

BRIEF PLAN FORMATIVO

1. INFORMACIÓN GENERAL DEL PLAN DE ESTUDIO	
Nombre Plan Formativo	Especialización Desarrollo de Aplicaciones Front-End
Perfil Asociado	Desarrollador Front-End Semi Senior
Tipo de Plan Formativo	De especialización
Duración	132 horas
Descripción ocupación y campo laboral asociado	<p>Este plan está diseñado para desarrollar las habilidades necesarias para construir y mantener aplicaciones o sitios web, usando el stack de tecnologías fundamentales para el desarrollo front end, es decir HTML, CSS y JavaScript. Asimismo, la implementación de aplicaciones web interactivas y aplicaciones web progresivas mediante la utilización del framework React.</p> <p>El campo laboral corresponde a Organizaciones, ya sean públicas o privadas, transversal a todas las industrias (retail, banca, salud, minería, manufactura, servicios), que realicen desarrollo, implementación y/o mantenimiento de software, ya sean productivas, servicios, gubernamentales, o que prestan servicios TI a otras organizaciones.</p>
Requisito de ingreso al plan formativo	<p>Los requisitos de entrada son:</p> <ul style="list-style-type: none"> ● Desarrolladores de aplicaciones web ● Conocimiento medio de html, css y javascript ● Al menos 2 años de experiencia
Competencias del plan formativo	<p>Competencia:</p> <ul style="list-style-type: none"> ● Desarrollar aplicaciones web desde el lado del cliente, que den solución a las necesidades de la organización, y aplicando las buenas prácticas de la industria para obtener un producto con niveles de calidad requeridos. <p>Unidades de Competencia:</p> <ul style="list-style-type: none"> ● Implementar interfaces de usuario web responsivas utilizando html5 y frameworks de estilos siguiendo las buenas prácticas de la industria

	<ul style="list-style-type: none"> • Codificar piezas de software de mediana complejidad en lenguaje Javascript utilizando paradigmas de orientación a objetos, orientación a eventos y programación asíncrona para resolver problemáticas de acuerdo a las necesidades de la industria • Implementar una interfaz de usuario web con elementos interactivos utilizando el framework Vue.js para dar solución a un requerimiento • Implementar una aplicación Front-End utilizando un framework orientado a componentes para dar solución a un requerimiento • Implementar una aplicación web progresiva utilizando el entorno React de acuerdo a los requerimientos
Descripción general del Plan Formativo	<p>Este plan formativo está diseñado para profesionales que trabajan en el área de desarrollo web, por lo tanto se da por supuesto que los participantes tienen conocimiento al menos básico o medio respecto a desarrollo web, es decir, HTML, CSS y JavaScript.</p> <p>Dado lo anterior, este plan formativo se enfoca en profundizar conceptos, técnicas y herramientas dentro de las cuales se incluye CSS (conceptos, metodologías CSS, responsividad, Bootstrap), JavaScript avanzado, implementación de aplicaciones interactivas con el framework React, y aplicaciones progresivas (PWA).</p> <p>Finalmente, se entregan conceptos relacionados con las metodologías ágiles (scrum) y devops, con el objeto que los participantes puedan situarse en una célula de desarrollo y ejecutar su rol de forma eficiente.</p>

2. MÓDULOS DEL PLAN FORMATIVO			
N° de Módulo	Nombre del módulo	Descripción	Horas de duración
1	Orientación al perfil y metodología del curso	<p>Módulo introductorio al curso que cubre los siguientes elementos:</p> <ul style="list-style-type: none"> • Comprender el perfil del analista de datos y los módulos del curso • Comprender la metodología del curso, el cual se centra en el portafolio de trabajos 	3

