políticas de seguridad organizacionales relacionadas con la protección de APIs.
- Importancia de la seguridad en la comunicación cliente-servidor.
- Principios de SSL/TLS y su uso en la protección de datos.
- Procedimientos para la configuración de certificados SSL/TLS.
- Definición y tipos de caché en aplicaciones web (HTTP caching, ETag).
- Importancia de la caché para la mejora del rendimiento y la experiencia del usuario.
- Principios y técnicas para la implementación de caché en aplicaciones web.

Requerimientos para el servicio web

- Importancia de los estándares y nomenclatura en el desarrollo de servicios web.
- Principios de consistencia y mantenibilidad en el desarrollo de software.
- Buenas prácticas para la nomenclatura de elementos y operaciones en servicios web.
- Importancia de la actualización regular de mecanismos de seguridad en servicios web.
- Principios de seguridad en la conexión a bases de datos.
- Políticas de seguridad organizacionales y su aplicación.
- Definición y tipos de errores y excepciones en servicios web.
- Importancia del control de errores en la disponibilidad y fiabilidad del servicio.
- Buenas prácticas para la gestión de errores y excepciones.
- Principios y medidas de seguridad en servicios web.
- Tipos de amenazas y vulnerabilidades comunes en servicios web.
- Políticas y estándares de seguridad organizacionales.
- Importancia de la paginación en la gestión de grandes volúmenes de datos.
- Principios y técnicas de paginación en servicios web.
- Impacto de la paginación en la experiencia del usuario y el rendimiento del servicio.
- Importancia del filtrado en la personalización de consultas en servicios web.
- Principios y técnicas de filtrado de datos.
- Impacto del filtrado en la experiencia del usuario y la relevancia de los resultados.

Autenticación y Autorización de acceso a recursos externos

- Definición y beneficios de la comunicación asíncrona en aplicaciones web.
- Técnicas de comunicación asíncrona (AJAX, WebSockets, Server-Sent Events, etc.).
- Impacto de la comunicación asíncrona en la eficiencia y experiencia del usuario.
- Importancia de la seguridad en la comunicación cliente-servidor.
- Protocolos de seguridad y autenticación (HTTPS, OAuth, JWT, etc.).
- Principios de confidencialidad, integridad y autenticidad de los datos.
- Importancia del monitoreo en la comunicación cliente-servidor.
- Herramientas de monitoreo de rendimiento y seguridad (New Relic, Nagios, Wireshark, etc.).
- Procedimientos para la identificación y resolución de problemas de rendimiento y seguridad.
- Importancia de los registros de errores y el seguimiento de rendimiento en servidores.
- Herramientas para el registro de errores y monitoreo de rendimiento (Log4j, ELK Stack, Prometheus, etc.).
- Buenas prácticas para la configuración y uso de registros de errores y seguimiento de rendimiento.
- Importancia de la colaboración en equipos multidisciplinarios.
- Roles y responsabilidades en un equipo multidisciplinario de desarrollo web.
- Beneficios de integrar diferentes perspectivas y conocimientos en el desarrollo de aplicaciones.

APIs externas y acceso a recursos

- Definición y tipos de APIs (REST, SOAP, GraphQL).
- Criterios para la evaluación de APIs: características, funcionalidades, documentación, reputación, soporte, etc.
- Importancia de la selección adecuada de una API para el éxito del proyecto web.
- Importancia de la autenticación y autorización en el uso de APIs.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

- Tipos de credenciales y tokens (API keys, OAuth tokens, JWT).
- Políticas de seguridad relacionadas con el manejo de credenciales en la organización.
- Importancia de una integración eficiente de APIs en aplicaciones web.
- Uso de bibliotecas y SDKs para facilitar la integración de APIs.
- Principios de adaptación de la funcionalidad de la API a los requisitos del proyecto.
- Importancia de las pruebas exhaustivas en la integración de APIs.
- Métodos de prueba de autenticación, manejo de errores y autorización.
- Escenarios de uso y casos de prueba para la verificación de APIs.
- Regulaciones de privacidad y protección de datos (GDPR, CCPA, etc.).
- Principios de seguridad en la autenticación y autorización de APIs.
- Normativas y mejores prácticas para la protección de datos de usuarios.
- Importancia de la documentación en el desarrollo de software.
- Componentes clave de una buena documentación técnica (instrucciones, ejemplos, solución de problemas).
- Buenas prácticas para la redacción de documentación de APIs.

Procedimentales:

Configuración de servidor y aplicaciones

- Definición y tipos de servidores web y de aplicaciones (Apache, Nginx, Tomcat, etc.).
- Importancia de seguir las especificaciones del proyecto para la instalación del servidor.
- Compatibilidad de versiones de software aprobadas por la organización.
- Principios de configuración de servidores web y de aplicaciones.
- Requisitos específicos del entorno de desarrollo y producción.
- Políticas de seguridad organizacionales relacionadas con servidores (puertos, protocolos, permisos de acceso).
- Implementación de los métodos HTTP necesarios en la aplicación web.
- Pruebas de las operaciones CRUD para asegurar su correcta funcionalidad.
- Documentación de las API y endpoints para facilitar su uso y mantenimiento.
- Implementación de mecanismos de autenticación y autorización en las APIs.
- Configuración y uso de OAuth, JWT u otros métodos según las necesidades del proyecto.
- Pruebas de seguridad para verificar la efectividad de la autenticación y autorización.
- Adquisición e instalación de certificados SSL/TLS.
- Configuración del servidor para utilizar SSL/TLS.
- Pruebas de comunicación segura para verificar la protección de los datos transmitidos.
- Configuración de HTTP caching y uso de ETag en la aplicación web.
- Pruebas de rendimiento para evaluar la efectividad de las técnicas de caché.
- Monitorización y ajuste continuo de las configuraciones de caché para optimizar el rendimiento.
-

Desarrollo de recursos para el servicio web

- Aplicación de los estándares y nomenclatura establecidos durante el desarrollo del servicio web.
- Revisión periódica del código para asegurar la adherencia a los estándares.
- Documentación de las convenciones y prácticas utilizadas en el desarrollo del servicio.
- Monitoreo de las actualizaciones y parches de seguridad relevantes.
- Actualización de los controladores de conexión a la base de datos según las políticas de seguridad.
- Pruebas de seguridad para asegurar la protección continua de los datos y sistemas.
- Implementación de mecanismos de control de errores y manejo de excepciones en el servicio web.
- Monitoreo y registro de errores para su análisis y resolución.
- Optimización del uso de recursos mediante una gestión eficiente de errores.
- Evaluación de las amenazas y vulnerabilidades potenciales en el servicio web.
- Implementación de medidas de seguridad (encriptación, autenticación, autorización, etc.).
- Pruebas de seguridad y auditorías para asegurar el cumplimiento de las políticas y estándares.
- Diseño e implementación de mecanismos de paginación en el servicio web.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

- Pruebas de rendimiento para evaluar la eficiencia de la paginación.
- Optimización de los mecanismos de paginación para mejorar la respuesta y la navegación.
- Diseño e implementación de métodos de filtrado en el servicio web.
- Pruebas de funcionalidad para asegurar la precisión y relevancia de los resultados filtrados.
- Documentación de los métodos de filtrado para facilitar su uso por parte de los usuarios.

Integración se los servicios WEB

- Implementación de AJAX para realizar peticiones asíncronas sin recargar la página.
- Configuración de WebSockets para permitir la comunicación en tiempo real entre el cliente y el servidor.
- Utilización de Server-Sent Events para actualizar automáticamente la interfaz de usuario con datos del servidor.
- Configuración de HTTPS para asegurar la comunicación entre cliente y servidor.
- Implementación de OAuth para la autenticación segura de usuarios.
- Uso de JWT para la transmisión segura de información de autenticación.
- Configuración de herramientas de monitoreo para supervisar la comunicación cliente-servidor.
- Establecimiento de procedimientos para la identificación y análisis de problemas de rendimiento y seguridad.
- Resolución proactiva de problemas detectados mediante el monitoreo.
- Configuración de herramientas de registro de errores para capturar y almacenar incidencias.
- Implementación de herramientas de seguimiento de rendimiento para monitorear el comportamiento del servidor.
- Análisis de los registros y datos de rendimiento para identificar y resolver problemas.
- Participación activa en reuniones y sesiones de trabajo en equipo.
- Integración de conocimientos y habilidades diversas en el proceso de diseño y desarrollo.
- Aseguramiento del cumplimiento de los requisitos del cliente y las expectativas de los usuarios finales.

Integración de APIs externas

- Investigación y comparación de varias APIs en base a los criterios establecidos.
- Revisión de la documentación y ejemplos de uso de cada API.
- Selección de la API que mejor se ajuste a los requisitos específicos del proyecto.
- Registro en el servicio proporcionado por el gestor de la API.
- Obtención de las credenciales necesarias (API keys, tokens) para la autenticación y autorización.
- Almacenamiento seguro y uso adecuado de las credenciales según las políticas de seguridad.
- Descarga e instalación de bibliotecas y SDKs proporcionados por el gestor de la API.
- Implementación de la API en el código fuente de la aplicación web.
- Verificación de que la funcionalidad proporcionada por la API se adapta a los requerimientos del proyecto.
- Desarrollo de casos de prueba para la autenticación, manejo de errores y autorización.
- Ejecución de pruebas exhaustivas en diversos escenarios de uso.
- Documentación y análisis de los resultados de las pruebas para asegurar el correcto funcionamiento de la API.
- Implementación de medidas de seguridad robustas para la protección de datos.
- Configuración de mecanismos de autenticación y autorización seguros (OAuth, JWT, etc.).
- Verificación del cumplimiento de las regulaciones y normativas de privacidad.
- Elaboración de una guía detallada de integración de la API externa.
- Inclusión de instrucciones de configuración, ejemplos prácticos de uso y problemas conocidos con sus soluciones.
- Revisión y actualización periódica de la documentación para asegurar su precisión y utilidad.