2	Desarrollo de la interfaz de usuario web	<p>Al finalizar este módulo el participante será capaz de personalizar una aplicación web utilizando el framework CSS bootstrap, la metodología BEM y el preprocesador Sass para el manejo de los estilos visuales.</p> <ul style="list-style-type: none"> - Fundamentos de CSS, modelos de caja, etc - Frameworks CSS (Bootstrap) - Metodologías BEM - Preprocesadores CSS - Responsividad 	18
3	Programación Avanzada Javascript	<p>A lo largo del módulo, el participante aprenderá:</p> <ul style="list-style-type: none"> ● Evolución ECMA Script ● Javascript Object Notation ● Codificar algoritmos con estructuras de datos ● Patrones de programación Javascript ● Programación funcional Javascript ● Orientación a objetos Javascript ● Programación asíncrona Javascript 	24
4	Desarrollo de interfaces interactivas con Framework React	<p>Al finalizar este módulo, el participante será capaz de construir una aplicación web orientada a componentes utilizando la librería React</p> <ul style="list-style-type: none"> - Concepto general - Ciclo de Vida - Componentes - Comunicación entre componentes - Frameworks de Componentes Visuales - Iconos - Formularios - Charts 	18

5	Desarrollo de aplicaciones Front-End con Framework React	<p>Al finalizar este módulo, el participante será capaz de construir una aplicación web con orientación a componentes utilizando librerías avanzadas React y prácticas de testing.</p> <ul style="list-style-type: none"> - API Fetch / Axios - Web Socket - Estados de la APP - Seguridad (OAuth, JWT) - ESLint - Test 	24
6	Desarrollo de aplicaciones web progresivas (PWA)	Al finalizar este módulo el participante conocerá el concepto de PWA y podrá implementar una aplicación mobile con dicha tecnología utilizando el entorno React.	12
7	Fundamentos de Desarrollo Agile	En este módulo, el participante adquiere los aprendizajes fundamentales de las metodologías ágiles y profundiza en el rol del desarrollador en un equipo Scrum, conociendo sus roles, ceremonias y prácticas.	12
8	Fundamentos de Integración Continua	En este módulo, el participante adquiere los aprendizajes fundamentales del pensamiento DevOps, en donde conocerá los procesos, técnicas y herramientas para la implementación de un circuito de integración continua, de forma que pueda integrarse en una célula y ejercer de mejor forma su rol de desarrollador	15
9	Portafolio	<p>Módulo en donde el participante prepara y presenta su portafolio de trabajos que preparó a lo largo de todo el curso.</p> <p>-</p>	6

Módulos Transversales

Módulo Formativo N°1	
Nombre	Orientación al Perfil y Metodología del Curso
N° de horas	3 horas
Requisitos de ingreso	Experiencia Laboral; Test de Selección;
Competencia del módulo	Identificar las competencias y habilidades del perfil profesional en junto al rol que poseen dentro del contexto de la industria TI, así como la metodología de trabajo del curso para adquirirlas

Aprendizajes esperados	Criterios de evaluación	Contenidos
1. Reconocer las características fundamentales del perfil profesional en cuestión, así como el contexto laboral en el cual se desenvuelve	1.1 Identifica las competencias técnicas y personales valoradas por la industria TI 1.2 Reconoce las competencias laborales del perfil en cuestión 1.3 Reconoce las características del contexto de la industria y del mercado laboral en el cual se desenvuelve el perfil en cuestión	La industria TI <ul style="list-style-type: none"> • Características de la industria • Perfiles más comunes en la industria TI • Competencias técnicas valoradas por la industria TI • Habilidades personales valoradas por la industria TI • Metodologías y forma de trabajo del área • El mercado laboral TI, estudios de sueldo El perfil profesional asociado al curso (*) <ul style="list-style-type: none"> • Qué es un perfil profesional • Competencias que posee el perfil • Habilidades que posee el perfil • Niveles de experiencia y seniority del perfil • Expectativas laborales del mercado actual para el perfil

		<ul style="list-style-type: none"> Proyección laboral del perfil <p>(*) Considerar el perfil específico del plan formativo a ejecutar</p>
<p>2. Distinguir la naturaleza de cada módulo que forma parte de la currícula así como las características del trabajo a desempeñar, productos obtenidos, y su contribución a la formación del perfil en cuestión</p>	<p>2.1 Distingue las competencias que serán adquiridas a lo largo de cada módulo de la currícula</p> <p>2.2 Reconoce la naturaleza del trabajo que será realizado junto con las herramientas que serán utilizadas a lo largo de cada módulo</p> <p>2.3 Reconoce la importancia de un portafolio de producto así como sus características para la formación de una identidad profesional</p> <p>2.4 Identifica los productos que serán obtenidos en cada módulo así como su contribución al portafolio de producto</p>	<p>Currícula del Curso</p> <ul style="list-style-type: none"> Módulos y competencias a formar a lo largo de la currícula Herramientas a utilizar durante el curso Características del trabajo técnico a realizar en cada módulo Productos obtenidos en cada módulo <p>Portafolio de Producto</p> <ul style="list-style-type: none"> Qué es un portafolio de producto Importancia de un portafolio de producto en la identidad profesional Contribución de cada módulo del curso al portafolio
<p>3. Comprender la metodología de trabajo a lo largo del curso distinguiendo la importancia del trabajo en equipo y el trabajo ético para contribuir al logro de las competencias del plan formativo</p>	<p>3.1 Distingue las competencias que serán adquiridas a lo largo de cada módulo de la currícula</p> <p>3.2 Reconoce la naturaleza del trabajo que será realizado junto con las herramientas que serán utilizadas a lo largo de cada módulo</p>	<p>Metodología de enseñanza-aprendizaje</p> <ul style="list-style-type: none"> Qué es el aprendizaje Activo Metodologías utilizadas a lo largo del curso Rol del facilitador y del participante <p>Herramientas a utilizar a lo largo del Curso</p> <ul style="list-style-type: none"> Herramientas de gestión del proceso de aprendizaje (LMS)

	<p>3.3 Reconoce la importancia de un portafolio de producto así como sus características para la formación de una identidad profesional</p> <p>3.4 Identifica los productos que serán obtenidos en cada módulo así como su contribución al portafolio de producto</p>	<ul style="list-style-type: none"> • Herramientas de coordinación y trabajo colaborativo • Herramientas propias de la competencia técnica <p>Habilidades requeridas a lo largo del Curso</p> <ul style="list-style-type: none"> • Trabajo en Equipo • Autoaprendizaje • Tolerancia a la frustración • Comportamiento ético <ul style="list-style-type: none"> ○ Importancia del comportamiento ético en la industria TI ○ Código de ética del curso
--	---	---

Estrategias Metodológicas para la implementación del módulo

El presente módulo tiene por objetivo dar una orientación fundamental al participante acerca del contexto general del presente plan formativo, de modo que pueda comprender el contexto de la industria TI en el cual está inserto, así como la metodología de trabajo que se empleará para el desarrollo de los aprendizajes.

En lo que respecta al primer aprendizaje relacionado con el perfil en cuestión y el contexto de la industria TI, se sugiere, entre otras cosas, la incorporación de actividades de charlas, paneles o conversatorios con profesionales referentes de la industria TI de forma de otorgar una mirada complementaria a la de los instructores del plan formativo, pudiendo ser profesionales que ocupan roles distintos tales como head hunters, managers, arquitectos, developers, entre otros. También se sugiere la incorporación de recursos de aprendizaje de tipo lecturas y video-lecturas, entre otros.

En el segundo aprendizaje, se sugiere diseñar actividades que disminuyan la componente expositiva y que inviten al participante a explorar el plan formativo, así como también buscar información en internet para posteriormente generar un espacio de discusión y de elaboración de conclusiones. Es muy importante que se haga hincapié en que en este módulo inicial se abre el portafolio que se irá complementando a lo largo de los módulos para ser finalizado en el módulo correspondiente a Portafolio de Producto. En este módulo el participante ya debería comprender que debe buscar una idea de proyecto para ir desarrollando a lo largo del curso.

En el tercer aprendizaje, es importante que el participante experimente y conozca las herramientas que utilizará a lo largo del curso. Se sugiere que durante este módulo se diseñen actividades que requieran el uso de la mayor parte de las herramientas y funcionalidades del sistema de gestión de aprendizajes (LMS), de las herramientas de trabajo colaborativo (por ejemplo, Teams, Trello, Canvas, u otros).

Como buena práctica, se sugiere la incorporación de actividades que fomente la interacción y conocimiento entre los participantes, de modo de facilitar la organización de grupos de estudio y desarrollo de comunidad, los cuales son elementos claves para enfrentar de mejor forma el bootcamp. Asimismo, se pueden incorporar dinámicas de trabajo inspiradas en las metodologías ágiles.