Actitudinales:

- Reconocimiento de la importancia de mantener la seguridad de la comunicación cliente-servidor.
- Actitud proactiva en la implementación de medidas de seguridad para proteger los datos del usuario.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

- Valoración de la colaboración entre el cliente y el servidor para lograr una comunicación eficiente.
- Actitud de trabajo en equipo para resolver desafíos relacionados con la comunicación cliente-servidor.
- Apertura a la exploración y adopción de nuevas tecnologías de comunicación cliente-servidor.
- Actitud de aprendizaje continuo para mantenerse actualizado en las mejores prácticas y herramientas.
- Compromiso ético en la gestión de datos del usuario durante la comunicación cliente-servidor.
- Conciencia sobre la privacidad y la responsabilidad en el manejo de información sensible.
- Mantener una actitud analítica y orientada a resultados.
- Mostrar responsabilidad en la implementación y mantenimiento de mecanismos de seguridad.
- Fomentar un ambiente de trabajo inclusivo y respetuoso.
- Ser diligente y detallista al crear y mantener la documentación.
- Ser proactivo en la resolución de problemas que puedan surgir durante la integración.

Requisitos básicos de infraestructura, espacio y apoyo a la impartición de clases

Infraestructura:

- Aula polivalente hasta 40 estudiantes

Instalaciones:

- Suministro eléctrico con tomas de corrientes.
- Iluminación adecuada.
- Sistema de ventilación.
- Sistema de aire acondicionado (aconsejable pero no excluyente).
- Accesibilidad universal a los espacios.
- Extintores y sistemas de seguridad.
- Conexión a Internet, por cable o red wifi.

Apoyo:

- Notebook o PC (docente).
- Proyector multimedia.
- Pizarra acrílica, pincel y borrador.
- Aplicaciones informáticas de uso general instaladas en los equipos.
- Sillas (una para cada estudiante).
- Mesas de trabajo (acorde a la cantidad de participantes), para trabajo con o sin computador.
- Escritorio y silla para el docente o instructor.

Equipamientos, herramientas manuales y elementos auxiliares para el desarrollo de las clases

Local/Infraestructura:

Espacio	Hasta 20 estudiantes	Hasta 40 estudiantes
Aula polivalente	1,5 m ² por estudiante	1,5 m ² por estudiante
Laboratorio de informática	40 m ²	80 m ²

Equipamientos / Herramientas/Materiales didácticos:

Aula polivalente

- Notebook o PC (docente).
- Proyector multimedia.
- Aplicaciones informáticas de uso general instaladas en los equipos.
- Sillas (una para cada estudiante).
- Mesas de trabajo (acorde a la cantidad de participantes), para trabajo con o sin computador.
- Escritorio y silla para el docente o instructor.
- Computadores necesarios en función del número de estudiantes con conexión a Internet (puede ubicarse en el laboratorio de informática o en el aula polivalente en el caso de disponer de carro portátil)
- Aplicaciones informáticas de uso general instaladas en los equipos.
- Pizarra acrílica, pincel y borrador.
- Otros materiales: regla T, escuadra (45°), cartabón (30 y 60°), compás, escalímetro, portaminas 0.3, 0.5 y 0.7, borrador, estilógrafo punta fina 0.1, 0.3 y 0.5, plantilla de letras, plantilla de símbolos, reglas curvas, hojas tamaño A2, A1 y A0 en distintos gramajes y tipos.
- Programas de diseño asistido por computador instalados en todos los equipos.
- Normativas:
- Normas de calidad.
- Reglamento de Baja Tensión de ANDE.
- Norma Paraguaya de instalaciones de BT (NP 2028 96).
- Ley N° 5668/2016 “De verificación y control de la calidad y seguridad de los productos y servicios en las instalaciones eléctricas” y Decreto N° 9265/2018 “Por el cual se Reglamenta la Ley N° 5668/2016...”
- Documentos de dibujo técnico.

Elementos necesarios, propios de la clase impartida:

- Software de base y de aplicaciones.
- Equipo informático.
- Equipos audiovisuales.
- Internet e intranet.
- Instrumentos de comunicaciones.
- Dispositivos multimedia.
- Materiales de oficina.
- Diccionario de datos.
- Lenguajes de programación
- Entornos integrados de desarrollo.

Elaborado por: Equipo A y DF-CIMA

Ministerio de Trabajo, Empleo y Seguridad Social (MTESS)
Sistema Nacional de Formación y Capacitación Laboral (SINAFOCAL)
Formato Estándar de Diseño Curricular de Módulos Formativos

Nombre del módulo	Gestión del ciclo de vida del software.	Código	MF0261_3
Familia profesional	Tecnologías de la Información y Comunicación (TICS).		
Perfil profesional	Desarrollo de aplicaciones web.		
Código del perfil profesional	Tics 0108_3		
Nivel de competencia (CNPP)	3 (tres)		
Duración del módulo	50 horas	Modalidad	Presencial
Perfil de entrada del participante	<p>Requisitos mínimos del participante para el acceso a la capacitación Cumplir uno de los siguientes requisitos:</p> <ul style="list-style-type: none"> - Educación Media, concluida - Educación Media abierta, concluida. - Educación Media para Personas Jóvenes y Adultas, concluida. - Certificación de competencias ligada a perfiles del CNPP de nivel 2 de cualificación. 		
Perfil del profesional formador	Perfil académico (competencias técnicas)	Experiencia profesional requerida	
	<ul style="list-style-type: none"> - Título de profesor / docente en la especialidad de la familia profesional correspondiente o, - Título de Técnico Superior o, - Título de grado universitario o postgrado relacionado con el módulo formativo. 	Con titulación	Sin titulación
		1 año de experiencia laboral	3 años de experiencia laboral
<p>Competencias pedagógicas</p> <ul style="list-style-type: none"> - Título de profesor / docente en cualquier especialidad o, - Habilitación en docencia en EFTP requerida por cada Administración responsable, o - Título de profesor profesionalizante o, Certificado de especialización en el área con formación metodológica o pedagógica. 			
Perfil de salida del participante	<p>Una vez concluido los "5" módulos, el participante tendrá el perfil de: Desarrollo de aplicaciones web</p> <ul style="list-style-type: none"> - Otras ocupaciones y posibles puestos de trabajo: - Programador informático - Programador, programación técnica - Programador, banco de datos - Programador, informática - Verificador, programas - Otros programadores informáticos no especificados - Desarrollador Web - Desarrollador <i>Front End</i> - Desarrollador <i>Back End</i> - Técnicos en Programación Informática - Técnico, análisis informático - Programador, informática 		

Objetivo General	Gestionar el ciclo de vida del software.
Resultados de aprendizaje (RA) y criterios de evaluación (CE) *	
RA.1. Planificar el proyecto de desarrollo de aplicaciones web definiendo los objetivos y alcances especificados en el proyecto.	
<p>C.E.1.1. Identificar las etapas del ciclo de vida del desarrollo de aplicaciones web, (concepción, planificación, análisis, diseño, implementación, pruebas, despliegue y mantenimiento).</p> <p>C.E.1.2. Seleccionar una metodología de desarrollo (<i>Cascada, Agile, Scrum, Kanban, entre otras</i>) que se alinee con los requerimientos del proyecto.</p> <p>C.E.1.3. Aplicar patrones de diseño apropiados para resolver problemas recurrentes en el desarrollo de aplicaciones web como <i>MVC (Model-View-Controller), Singleton, Factory, entre otros</i>, asegurando soluciones eficientes, mantenibles y escalables.</p> <p>C.E.1.4. Realizar un análisis detallado de las vulnerabilidades <i>OWASP ZAP, Burp Suite, etc.</i> potenciales y las situaciones específicas que puedan surgir en cada etapa del ciclo de vida del desarrollo de aplicaciones web.</p> <p>C.E.1.5. Implementar mejoras continuas en la optimización, rendimiento y calidad de la aplicación web a lo largo de su ciclo de vida.</p> <p>C.E.1.6. Gestionar de manera efectiva los requerimientos del proyecto con las técnicas (<i>MoSCoW, Kano, etc.</i>), recolectando, documentando y priorizando tanto los requerimientos funcionales como no funcionales de la aplicación web.</p> <p>C.E.1.7. Seleccionar un sistema de control de versiones <i>Git, SVN, Mercurial, etc.</i> basado en las necesidades específicas del proyecto, la experiencia del equipo y las mejores prácticas del sector.</p> <p>C.E.1.8. Aplicar protocolos para el control de versiones, incluyendo la creación de ramas (<i>feature branches, release branches, hotfix branches, etc.</i>) para el desarrollo paralelo,</p> <p>C.E.1.9. Realizar fusiones de ramas para integración de cambios y la resolución efectiva de conflictos.</p>	
RA.2. Realizar el plan de acción en base a los requisitos y la definición de especificaciones de los objetivos del proyecto y las necesidades del cliente.	
<p>C.E.2.1. Desarrollar especificaciones claras y detalladas para la aplicación web utilizando formatos estándar (<i>user stories, casos de uso, requisitos funcionales y no funcionales</i>) describiendo su funcionalidad, comportamiento esperado en diversos escenarios.</p> <p>C.E.2.2. Realizar una planificación detallada del tiempo y los recursos necesarios para el desarrollo de la aplicación web utilizando técnicas (<i>PERT, CPM, estimaciones ágiles, etc.</i>)</p> <p>C.E.2.3. Desarrollar guías detalladas para las pruebas de la aplicación web unitarias, de integración, de sistema, de aceptación, etc. asegurando que todos los requisitos definidos se verifiquen durante las pruebas.</p> <p>C.E.2.4. Implementar un proceso estructurado para la gestión de cambios en el proyecto de desarrollo de la aplicación web utilizando las herramientas y técnicas (<i>sistemas de seguimiento de cambios, flujos de trabajo, etc.</i>).</p> <p>C.E.2.5. Realizar en un supuesto caso práctico el plan de comunicación de:</p> <ul style="list-style-type: none"> - los ajustes según sean necesarios - los cambios realizados a todos los <i>stakeholders</i> involucrados (<i>reuniones, informes, herramientas de colaboración, etc.</i>). <p>C.E.2.7. Demuestra un reconocimiento constante de la importancia de entregar software de alta calidad, asegurando que los productos cumplen con los estándares y expectativas de los usuarios y stakeholders.</p> <p>C.E.2.8. Colaborar de manera efectiva con equipos multidisciplinarios a lo largo de todo el ciclo de vida del desarrollo, promoviendo un entorno de trabajo cohesivo y productivo.</p>	
RA.3. Aplicar metodologías de desarrollo de software teniendo en cuenta el entorno de desarrollo de aplicaciones web.	
<p>C.E.3.1. Recopilar de manera exhaustiva los requerimientos del desarrollo de la aplicación web a partir de técnicas de recolección (<i>entrevistas, encuestas, talleres de trabajo, etc</i>)</p> <p>C.E.3.2. Identificar los resultados esperados de los requerimientos asignándoles un nivel de prioridad según su importancia para el proyecto utilizando técnicas técnicas como <i>MoSCoW</i> o matrices de prioridad.</p> <p>C.E.3.3. Seleccionar las técnicas de planificación y organización realizando la agrupación de los resultados esperados de los usuarios en <i>sprints</i> de desarrollo, organizándolos según su nivel de prioridad para asegurar una implementación eficiente y enfocada en las necesidades más críticas del proyecto.</p>	