También se sugiere complementar las actividades de este aprendizaje con charlas, paneles o conversatorios en donde se incorporen participantes que ya han egresado de cursos similares con metodología bootcamp. Asimismo, se pueden incorporar videos con testimonios o que den una idea de la metodología.

Estrategia Evaluativa del módulo

La estrategia debe centrarse en la evaluación del Proceso, mediante la observación de la participación y desempeño del participante en las dinámicas de grupo, actividades, juegos de rol, etc, para lo cual pueden utilizarse instrumentos tales como rúbricas o escalas de valoración, o incluso actividades de autoevaluación.

Perfil del Facilitador

Características Transversales

El instructor debe poseer competencias pedagógicas tales como:

- Planificar la clase
- Demostrar dominio de los temas tratados
- Capacidad de motivar
- Capacidad de contextualizar los temas en el área laboral
- Capacidad de cumplir un rol de facilitador
- Habilidades comunicacionales acordes a la función

Características Específicas

- Formación profesional en el área de las ingenierías, como diseñador para el caso del plan UX/UI. En todos los casos anteriores, con título
- Experiencia laboral de mínimo 3 años desempeñando funciones de laborales ya sea como desarrollador, científico de datos, o diseñador UX/UI, dependiendo del plan formativo, o bien gerenciando áreas en estas materias
- Experiencia como facilitador/a de capacitación para personas adultas, y/o bien como docente en instituciones de educación superior, sean universidades, institutos profesionales o centros de formación técnica, de mínimo 100 horas cronológicas, demostrable

Recursos materiales para la implementación del módulo formativo

Infraestructura	Equipos y Herramientas	Materiales e Insumos
<ul style="list-style-type: none"> • Laboratorio computacional que cuente al menos con 1,5 m2 por participante, implementada con: Puestos de trabajo individuales que considere mesa y silla. Escritorio y silla para el facilitador. • Conexiones para utilizar medios didácticos tales como data y salida a internet. • Sistema de iluminación y ventilación adecuados • Servicios higiénicos separados para hombres y mujeres en recintos de aulas y de actividades prácticas. 	<ul style="list-style-type: none"> • Proyector multimedia • Notebook o PC, para el facilitador • Telón • Pizarra • Equipo computacional, uno para cada alumno, al menos Pentium Core i5, 8 GB RAM y 500 GB HD, y conexión a internet, con el siguiente software instalado: 	<ul style="list-style-type: none"> • Set de oficina, uno por participante, compuesto por: Carpeta o archivador. Cuaderno o croquera. Lápiz pasta. Lápiz grafito. Goma de borrar. Liquido corrector. Regla. • Plumones para pizarrón. • Lista de participantes. • Carpeta de registro de evidencias para el facilitador • Material audiovisual que contenga información del curso: • Manual didáctico que contemple todos los contenidos especificados para este módulo, por cada participante.

Especialización Desarrollo Front-End

Módulo Formativo N° 01	
Nombre	Desarrollo interfaz usuario web
N° de horas	18 horas
Requisitos de ingreso	Título técnico o profesional; Experiencia Laboral. Test de Conocimientos: <ul style="list-style-type: none"> • HTML Básico • CSS Básico • Javascript Básico
Competencia del módulo	<ul style="list-style-type: none"> • Implementar interfaces de usuario web responsivas utilizando html5 y frameworks de estilos siguiendo las buenas prácticas de la industria

Aprendizajes esperados	Criterios de evaluación	Contenidos
1. Describir las características del proceso de diseño e implementación de un producto digital distinguiendo el rol del desarrollador front-end y las buenas prácticas para la organización de estilos	1.1 Distingue el rol de un desarrollador front-end y de un diseñador UX/UI en el proceso de desarrollo de un producto digital 1.2 Describe el rol y la importancia de las metodologías para la organización y modularización de estilos en una página web	El proceso de desarrollo de un producto visual <ul style="list-style-type: none"> • Diferencias entre Front-End, Back-End y Full-Stack • Aplicaciones SPA v/s monolitos • Renderización cliente y servidor • El proceso de ideación de un producto digital • El proceso de implementación de un producto digital • El rol del diseñador UX/UI • El rol del desarrollador Front-End