C.E.3.4. Gestionar de manera efectiva las tareas que no se incluyeron en los *sprints* planificados en la priorización y categorización de tareas en las reuniones de planificación (*Sprint Planning*).

C.E.3.5. Realizar la creación y mantenimiento de *backlog* de *sprints* y de producto que incluya todas las tareas pendientes y nuevas historias de usuario.

C.E.3.6. Utilizar métodos de retroalimentación en metodologías ágiles sobre los procesos de planificación y ejecución de cada *sprint*, con el objetivo de identificar áreas de mejora ajustando los procedimientos según los lineamientos establecidos.

C.E.3.7. Utilizar herramientas ágiles como tableros *Kanban* o herramientas específicas de seguimiento de *sprints* (*Jira*, *Trello*, *Asana*, etc.) para gestionar de manera eficiente el trabajo, asegurando la visualización clara del progreso y el cumplimiento de los tiempos establecidos para cada fase del proyecto.

C.E.3.8. Demostrar compromiso con la planificación detallada y la ejecución exitosa de proyectos web, cumpliendo con los plazos y los objetivos establecidos.

C.E.3.9. Asume responsabilidad en la implementación de medidas de seguridad y privacidad, protegiendo los datos de los usuarios y la integridad de las aplicaciones web.

RA.4. Aplicar prácticas de versionado, documentación y testeado para la codificación de software de la aplicación.

C.E.4.1. Aplicar herramientas de gestión de versiones conforme a los lineamientos del proyecto, asegurando la creación y un manejo eficiente de los *branches* en diferentes entornos como producción, desarrollo y pruebas.

C.E.4.2. Elaborar la redacción detallada de la documentación técnica con todos los requisitos esenciales para la ejecución del software, incluyendo información sobre librerías utilizadas, versiones de lenguaje de programación, estructura de directorios, permisos requeridos y otros recursos críticos asegurando la revisión y actualización periódica para mantener su relevancia y precisión.

C.E.4.3. Implementar pruebas específicas para el software desarrollado y la documentación de los casos de uso correspondientes, siguiendo prácticas y estándares reconocidos en el mercado para asegurar la calidad y funcionalidad del software.

C.E.4.4. Aplicar convenciones de versionado semántico que permitan una gestión coherente y comprensible de las versiones del software, facilitando la comunicación y el seguimiento de cambios entre los equipos de desarrollo y otros *stakeholders*.

C.E.4.5. Mantener un registro de cambios detallado (*changelog*) que documente de manera precisa todas las modificaciones realizadas en cada versión del software, cumpliendo con los lineamientos establecidos para asegurar la trazabilidad y transparencia en el desarrollo del software.

C.E.4.6. Asumir una actitud de liderazgo y toma de decisiones informadas en momentos clave del ciclo de vida del proyecto, guiando al equipo hacia el éxito.

C.E.4.7. Fomentar una cultura de mejora continua en todos los aspectos del desarrollo web, buscando constantemente maneras de perfeccionar los procesos y resultados.

RA.5. Desplegar aplicaciones web en las plataformas de distribución, cumpliendo con los requisitos y directrices específicas por la tienda de aplicaciones.

C.E.5.1. Identificar los requisitos específicos y directrices establecidos por las tiendas de aplicaciones (*Microsoft Store*, etc.) pertinentes para el despliegue de aplicaciones web, asegurando así la conformidad con las políticas y estándares de cada plataforma.

C.E.5.2. Ajustar la aplicación web siguiendo los estándares y requisitos técnicos de empaquetado establecidos por cada plataforma de distribución de aplicaciones, garantizando la compatibilidad y funcionalidad adecuada de la aplicación en cada entorno.

C.E.5.3. Realizar pruebas exhaustivas en la aplicación web de acuerdo con las políticas legales y normativas de la tienda de aplicaciones, así como los requerimientos específicos del cliente, para asegurar que la aplicación funcione sin errores, ni advertencias.

C.E.5.4. Validar las funcionalidades de la aplicación web desarrollada, considerando los requisitos de la plataforma de distribución adoptada, para responder eficazmente a los comentarios y solicitudes del cliente durante el proceso de desarrollo.

C.E.5.5. Integrar elementos visuales a la presentación de la aplicación web desarrollada, cumpliendo con las directrices y requisitos específicos de las tiendas de aplicaciones, para garantizar que los elementos visuales cumplan con los estándares de calidad y formato exigidos por las plataformas de distribución.

C.E.5.6. Realizar actualizaciones y mejoras periódicas en la aplicación web desarrolladas, respetando las directivas establecidas por las tiendas de aplicaciones, con el objetivo de mantener una experiencia de usuario óptima y satisfactoria en todo momento.

C.E.5.7. Monitorear de manera continua el desempeño y la aceptación de la aplicación en las plataformas de distribución mediante el uso de métricas, facilitando así la evaluación precisa de la implementación y la identificación de áreas de mejora.

C.E.5.8. Demostrar una alta conciencia ética en la gestión y procesamiento de datos, asegurando que todas las prácticas cumplen con las normativas y estándares de privacidad y seguridad.

Contenidos Formativos

Conceptuales:

Plan del proyecto de desarrollo de aplicaciones web

- Definición del ciclo de vida del desarrollo de software (SDLC).
- Etapas del ciclo de vida del desarrollo de aplicaciones web: concepción, planificación, análisis, diseño, implementación, pruebas, despliegue y mantenimiento.
- Importancia de cada etapa en el desarrollo de aplicaciones web.
- Descripción de las metodologías de desarrollo de software: Cascada, Agile, Scrum, Kanban, entre otras.
- Comparación entre metodologías tradicionales y ágiles.
- Criterios para la selección de metodologías de desarrollo.
- Definición de patrones de diseño y su importancia en el desarrollo de software.
- Descripción detallada de patrones como MVC, Singleton, Factory, Observer, etc.
- Beneficios de usar patrones de diseño en términos de eficiencia, mantenibilidad y escalabilidad.
- Identificación de las principales vulnerabilidades en el desarrollo de aplicaciones web.
- Explicación de los riesgos asociados a cada etapa del ciclo de vida del desarrollo.
- Técnicas y herramientas para el análisis y prevención de vulnerabilidades.
- Principios de optimización y rendimiento en aplicaciones web.
- Indicadores de calidad en el desarrollo de software.
- Estrategias para la mejora continua en el ciclo de vida del desarrollo.
- Definición y tipos de requerimientos: funcionales y no funcionales.
- Métodos de recolección y documentación de requerimientos.
- Técnicas de priorización de requerimientos (MoSCoW, Kano, etc.).
- Importancia del control de versiones en el desarrollo de software.
- Descripción de sistemas de control de versiones: Git, SVN, Mercurial, etc.
- Criterios para la selección de un sistema de control de versiones adecuado.
- Definición y tipos de ramas en el control de versiones (feature branches, release branches, hotfix branches, etc.).
- Métodos de fusión y resolución de conflictos en sistemas de control de versiones.
- Mejores prácticas para la gestión de ramas y fusiones.

Requisitos y especificaciones para el desarrollo de aplicaciones

- Definición de especificaciones de software.
- Componentes clave de una especificación detallada: funcionalidad, comportamiento esperado, requisitos funcionales y no funcionales.
- Importancia de las especificaciones claras y detalladas para el éxito del proyecto.
- Importancia de la planificación en el desarrollo de software.
- Técnicas de estimación de tiempo y recursos (PERT, CPM, estimaciones ágiles, etc.).
- Componentes de un plan de proyecto: cronograma, recursos, hitos, riesgos y dependencias.
- Importancia de las pruebas en el ciclo de vida del desarrollo de software.
- Tipos de pruebas: unitarias, de integración, de sistema, de aceptación, etc.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

- Elementos de una guía de pruebas: casos de prueba, datos de prueba, procedimientos de prueba, criterios de aceptación.
- Importancia de la gestión de cambios en proyectos de desarrollo de software.
- Componentes de un proceso de gestión de cambios: solicitud de cambio, evaluación, aprobación, implementación, seguimiento.
- Herramientas y técnicas para la gestión de cambios (sistemas de seguimiento de cambios, flujos de trabajo, etc.).
- Importancia de la comunicación efectiva en la gestión de proyectos.
- Técnicas y herramientas para la comunicación de cambios (reuniones, informes, herramientas de colaboración, etc.).
- Principios de gestión de ajustes y control de cambios en el desarrollo de software.