	<p>1.3 Describe el rol y la importancia de un preprocesador para el desarrollo front</p>	<ul style="list-style-type: none"> • Importancia de las guías de estilos y representaciones visuales en la maquetación <p>El lenguaje de Marcas</p> <ul style="list-style-type: none"> • Qué es lenguaje de marcas • Qué es el DOM • Acerca de HTML4 y anteriores • HTML5 y sus diferencias sobre HTML tradicional • HTML5 en dispositivos móviles • Qué es XHTML y cómo se usa <p>Metodologías para la organización y modularización de estilos</p> <ul style="list-style-type: none"> • Ventajas de utilizar una metodología • Metodologías más utilizadas (BEM, OOCSS, SMACCS) <p>Preprocesadores CSS</p> <ul style="list-style-type: none"> • Qué es un procesador CSS • Importancia del preprocesador en el desarrollo front • Preprocesadores más utilizados (SASS, LESS)
<p>2. Describir conceptos avanzados relacionados con el desarrollo web front-end</p>	<p>2.1 Describe las características principales y aplicaciones del protocolo HTTP en el desarrollo web</p> <p>2.2 Reconoce la utilidad de un CDN para el manejo de contenidos estáticos en un aplicativo web</p>	<p>Conceptos avanzados del desarrollo Web</p> <ul style="list-style-type: none"> • Protocolo HTTP <ul style="list-style-type: none"> o Qué es el protocolo HTTP o Dónde se utiliza el protocolo HTTP o Entendiendo los métodos y sus usos o Qué se entiende por Endpoint o Interpretando los códigos de retorno • Qué es un CDN y cómo se usa • Implementando la seguridad con HTTPS

	<p>2.3 Describe la utilidad de cookies en un aplicativo web</p> <p>2.4 Reconoce el concepto de almacenamiento local de datos distinguiendo entre local store y session storage</p>	<ul style="list-style-type: none"> ● Entendiendo el Encoding de páginas ● Acerca de Cookies y su uso en los navegadores ● Compatibilidad del desarrollo web en los distintos Navegadores ● Entendiendo el LocalStorage y SessionStorage como almacenaje local de los datos en un navegador ● Aplicaciones de consola. Otra forma de consumir aplicaciones Web (Lynx, Wget, Curl)
<p>3. Aplicar hojas de estilo CSS básicas distinguiendo elementos de responsividad para personalizar la presentación de un documento HTML acorde a un requerimiento entregado</p>	<p>3.1 Reconoce los principios y usos de las hojas de estilo CSS para el manejo de los aspectos visuales básicos de un documento HTML</p> <p>3.2 Utiliza rutas absolutas y relativas para el manejo de assets e imágenes en la incorporación de hojas de estilo al documento HTML</p> <p>3.3 Codifica un documento HTML utilizando la sintaxis y reglas de estilos CSS para modificar aspectos visuales y resolver un problema planteado acorde a las buenas prácticas de la industria</p> <p>3.4 Identifica los conceptos claves de la responsividad de un reconociendo los mecanismos para implementarla en un documento html</p> <p>3.5 Utiliza las herramientas para desarrolladores provista por el navegador</p>	<p>Hojas de estilo</p> <ul style="list-style-type: none"> ● Acerca de CSS y su uso en la interfaz de usuario Web ● Ventajas de usar CSS ● Desde CSS1 hasta CSS3 ● Aspectos avanzados de CSS <ul style="list-style-type: none"> ○ Patrones ○ Bloques de declaración ○ Uso de estilos ○ Fuentes ○ Especificidad ○ Herencia ○ Posicionamiento ● Limitaciones de CSS ● Soporte en distintos navegadores web ● CSS4 y el futuro de las hojas de estilo <p>Responsividad</p> <ul style="list-style-type: none"> ● Qué es Responsividad ● Por qué desarrollar aplicaciones web Responsivas ● Impacto de la responsividad en la experiencia de usuario ● Cómo funciona la adaptabilidad a distintos anchos de pantalla