Metodologías de desarrollo de software

- Definición de requerimientos en el desarrollo de software.
- Tipos de requerimientos: funcionales y no funcionales.
- Técnicas de recolección de requerimientos (entrevistas, encuestas, talleres de trabajo).
- Principios de la metodología ágil y la planificación de sprints.
- Importancia de la priorización y agrupación de tareas en sprints.
- Técnicas para la planificación y organización de sprints (Scrum, Kanban).
- Definición y propósito del backlog en metodologías ágiles.
- Diferencia entre backlog de producto y backlog de sprint.
- Técnicas para la gestión efectiva del backlog de tareas.
- Importancia de la retroalimentación en el ciclo de vida de los sprints.
- Métodos de retroalimentación en metodologías ágiles (retrospectivas de sprint).
- Principios de mejora continua en el desarrollo ágil de software.
- Importancia de las herramientas ágiles en la gestión de proyectos.
- Descripción y propósito de tableros Kanban y herramientas de seguimiento de sprints (Jira, Trello, Asana).
- Principios de visualización del trabajo y gestión del flujo de trabajo.

Prácticas, documentación y testeo de versionados

- Importancia de la gestión de versiones en el desarrollo de software.
- Principios básicos de sistemas de control de versiones (Git, SVN, Mercurial).
- Estrategias de branching para diferentes entornos (producción, desarrollo, pruebas).
- Importancia de la documentación técnica en el desarrollo de software.
- Componentes clave de la documentación técnica: librerías, versiones, estructura de directorios, permisos, recursos críticos.
- Buenas prácticas para la redacción de documentación técnica.
- Importancia de las pruebas en el desarrollo de software.
- Tipos de pruebas: unitarias, de integración, de sistema, de aceptación.
- Estándares y prácticas reconocidas para la implementación de pruebas (TDD, BDD).
- Definición y propósito del versionado semántico.
- Reglas y convenciones del versionado semántico (MAJOR.MINOR.PATCH).
- Importancia del versionado coherente en la gestión del software.
- Definición y propósito de un changelog.
- Componentes clave de un changelog: fecha, versión, cambios, contribuyentes.
- Importancia de la trazabilidad y transparencia en el desarrollo de software.

Aplicaciones web y plataformas de distribución

- Definición de requisitos y directrices de tiendas de aplicaciones (App Store, Google Play, Microsoft Store).
- Importancia de la conformidad con las políticas y estándares de las plataformas.
- Principales directrices y requisitos comunes (seguridad, privacidad, contenido, calidad técnica).

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

- Definición y propósito de los estándares y requisitos técnicos de empaquetado.
- Diferencias entre los estándares de empaquetado de las principales plataformas de distribución.
- Importancia de la compatibilidad y funcionalidad de la aplicación en diferentes entornos.
- Importancia de las pruebas exhaustivas para el cumplimiento de políticas legales y normativas.
- Tipos de pruebas: funcionales, de rendimiento, de seguridad, de compatibilidad.
- Requerimientos específicos del cliente y su impacto en las pruebas.
- Importancia de la validación de funcionalidades en el desarrollo de aplicaciones web.
- Requisitos específicos de las plataformas de distribución adoptadas.
- Técnicas y herramientas para la validación de funcionalidades.
- Importancia de los elementos visuales en la presentación de la aplicación web.
- Directrices y requisitos específicos de las tiendas de aplicaciones para elementos visuales (iconos, capturas de pantalla, vídeos, etc.).
- Herramientas y técnicas para la creación y optimización de elementos visuales.
- Importancia de las actualizaciones y mejoras periódicas en la aplicación web.
- Directivas y políticas de las tiendas de aplicaciones respecto a las actualizaciones.
- Técnicas para planificar y realizar actualizaciones efectivas.
- Importancia del monitoreo continuo del desempeño y la aceptación de la aplicación.
- Métricas y análisis clave para evaluar el desempeño de la aplicación (descargas, ratings, comentarios, errores, etc.).
- Herramientas para el monitoreo y análisis de métricas (Google Analytics, App Annie, Firebase, etc.).

Contenidos Procedimentales

Planificación del proyecto de desarrollo de aplicaciones web

- Análisis de casos de estudio para identificar las etapas del ciclo de vida.
- Elaboración de diagramas de flujo que representen cada etapa del ciclo de vida.
- Evaluación de proyectos web existentes para reconocer las etapas del ciclo de vida implementadas.
- Evaluación de los procesos de una organización para determinar la metodología de desarrollo más adecuada.
- Aplicación de matrices de decisión para seleccionar la metodología adecuada.
- Realización de talleres prácticos para implementar diferentes metodologías en escenarios simulados.
- Implementación práctica de patrones de diseño en proyectos de desarrollo de aplicaciones web.
- Análisis de casos de estudio donde se apliquen diferentes patrones de diseño.
- Desarrollo de ejercicios de codificación para aplicar y entender el funcionamiento de los patrones de diseño.
- Realización de auditorías de seguridad en aplicaciones web.
- Aplicación de herramientas de análisis de vulnerabilidades como OWASP ZAP, Burp Suite, etc.
- Elaboración de informes de seguridad que detallen las vulnerabilidades encontradas y las recomendaciones para mitigarlas.
- Monitoreo y evaluación del rendimiento de aplicaciones web utilizando herramientas como Google Lighthouse, New Relic, etc.
- Aplicación de técnicas de optimización de código, bases de datos y recursos web.
- Realización de pruebas de rendimiento y calidad de manera periódica para identificar áreas de mejora.
- Realización de entrevistas, encuestas y talleres con stakeholders para la recolección de requerimientos.
- Documentación de requerimientos utilizando formatos estándar como user stories, casos de uso, etc.
- Aplicación de técnicas de priorización para definir el backlog del proyecto.
- Evaluación de las necesidades del proyecto y la experiencia del equipo para seleccionar un sistema de control de versiones.
- Configuración e implementación de un sistema de control de versiones.
- Realización de tutoriales y capacitaciones para el equipo sobre el uso del sistema seleccionado.
- Creación de ramas para el desarrollo paralelo de funcionalidades.
- Realización de fusiones de ramas utilizando herramientas de control de versiones.
- Resolución de conflictos de manera efectiva y documentación de los cambios realizados.

Análisis de requisitos y especificaciones

- Recolección de información relevante para la creación de especificaciones mediante entrevistas y encuestas con stakeholders.
- Redacción de especificaciones detalladas utilizando formatos estándar (user stories, casos de uso, requisitos funcionales y no funcionales).
- Revisión y validación de especificaciones con stakeholders para asegurar su claridad y completitud.
- Identificación de tareas y actividades necesarias para el desarrollo del proyecto.
- Estimación de tiempos y asignación de recursos utilizando técnicas adecuadas.
- Creación de un cronograma detallado y asignación de responsabilidades.
- Redacción de casos de prueba detallados para verificar cada requisito definido.
- Desarrollo de datos y scripts de prueba para automatizar las pruebas.
- Ejecución de pruebas y documentación de resultados.
- Desarrollo de un procedimiento formal para la gestión de cambios.
- Implementación de herramientas para el seguimiento y control de cambios.
- Realización de evaluaciones de impacto y priorización de solicitudes de cambio.
- Identificación de los stakeholders relevantes y sus necesidades de información.
- Elaboración de planes de comunicación para informar sobre cambios y ajustes.
- Uso de herramientas y técnicas adecuadas para comunicar cambios de manera efectiva.

Aplicación de metodologías de desarrollo de software

- Realización de entrevistas y talleres con stakeholders para recopilar requerimientos.
- Documentación de requerimientos utilizando formatos estándar (user stories, casos de uso).
- Asignación de niveles de prioridad a los requerimientos utilizando técnicas como MoSCoW o matrices de prioridad.
- Organización de reuniones de planificación de sprints (Sprint Planning).
- Agrupación de los resultados esperados en sprints según su nivel de prioridad.
- Creación de un backlog de sprint que incluya todas las tareas a realizar en el sprint correspondiente.
- Creación y mantenimiento de un backlog de producto que incluya todas las tareas pendientes y nuevas historias de usuario.
- Priorización y categorización de tareas en el backlog según su importancia y urgencia.
- Revisión y actualización regular del backlog para reflejar el estado actual del proyecto.
- Organización y realización de reuniones de retrospectiva al final de cada sprint.
- Documentación y análisis de los puntos fuertes y áreas de mejora identificadas durante las retrospectivas.
- Implementación de acciones de mejora basadas en la retroalimentación recibida.
- Configuración y uso de tableros Kanban o herramientas específicas de seguimiento de sprints.
- Actualización regular de las herramientas para reflejar el estado actual del trabajo y el progreso del proyecto.
- Monitoreo del flujo de trabajo y ajuste de tareas según sea necesario para cumplir con los tiempos establecidos.

Aplicación de prácticas de versionado

- Configuración de herramientas de gestión de versiones según los lineamientos organizacionales.
- Creación y manejo de branches específicos para desarrollo, pruebas y producción.
- Fusión de branches y resolución de conflictos de manera eficiente.
- Recolección de información relevante para la documentación técnica.
- Redacción detallada de la documentación incluyendo todos los componentes clave.
- Revisión y actualización periódica de la documentación técnica para mantener su relevancia y precisión.
- Diseño e implementación de pruebas específicas para el software desarrollado.
- Documentación de casos de uso y resultados de pruebas.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

- Uso de herramientas de automatización de pruebas (JUnit, Selenium) y generación de informes de prueba.
- Aplicación de convenciones de versionado semántico en el ciclo de vida del software.
- Documentación y comunicación de las versiones de software a los equipos de desarrollo y stakeholders.
- Uso de herramientas y flujos de trabajo que soporten el versionado semántico.
- Creación y mantenimiento de un changelog detallado y preciso.
- Documentación de cambios en el software conforme se desarrollan nuevas versiones.
- Revisión y actualización continua del changelog para reflejar el estado actual del software.

Despliegue de aplicaciones en plataformas de distribución

- Investigación y documentación de los requisitos y directrices de las tiendas de aplicaciones pertinentes.
- Análisis comparativo de los requisitos específicos de cada plataforma.
- Verificación de la conformidad de la aplicación web con los requisitos y directrices establecidos.
- Ajuste del código y recursos de la aplicación para cumplir con los requisitos técnicos de empaquetado.
- Pruebas de compatibilidad en diferentes entornos de desarrollo y producción.
- Implementación de soluciones técnicas para garantizar la funcionalidad adecuada en cada plataforma.
- Planificación y ejecución de pruebas exhaustivas basadas en las políticas legales y normativas.
- Utilización de herramientas de prueba automatizadas y manuales.
- Documentación de resultados de pruebas y corrección de errores detectados.
- Realización de pruebas de funcionalidad para asegurar el cumplimiento de los requisitos de la plataforma.
- Documentación y análisis de los comentarios y solicitudes del cliente.
- Ajuste y mejora continua de la aplicación según el feedback recibido.
- Creación e integración de elementos visuales siguiendo las directrices de las plataformas.
- Optimización de elementos visuales para asegurar su calidad y cumplimiento con los estándares.
- Revisión y ajuste continuo de los elementos visuales para mejorar la presentación de la aplicación.
- Planificación de actualizaciones y mejoras periódicas según las directivas de las plataformas.
- Implementación de actualizaciones y mejoras para asegurar una experiencia de usuario óptima.
- Monitoreo y evaluación del impacto de las actualizaciones en la funcionalidad y rendimiento de la aplicación.
- Configuración y uso de herramientas de monitoreo y análisis de métricas.
- Análisis de datos para evaluar el desempeño y la aceptación de la aplicación.

Identificación y priorización de áreas de mejora basadas en el análisis de métricas.

Contenidos Actitudinales

- Reconocimiento de la importancia de entregar software de alta calidad.
- Actitud proactiva para buscar constantemente eficiencias en el ciclo de vida.
- Habilidad para adaptarse a cambios y nuevas tecnologías.
- Colaboración efectiva con equipos multidisciplinares durante todo el ciclo de vida.
- Compromiso con la planificación y ejecución exitosa de proyectos web.
- Actitud de liderazgo y toma de decisiones en momentos clave del ciclo de vida.
- Cultura de mejora continua en todos los aspectos del desarrollo web.
- Aceptación de retroalimentación constructiva para perfeccionar procesos.
- Conciencia ética en la gestión y procesamiento de datos en aplicaciones web.
- Responsabilidad en la implementación de medidas de seguridad y privacidad.
- Compromiso con el uso eficiente de herramientas ágiles.
- Actitud colaborativa y proactiva para asegurar que todos los stakeholders estén informados y alineados.
- Compromiso con la transparencia y claridad en la comunicación de cambios.
- Compromiso con el monitoreo continuo y la evaluación precisa del desempeño de la aplicación.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

Requisitos básicos de infraestructura, espacio y apoyo a la impartición de clases

Infraestructura:

- Aula polivalente hasta 40 estudiantes

Instalaciones:

- Suministro eléctrico con tomas de corrientes.
- Iluminación adecuada.
- Sistema de ventilación.
- Sistema de aire acondicionado (aconsejable pero no excluyente).
- Accesibilidad universal a los espacios.
- Extintores y sistemas de seguridad.
- Conexión a Internet, por cable o red wifi.

Apoyo:

- Notebook o PC (docente).
- Proyector multimedia.
- Pizarra acrílica, pincel y borrador.
- Aplicaciones informáticas de uso general instaladas en los equipos.
- Sillas (una para cada estudiante).
- Mesas de trabajo (acorde a la cantidad de participantes), para trabajo con o sin computador.
- Escritorio y silla para el docente o instructor.

Equipamientos, herramientas manuales y elementos auxiliares para el desarrollo de las clases

Local/Infraestructura:

Espacio	Hasta 20 estudiantes	Hasta 40 estudiantes
Aula polivalente	1,5 m ² por estudiante	1,5 m ² por estudiante
Laboratorio de informática	40 m ²	80 m ²

Equipamientos / Herramientas/Materiales didácticos:

Aula polivalente

- Notebook o PC (docente).
- Proyector multimedia.
- Aplicaciones informáticas de uso general instaladas en los equipos.
- Sillas (una para cada estudiante).
- Mesas de trabajo (acorde a la cantidad de participantes), para trabajo con o sin computador.
- Escritorio y silla para el docente o instructor.
- Computadores necesarios en función del número de estudiantes con conexión a Internet (puede ubicarse en el laboratorio de informática o en el aula polivalente en el caso de disponer de carro portátil)
- Aplicaciones informáticas de uso general instaladas en los equipos.
- Pizarra acrílica, pincel y borrador.
- Otros materiales: regla T, escuadra (45°), cartabón (30 y 60°), compás, escalímetro, portaminas 0.3, 0.5 y 0.7, borrador, estilógrafo punta fina 0.1, 0.3 y 0.5, plantilla de letras, plantilla de símbolos, reglas curvas, hojas tamaño A2, A1 y A0 en distintos gramajes y tipos.
- Programas de diseño asistido por computador instalados en todos los equipos.
- Normativas:
- Normas de calidad.
- Reglamento de Baja Tensión de ANDE.
- Norma Paraguaya de instalaciones de BT (NP 2028 96).
- Ley N° 5668/2016 "De verificación y control de la calidad y seguridad de los productos y servicios en las instalaciones eléctricas" y Decreto N° 9265/2018 "Por el cual se Reglamenta la Ley N° 5668/2016..."
- Documentos de dibujo técnico.

Elementos necesarios, propios de la clase impartida:

- Software de base y de aplicaciones.
- Equipo informático.
- Equipos audiovisuales.
- Internet e intranet.
- Instrumentos de comunicaciones.
- Dispositivos multimedia.
- Materiales de oficina.
- Diccionario de datos.
- Lenguajes de programación
- Entornos integrados de desarrollo.

Elaborado por: Equipo A y DF-CIMA

Ministerio de Trabajo, Empleo y Seguridad Social (MTESS)
Sistema Nacional de Formación y Capacitación Laboral (SINAFOCAL)
Formato Estándar de Diseño Curricular de Módulos Formativos

Nombre del módulo	Realización de pruebas de calidad y automatización.	Código	MF0262_3
Familia profesional	Tecnologías de la Información y Comunicación (TICS).		
Perfil profesional	Desarrollo de aplicaciones web.		
Código del perfil profesional	TICS 0108_3		
Nivel de competencia (CNPP)	3 (tres)		
Duración del módulo	70 Horas	Modalidad	Presencial
Perfil de entrada del participante	<p>Requisitos mínimos del participante para el acceso a la capacitación Cumplir uno de los siguientes requisitos:</p> <ul style="list-style-type: none"> - Educación Media, concluida - Educación Media abierta, concluida. - Educación Media para Personas Jóvenes y Adultas, concluida. - Certificación de competencias ligada a perfiles del CNPP de nivel 2 de cualificación. 		
Perfil del profesional formador	Perfil académico (competencias técnicas)	Experiencia profesional requerida	
	<ul style="list-style-type: none"> - Título de profesor / docente en la especialidad de la familia profesional correspondiente o, - Título de Técnico Superior o, - Título de grado universitario o postgrado relacionado con el módulo formativo. 	Con titulación	Sin titulación
		1 año de experiencia laboral	3 años de experiencia laboral
<p>Competencias pedagógicas</p> <ul style="list-style-type: none"> - Título de profesor / docente en cualquier especialidad o, - Habilitación en docencia en EFTP requerida por cada Administración responsable, o - Título de profesor profesionalizante o, Certificado de especialización en el área con formación metodológica o pedagógica. 			
Perfil de salida del participante	<p>Una vez concluido los "5" módulos, el participante tendrá el perfil de: Desarrollo de aplicaciones web</p> <ul style="list-style-type: none"> - Otras ocupaciones y posibles puestos de trabajo: - Programador informático - Programador, programación técnica - Programador, banco de datos - Programador, informática - Verificador, programas - Otros programadores informáticos no especificados - Desarrollador Web - Desarrollador <i>Front End</i> - Desarrollador <i>Back End</i> - Técnicos en Programación Informática - Técnico, análisis informático - Programador, informática 		

Objetivo General	Realizar pruebas de calidad y automatización.
Resultados de aprendizaje (RA) y criterios de evaluación (CE) *	
RA.1. Verificar que todas las características y funcionalidades de la aplicación funcionan según lo previsto, asegurándose que los flujos de trabajo del usuario son intuitivos y libres de errores.	
<p>C.E.1.1. Definir escenarios de prueba que abarquen todas las características y funcionalidades de la aplicación garantizando que los escenarios reflejen situaciones reales de uso por parte de los usuarios.</p> <p>C.E.1.2. Evaluar los requerimientos del proyecto para establecer objetivos claros y específicos para las pruebas, teniendo en cuenta la complejidad de la aplicación y los criterios de éxito definidos.</p> <p>C.E.1.3. Desarrollar un plan de pruebas detallado que incluya casos de prueba específicos y criterios de aceptación bien definidos, garantizando que todas las funcionalidades críticas de la aplicación sean evaluadas adecuadamente.</p> <p>C.E.1.4. Seleccionar las herramientas y tecnologías apropiadas para ejecutar las pruebas, evaluando su compatibilidad con el entorno de desarrollo y asegurándose de que cumplan con los requisitos específicos del proyecto de desarrollo de aplicaciones.</p> <p>C.E.1.5. Demostrar responsabilidad y dedicación autónoma al cumplir con las tareas asignadas, asegurando que los plazos y objetivos del proyecto se cumplan eficazmente.</p> <p>C.E.1.6. Mantener una actitud proactiva en la búsqueda continua de mejoras en el rendimiento y la eficiencia de los servicios móviles, optimizando el funcionamiento de las aplicaciones.</p>	
RA.2. Evaluar la usabilidad de la aplicación, asegurando que la interfaz de usuario (UI) fluida y responsiva en diferentes dispositivos y resoluciones de pantalla.	
<p>C.E.2.1. Desarrollar scripts de prueba utilizando herramientas y lenguajes de automatización apropiados aplicando las mejores prácticas y cumpliendo con los estándares de la industria para garantizar la calidad y eficiencia de las pruebas automatizadas.</p> <p>C.E.2.2. Identificar los dispositivos y las resoluciones de pantalla necesarias para realizar pruebas de usabilidad que aseguren una experiencia consistente y adecuada para los usuarios en diferentes dispositivos.</p> <p>C.E.2.3. Configurar entornos de pruebas automatizadas que incluyan la gestión de datos necesaria y la simulación de usuarios, asegurando que los entornos estén preparados para ejecutar pruebas de manera efectiva y replicable.</p> <p>C.E.2.4. Aplicar pruebas automatizadas en diferentes niveles (unitarias, de integración, de sistema y de aceptación), validando la efectividad y la cobertura de las pruebas para asegurar que la aplicación funcione y cumpla con los requisitos especificados.</p> <p>C.E.2.5. Mantener una actitud proactiva en la búsqueda continua de mejoras en el rendimiento y la eficiencia de los servicios móviles, optimizando el funcionamiento de las aplicaciones.</p> <p>C.E.2.6. Implementar buenas prácticas en el diseño y desarrollo de bases de datos móviles, asegurando la calidad, integridad y eficiencia de los datos gestionados.</p>	
RA.3. Testear la aplicación en múltiples dispositivos con diferentes tamaños de pantalla, sistemas operativos y configuraciones de hardware, identificando y resolviendo problemas específicos de cada dispositivo o plataforma.	
<p>C.E.3.1. Utilizar herramientas de gestión de versionado conforme a los lineamientos establecidos para el manejo de branches, asegurando una estructura adecuada y controlada para el desarrollo, pruebas y despliegue del software.</p> <p>C.E.3.2. Elaborar la documentación técnica completa que detalle los requisitos esenciales para la ejecución del software, incluyendo información sobre librerías, versión del lenguaje de programación, directorios adicionales, permisos y otros recursos necesarios, asegurando que el entorno esté configurado y preparado.</p> <p>C.E.3.3. Documentar casos de prueba específicos utilizando estándares reconocidos del mercado, garantizando que las pruebas sean completas y efectivas para validar las funcionalidades del software.</p> <p>C.E.3.4. Configurar entornos de pruebas automatizadas que incluyan la gestión de datos necesaria y la simulación de usuarios, asegurando que los entornos estén a preparados para ejecutar pruebas de manera eficiente y repetible.</p> <p>C.E.3.5. Verificar que las características y funcionalidades de la aplicación cumplan con los requisitos especificados, asegurando su correcto funcionamiento bajo condiciones normales de uso antes de proceder con pruebas adicionales.</p>	

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

C.E.3.6. Realizar pruebas tanto manuales como automatizadas para evaluar la funcionalidad, rendimiento, seguridad y usabilidad del software, cumpliendo con los estándares de calidad y satisfaga las expectativas del usuario final.

C.E.3.7. Reconocer la importancia de los servicios móviles en la mejora de la funcionalidad y el rendimiento de las aplicaciones, contribuyendo activamente a su desarrollo y optimización.

RA.4. Verificar el rendimiento y la escalabilidad de la aplicación, evaluando bajo diferentes condiciones de uso, incluyendo la carga máxima y el uso prolongado.

C.E.4.1. Medir la capacidad de respuesta de la aplicación, evaluando los tiempos de carga y respuesta en diversos escenarios de uso, garantizando que estos se mantengan dentro de los límites establecidos para proporcionar una experiencia de usuario óptima.

C.E.4.2. Realizar pruebas de carga máxima simulando el uso simultáneo de la aplicación por múltiples usuarios, identificando y documentando los puntos de falla para determinar la capacidad máxima de usuarios concurrentes que la aplicación puede manejar con rendimiento óptimo.

C.E.4.3. Ejecutar pruebas de estrés utilizando herramientas aumentando gradualmente la carga sobre la aplicación para evaluar su comportamiento en situaciones extremas, asegurando que no se produzcan errores críticos ni caídas del sistema bajo condiciones de alta demanda.

C.E.4.4. Analizar el uso de recursos monitoreando el consumo del microprocesador de memoria, ancho de banda y otras métricas pertinentes para asegurar que la aplicación utilice los recursos de manera eficiente, evitando sobrecargar el sistema operativo y los dispositivos donde se despliega.

C.E.4.5. Evaluar la estabilidad de la aplicación ejecutándola de forma continua durante períodos prolongados, identificando y corrigiendo problemas potenciales como fugas de memoria, fugas de recursos o degradación del rendimiento para mantener un funcionamiento óptimo a lo largo del tiempo.

C.E.4.6. Realizar pruebas de rendimiento de forma periódica para verificar que la aplicación mantenga su rendimiento óptimo ante actualizaciones, cambios en la infraestructura o incrementos en la base de usuarios, garantizando una experiencia de usuario consistente y de alta calidad a lo largo del tiempo.

C.E.4.7. Documentar los resultados de las pruebas de rendimiento mediante un informe detallado que contenga hallazgos, conclusiones y recomendaciones para mejoras, asegurando que cumpla con el formato y las directrices establecidas.

RA.5. Garantizar la seguridad y privacidad, probando la aplicación para detectar vulnerabilidades que puedan ser explotadas por atacantes.

C.E. 5.1. Realizar el análisis de vulnerabilidades utilizando herramientas automáticas y manuales, identificando y documentando posibles puntos débiles en la aplicación conforme a las mejores prácticas de seguridad establecidas.

C.E. 5.2. Realizar pruebas para verificar la seguridad, asegurando que los mecanismos de control de acceso y autenticación de usuarios funcionen y estén protegidos contra ataques de fuerza bruta y suplantación de identidad.

C.E. 5.3. Realizar pruebas *pen test* utilizando metodologías reconocidas como las de OWASP simulando ataques reales para identificar y mitigar vulnerabilidades, inyecciones de SQL, XSS, CSRF y otros tipos de ataques.

C.E. 5.4. Verificar la gestión de sesiones asegurando que se manejen de forma segura, implementando mecanismos apropiados para el inicio, cierre, expiración y protección contra la fijación de sesiones, garantizando así la seguridad de la aplicación.

C.E. 5.5. Evaluar la seguridad de las interfaces de programación de aplicaciones (API), asegurándose de que estén libres de vulnerabilidades que puedan ser explotadas por atacantes, garantizando la integridad y seguridad de los datos y servicios expuestos.

C.E. 5.6. Demostrar una actitud proactiva hacia la exploración y comprensión de nuevas tecnologías y tendencias en servicios móviles, manteniéndose actualizado y aplicando innovaciones al trabajo.

C.E. 5.7. Colaborar con equipos de desarrollo para asegurar una integración efectiva de servicios móviles, facilitando el trabajo conjunto y el logro de objetivos comunes.

RA.6. Cumplir con los estándares de calidad y las normativas aplicables garantizando que la aplicación sea confiable y útil para los usuarios.

C.E. 6.1. Revisar las guías de publicación de la plataforma de distribución, identificando todos los requisitos y restricciones relevantes, documentando los puntos críticos que deben cumplirse para asegurar la aceptación de la aplicación en la tienda de aplicaciones correspondiente.

C.E. 6.2. Verificar la compatibilidad de la aplicación con los dispositivos y versiones de sistemas operativos especificados en las guías de la plataforma de distribución, asegurándose de que la aplicación funcione en todas las plataformas soportadas sin errores significativos.

C.E. 6.3. Generar la documentación completa y precisa de la aplicación, incluyendo descripciones detalladas, capturas de pantalla relevantes, videos de demostración si es necesario, y cualquier otra información requerida por las tiendas de aplicaciones para facilitar el proceso de publicación.

C.E. 6.4. Documentar el cumplimiento de normativas y procedimientos, manteniendo registros detallados de todas las acciones realizadas para asegurar el cumplimiento de las guías de las plataformas de distribución.

C.E. 6.5. Mostrar una alta sensibilidad hacia la privacidad y seguridad de los datos en servicios móviles, implementando prácticas que protejan la información de los usuarios.

C.E. 6.6. Demostrar habilidad para anticiparse y adaptarse proactivamente a cambios en el panorama de desarrollo móvil, asegurando la relevancia y competitividad de las aplicaciones desarrolladas.

Contenidos Formativos

Conceptuales:

Características y funcionalidades

- Definición de escenarios de prueba en el contexto del desarrollo de aplicaciones móviles.
- Importancia de los escenarios de prueba para asegurar la calidad y funcionalidad de la aplicación.
- Técnicas para identificar escenarios de prueba basados en el comportamiento y las expectativas de los usuarios.
- Importancia de la evaluación de requerimientos para establecer objetivos de prueba claros.
- Relación entre los requerimientos del proyecto, la complejidad de la aplicación y los objetivos de las pruebas.
- Criterios de éxito en pruebas de aplicaciones móviles.
- Componentes de un plan de pruebas detallado: casos de prueba, criterios de aceptación, cronograma de pruebas.
- Importancia de los casos de prueba específicos y los criterios de aceptación bien definidos.
- Técnicas para la creación de casos de prueba efectivos.
- Importancia de seleccionar las herramientas y tecnologías adecuadas para las pruebas.
- Factores a considerar al seleccionar herramientas de prueba: compatibilidad, funcionalidad, facilidad de uso, costo.
- Principales herramientas y tecnologías de prueba para aplicaciones móviles (Appium, Espresso, XCUITest, etc.).

Usabilidad de la aplicación

- Definición y propósito de los scripts de prueba automatizados.
- Herramientas y lenguajes de automatización de pruebas (Selenium, JUnit, TestNG).
- Mejores prácticas y estándares de la industria para el desarrollo de scripts de prueba.
- Importancia de las pruebas de usabilidad en el desarrollo de aplicaciones móviles.
- Diversidad de dispositivos móviles y resoluciones de pantalla en el mercado.
- Criterios para la selección de dispositivos y resoluciones representativas para las pruebas.
- Definición y componentes de un entorno de pruebas automatizadas.
- Importancia de la gestión de datos y la simulación de usuarios en las pruebas automatizadas.
- Principios para la configuración de entornos de pruebas efectivos y replicables.
- Niveles de pruebas automatizadas: unitarias, de integración, de sistema y de aceptación.
- Importancia de la efectividad y cobertura de las pruebas en el desarrollo de aplicaciones móviles.
- Técnicas para validar la efectividad y cobertura de las pruebas automatizadas.

Aplicación responsiva

- Definición y propósito de las herramientas de gestión de versionado (Git, SVN, Mercurial).
- Importancia del manejo de *branches* en el desarrollo de software.
- Estrategias de branching (feature branches, release branches, hotfix branches).
- Importancia de la documentación técnica en el desarrollo de software.
- Componentes clave de la documentación técnica: librerías, versiones de lenguaje, estructura de directorios, permisos, recursos necesarios.
- Buenas prácticas para la redacción de documentación técnica.
- Importancia de la documentación de casos de prueba en el ciclo de vida del desarrollo de software.
- Estándares reconocidos para la documentación de pruebas (IEEE 829, ISTQB).
- Componentes de un caso de prueba: descripción, precondiciones, pasos, resultados esperados.
- Definición y componentes de un entorno de pruebas automatizadas.
- Importancia de la gestión de datos y la simulación de usuarios en las pruebas automatizadas.
- Principios para la configuración de entornos de pruebas efectivos y repetibles.
- Importancia de la verificación de características y funcionalidades en el desarrollo de software.
- Métodos y técnicas de verificación de software (revisiones, inspecciones, *walkthroughs*).
- Criterios de aceptación y requisitos especificados del proyecto.
- Diferencias y complementariedad de las pruebas manuales y automatizadas.
- Tipos de pruebas: funcionales, de rendimiento, de seguridad, de usabilidad.
- Estándares de calidad y expectativas del usuario final en el desarrollo de software.

Rendimiento y la escalabilidad de la aplicación

- Importancia de la capacidad de respuesta en la experiencia de usuario.
- Definición de tiempos de carga y respuesta en aplicaciones.
- Estándares y métricas de rendimiento establecidos por la organización.
- Definición y propósito de las pruebas de carga máxima.
- Importancia de identificar la capacidad máxima de usuarios concurrentes.
- Principios de simulación de usuarios para pruebas de carga.
- Definición y propósito de las pruebas de estrés.
- Diferencias entre pruebas de carga y pruebas de estrés.
- Principios para evaluar el comportamiento de la aplicación bajo condiciones extremas.
- Importancia del uso eficiente de recursos en aplicaciones.
- Métricas clave para el monitoreo del uso de recursos (CPU, memoria, ancho de banda).
- Herramientas para el monitoreo de recursos (como New Relic, Dynatrace, Nagios).
- Definición de estabilidad en el contexto de aplicaciones.
- Problemas comunes que afectan la estabilidad (fugas de memoria, fugas de recursos, degradación del rendimiento).
- Técnicas para la evaluación y mejora de la estabilidad de la aplicación.
- Importancia de la documentación de resultados de pruebas de rendimiento.
- Componentes clave de un informe de pruebas de rendimiento: hallazgos, conclusiones, recomendaciones.
- Directrices organizacionales para la documentación de informes de pruebas.
- Importancia de las pruebas de rendimiento periódicas en el ciclo de vida del software.
- Factores que pueden afectar el rendimiento de la aplicación (actualizaciones, cambios en la infraestructura, incremento en la base de usuarios).
- Métodos y técnicas para realizar pruebas de rendimiento periódicas.

Seguridad y privacidad

- Definición y propósito del análisis de vulnerabilidades.
- Diferencias entre herramientas automáticas y manuales para el análisis de vulnerabilidades.
- Principales herramientas de análisis de vulnerabilidades (OWASP ZAP, Burp Suite, Nessus).
- Importancia de la autenticación y autorización en la seguridad de aplicaciones.
- Métodos y mecanismos de autenticación y control de acceso (OAuth, JWT, SAML).
- Principales amenazas a la autenticación y autorización (ataques de fuerza bruta, suplantación de identidad).
- Definición y propósito de las pruebas de penetración.
- Principales tipos de vulnerabilidades (SQL Injection, XSS, CSRF).
- Metodologías reconocidas para pruebas de penetración (OWASP Testing Guide).
- Importancia de la gestión segura de sesiones en aplicaciones web.
- Principales vulnerabilidades asociadas a la gestión de sesiones (fijación de sesión, secuestro de sesión).
- Buenas prácticas para la gestión de sesiones (inicio y cierre seguro, expiración de sesiones, uso de tokens).
- Importancia de la seguridad en las API.
- Principales vulnerabilidades en API (API security misconfigurations, Broken Object Level Authorization).
- Buenas prácticas para asegurar la integridad y seguridad de las API (autenticación, autorización, validación de datos).

Estándares de calidad y normativas aplicables

- Importancia de las guías de publicación de las plataformas de distribución.
- Requisitos y restricciones comunes en las guías de publicación (contenido, seguridad, privacidad, funcionalidad).
- Consecuencias del incumplimiento de los requisitos de las plataformas.
- Importancia de la compatibilidad en el desarrollo de aplicaciones móviles.
- Principales dispositivos y versiones de sistemas operativos soportados por las plataformas de distribución.
- Métodos para asegurar la compatibilidad de la aplicación (pruebas en dispositivos reales, emuladores).
- Importancia de la documentación en el proceso de publicación de aplicaciones.
- Componentes clave de la documentación requerida por las tiendas de aplicaciones (descripciones, capturas de pantalla, videos de demostración).
- Mejores prácticas para la creación de documentación completa y precisa.
- Importancia de la documentación del cumplimiento de normativas y procedimientos.
- Normativas y procedimientos comunes en las plataformas de distribución de aplicaciones.
- Métodos para mantener registros detallados de las acciones realizadas.

Contenidos Procedimentales:

Identificación de todas las características y funcionalidades de la aplicación móvil.

- Recolección de datos sobre el uso esperado de la aplicación móvil mediante entrevistas, encuestas y análisis de usuarios.
- Definición de escenarios de prueba que cubran todas las características y funcionalidades de la aplicación.
- Documentación de los escenarios de prueba, incluyendo descripciones detalladas y pasos a seguir.
- Análisis detallado de los requerimientos del proyecto para el entendimiento de las funcionalidades y características clave.
- Establecimiento de objetivos específicos para las pruebas, alineados con los requerimientos y la complejidad de la aplicación.
- Definición de criterios de éxito para las pruebas basados en los objetivos establecidos.
- Desarrollo de un plan de pruebas detallado que incluya todos los casos de prueba necesarios.
- Definición de criterios de aceptación para cada caso de prueba.
- Revisión y aprobación del plan de pruebas con los *stakeholders* del proyecto.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

- Evaluación de las herramientas y tecnologías disponibles para las pruebas.
- Selección de las herramientas más adecuadas basadas en su compatibilidad con el entorno de desarrollo y los requisitos del proyecto.
- Configuración e integración de las herramientas seleccionadas en el proceso de desarrollo.

Evaluación de la usabilidad de la aplicación

- Configuración de las herramientas de automatización de pruebas seleccionadas.
- Desarrollo de scripts de prueba utilizando las mejores prácticas (modularidad, reutilización de código, claridad).
- Ejecución y validación de los scripts de prueba para asegurar su efectividad y eficiencia.
- Investigación y análisis del mercado para identificar los dispositivos móviles y resoluciones de pantalla más utilizados por los usuarios.
- Selección de una muestra representativa de dispositivos y resoluciones para realizar las pruebas.
- Configuración de los dispositivos seleccionados para llevar a cabo pruebas de usabilidad.
- Configuración de entornos de pruebas automatizadas que incluyan herramientas y datos necesarios.
- Implementación de simulaciones de usuarios para replicar condiciones reales de uso.
- Verificación de la preparación y replicabilidad de los entornos de pruebas antes de la ejecución.
- Desarrollo y ejecución de pruebas automatizadas en diferentes niveles.
- Monitoreo y análisis de los resultados de las pruebas para validar su efectividad.
- Evaluación de la cobertura de las pruebas para asegurar que todos los requisitos especificados sean cumplidos.

Prueba de aplicaciones en múltiples dispositivos

- Configuración y uso de herramientas de gestión de versionado según los lineamientos organizacionales.
- Creación y manejo de branches para desarrollo, pruebas y despliegue.
- Fusión y resolución de conflictos de código de manera controlada.
- Recolección de información relevante sobre los requisitos esenciales para la ejecución del software.
- Redacción detallada de la documentación técnica, incluyendo todos los componentes necesarios.
- Revisión y actualización periódica de la documentación técnica para mantener su precisión y relevancia.
- Creación y documentación de casos de prueba específicos según los estándares reconocidos.
- Verificación de que los casos de prueba cubran todas las funcionalidades críticas del software.
- Revisión y actualización de los casos de prueba en base a los resultados obtenidos y el feedback recibido.
- Configuración de entornos de pruebas automatizadas que incluyan herramientas y datos necesarios.
- Implementación de simulaciones de usuarios para replicar condiciones reales de uso.
- Verificación de la preparación y replicabilidad de los entornos de pruebas antes de la ejecución.
- Verificación de características y funcionalidades de la aplicación contra los requisitos especificados.
- Realización de pruebas bajo condiciones normales de uso para asegurar el correcto funcionamiento.
- Documentación de los resultados de la verificación y preparación para pruebas adicionales.
- Planificación y ejecución de pruebas manuales y automatizadas para evaluar todas las áreas del software.
- Utilización de herramientas y técnicas adecuadas para cada tipo de prueba.
- Análisis y documentación de los resultados de las pruebas, incluyendo recomendaciones de mejora.

Verificación del rendimiento y escalabilidad de la aplicación

- Identificación de escenarios de uso representativos para la medición de tiempos de carga y respuesta.
- Uso de herramientas de monitoreo y prueba de rendimiento (como Google Lighthouse, JMeter, GTmetrix).
- Análisis de resultados para asegurar que los tiempos se mantengan dentro de los límites establecidos.
- Configuración de herramientas de prueba de carga (como Apache JMeter, LoadRunner) para simular múltiples usuarios concurrentes.
- Ejecución de pruebas de carga máxima y monitoreo del rendimiento de la aplicación.
- Identificación y documentación de puntos de falla y capacidad máxima de usuarios concurrentes.
- Planificación y configuración de pruebas de estrés utilizando herramientas adecuadas.
- Ejecución gradual de pruebas de estrés para evaluar el comportamiento de la aplicación.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

- Monitoreo y análisis de resultados para asegurar que no se produzcan errores críticos ni caídas del sistema.
- Configuración de herramientas de monitoreo para evaluar el consumo de recursos.
- Monitoreo continuo y análisis del uso de recursos durante la ejecución de la aplicación.
- Identificación de áreas de mejora para optimizar el uso de recursos.
- Ejecución continua de la aplicación durante períodos prolongados para evaluar su estabilidad.
- Monitoreo y detección de problemas potenciales como fugas de memoria y recursos.
- Implementación de soluciones para corregir problemas y mejorar la estabilidad de la aplicación.
- Recopilación y análisis de resultados de pruebas de rendimiento.
- Redacción de un informe detallado que incluya todos los componentes clave.
- Revisión y ajuste del informe para asegurar el cumplimiento con las directrices organizacionales.
- Planificación y ejecución de pruebas de rendimiento periódicas.
- Monitoreo y análisis de resultados para verificar el mantenimiento del rendimiento óptimo.
- Implementación de ajustes y mejoras basadas en los resultados de las pruebas periódicas.

Detección de vulnerabilidades

- Configuración y uso de herramientas automáticas para identificar vulnerabilidades.
- Ejecución de análisis manuales para complementar los resultados de las herramientas automáticas.
- Documentación de puntos débiles identificados y recomendaciones para su mitigación.
- Configuración y ejecución de pruebas para verificar la robustez de los mecanismos de autenticación y autorización.
- Simulación de ataques de fuerza bruta y suplantación de identidad para evaluar la seguridad.
- Documentación de resultados y recomendaciones para mejorar la seguridad de la autenticación y autorización.
- Planificación y ejecución de pruebas de penetración simulando ataques reales.
- Uso de herramientas y técnicas específicas para identificar vulnerabilidades.
- Documentación de vulnerabilidades encontradas y recomendaciones para su mitigación.
- Verificación de la implementación de mecanismos seguros para el inicio y cierre de sesión.
- Configuración de políticas de expiración de sesiones y protección contra la fijación de sesiones.
- Realización de pruebas para asegurar la gestión segura de sesiones.
- Evaluación de la seguridad de las API utilizando herramientas y técnicas específicas.
- Identificación de vulnerabilidades y puntos débiles en las API.
- Implementación de medidas de seguridad para proteger las API contra ataques.

Cumplimiento de los estándares de calidad y las normativas aplicables

- Revisión detallada de las guías de publicación de la plataforma de distribución.
- Identificación y documentación de todos los requisitos y restricciones relevantes.
- Creación de una lista de verificación con los puntos críticos que deben cumplirse para asegurar la aceptación de la aplicación.
- Verificación de la compatibilidad de la aplicación con los dispositivos y versiones de sistemas operativos especificados en las guías de la plataforma.
- Realización de pruebas en dispositivos reales y emuladores para asegurar el funcionamiento adecuado.
- Documentación de los resultados de las pruebas y resolución de cualquier error significativo identificado.
- Creación de descripciones detalladas de la aplicación, destacando sus características y beneficios.
- Captura de pantallas relevantes y, si es necesario, creación de videos de demostración.
- Revisión y compilación de toda la información requerida por las tiendas de aplicaciones.
- Documentación de todas las acciones realizadas para cumplir con las normativas y procedimientos de las plataformas de distribución.
- Mantenimiento de registros detallados y organizados que evidencien el cumplimiento de las guías.
- Revisión periódica de los registros para asegurar su precisión y completitud.

Contenidos Actitudinales:

- Responsable ante las tareas asignadas y dedicación autónoma en cumplimiento con los plazos y objetivos del proyecto.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

- Actitud proactiva hacia la mejora continua del rendimiento y eficiencia en servicios móviles.
- Reconocimiento de la importancia de los servicios móviles en la mejora de la funcionalidad y rendimiento de las aplicaciones.
- Actitud proactiva hacia la exploración y comprensión de nuevas tecnologías y tendencias en servicios móviles.
- Sensibilidad hacia la privacidad y seguridad de los datos en servicios móviles.
- Compromiso con la implementación de buenas prácticas en el diseño y desarrollo de bases de datos móviles.
- Compromiso con la calidad y estabilidad en el despliegue de servicios móviles.
- Creativo proponiendo soluciones innovadoras resolviendo problemas durante el desarrollo de aplicaciones móviles.
- Flexibilidad y adaptabilidad para evaluar y cambiar herramientas según sea necesario.
- Habilidad para colaborar en equipos de desarrollo para la integración efectiva de servicios móviles.
- Actitud abierta y receptiva hacia la adopción de nuevas tecnologías en servicios móviles.
- Habilidad para anticipar y adaptarse proactivamente a cambios en el panorama de desarrollo móvil.
- Diligencia en la ejecución y revisión de pruebas en todos los niveles.
- Compromiso con la protección de los datos y servicios expuestos a través de las APIs.
- Dedicación a la transparencia y cumplimiento de las guías de las plataformas de distribución.

Misión: Regular, certificar y acreditar los procesos de formación y capacitación laboral, con enfoque inclusivo en todos los niveles ocupacionales, garantizando y contribuyendo a la productividad y competitividad de la fuerza laboral activa para la inserción laboral, social y económica de los beneficiarios, articulados con los sectores público y privado.

Requisitos básicos de infraestructura, espacio y apoyo a la impartición de clases

Infraestructura:

- Aula polivalente hasta 40 estudiantes

Instalaciones:

- Suministro eléctrico con tomas de corrientes.
- Iluminación adecuada.
- Sistema de ventilación.
- Sistema de aire acondicionado (aconsejable pero no excluyente).
- Accesibilidad universal a los espacios.
- Extintores y sistemas de seguridad.
- Conexión a Internet, por cable o red wifi.

Apoyo:

- Notebook o PC (docente).
- Proyector multimedia.
- Pizarra acrílica, pincel y borrador.
- Aplicaciones informáticas de uso general instaladas en los equipos.
- Sillas (una para cada estudiante).
- Mesas de trabajo (acorde a la cantidad de participantes), para trabajo con o sin computador.
- Escritorio y silla para el docente o instructor.

Equipamientos, herramientas manuales y elementos auxiliares para el desarrollo de las clases

Local/Infraestructura:

Espacio	Hasta 20 estudiantes	Hasta 40 estudiantes
Aula polivalente	1,5 m ² por estudiante	1,5 m ² por estudiante
Laboratorio de informática	40 m ²	80 m ²

Equipamientos / Herramientas/Materiales didácticos:

Aula polivalente

- Notebook o PC (docente).
- Proyector multimedia.
- Aplicaciones informáticas de uso general instaladas en los equipos.
- Sillas (una para cada estudiante).
- Mesas de trabajo (acorde a la cantidad de participantes), para trabajo con o sin computador.
- Escritorio y silla para el docente o instructor.
- Computadores necesarios en función del número de estudiantes con conexión a Internet (puede ubicarse en el laboratorio de informática o en el aula polivalente en el caso de disponer de carro portátil)
- Aplicaciones informáticas de uso general instaladas en los equipos.
- Pizarra acrílica, pincel y borrador.
- Otros materiales: regla T, escuadra (45°), cartabón (30 y 60°), compás, escalímetro, portaminas 0.3, 0.5 y 0.7, borrador, estilógrafo punta fina 0.1, 0.3 y 0.5, plantilla de letras, plantilla de símbolos, reglas curvas, hojas tamaño A2, A1 y A0 en distintos gramajes y tipos.
- Programas de diseño asistido por computador instalados en todos los equipos.
- Normativas:
- Normas de calidad.
- Reglamento de Baja Tensión de ANDE.
- Norma Paraguaya de instalaciones de BT (NP 2028 96).
- Ley N° 5668/2016 “De verificación y control de la calidad y seguridad de los productos y servicios en las instalaciones eléctricas” y Decreto N° 9265/2018 “Por el cual se Reglamenta la Ley N° 5668/2016...”
- Documentos de dibujo técnico.

Elementos necesarios, propios de la clase impartida:

- Software de base y de aplicaciones.
- Equipo informático.
- Equipos audiovisuales.
- Internet e intranet.
- Instrumentos de comunicaciones.
- Dispositivos multimedia.
- Materiales de oficina.
- Diccionario de datos.
- Lenguajes de programación
- Entornos integrados de desarrollo.

Elaborado por: Equipo A y DF-CIMA