	para la inspección de los estilos aplicados en el documento	<ul style="list-style-type: none"> • Diseño Web Responsivo vs Diseño Adaptativo • Cómo crear aplicaciones web responsivas • Los bloques de construcción del Diseño Web Responsivo • Puntos de ruptura Responsivas comunes • Unidades y valores del CSS para el diseño responsivo
4. Implementar una interfaz web utilizando el modelo de cajas acorde a las especificaciones de un prototipo	<p>4.1 Distingue los elementos principales para el diagramado y posicionamiento de elementos en una interfaz web</p> <p>4.2 Utiliza el modelo de cajas para la organización de elementos en una interfaz web de acuerdo a lo requerido</p> <p>4.3 Utiliza propiedades de posicionamiento y visualización para la organización de elementos en una interfaz web de acuerdo a lo requerido</p> <p>4.4 Implementa distintos tipos de layout para la organización de una interfaz web de acuerdo a lo requerido</p>	<p>Modelo de cajas</p> <ul style="list-style-type: none"> • Qué es el modelo de cajas. ¿Existen otros modelos? • Propiedades que componen el modelo de cajas. • Tipos de cajas. Diferencias entre elementos de bloque y elementos de línea. • Inspeccionando elementos con navegador para identificar las cajas. <p>Posicionamiento de elementos y visualización</p> <ul style="list-style-type: none"> • Conceptos básicos de propiedades de posicionamiento. • Tipo de Posicionamiento (normal, relativo, absoluto, fijo, flotante) • Visualización <p>Layout</p> <ul style="list-style-type: none"> • Qué es un layout • Tipos de layout (fluido, fijo, elástico, absoluto) • Ventajas y desventajas
5. Implementar un preprocesamiento CSS para la organización y modularización de estilos en una	5.1 Reconoce las características principales del preprocesador SASS para la organización y modularización de estilos	<p>El Preprocesador SASS</p> <ul style="list-style-type: none"> • Qué es Sass y por qué utilizarlo • Conocer Sass y aprovechar las ventajas en el proceso de construcción de un sitio web.

<p>interfaz web acorde a las buenas prácticas</p>	<p>5.2 Utiliza variables para la reutilización de código CSS</p> <p>5.3 Implementa estilos en una interfaz web de baja complejidad utilizando preprocesador SASS</p>	<ul style="list-style-type: none"> ● Instalar Sass y linters para editores de código ● Conocer patrón 7-1 ● Sintaxis, flujo de trabajo y buenas prácticas usando Sass <ul style="list-style-type: none"> ○ Utilización de Sass desde línea de comandos. ○ Instalación de plugins de Sass en editores de texto para automatización de tareas. ○ Uso de variables para reutilización de código. ○ Elementos anidados y namespaces. ○ Manejo de parciales e imports ○ Manejo de mixins e includes
<p>6. Implementar una metodología para la organización y modularización de estilos en una interfaz web acorde a las buenas prácticas</p>	<p>6.1 Explica las características de la modularización de estilos y las ventajas de la utilización de una metodología de modularización</p> <p>6.2 Describe las características principales de la metodología BEM para la organización y modularización de estilos</p> <p>6.3 Describe las características principales de la metodología OOCSS para la organización y modularización de estilos</p> <p>6.4 Describe las características principales de la metodología SMACSS para la organización y modularización de estilos</p> <p>6.5 Implementar una interfaz web de baja complejidad utilizando metodologías para la organización y modularización de estilos</p>	<p>Modularización de estilos</p> <ul style="list-style-type: none"> ● Por qué modularizar ● Ventajas de modularizar los estilos ● Utilización de metodologías de modularización ● Arquitectura CSS <p>Metodologías de modularización</p> <ul style="list-style-type: none"> ● Qué es una metodología de modularización ● Para qué sirve una metodología ● Ventajas de usar una metodología ● Identificando algunas metodologías y sus características (BEM, OOCSS, SMACSS, ITCSS) <p>Metodología BEM</p> <ul style="list-style-type: none"> ● En qué consiste esta metodología ● Conceptos claves de BEM ● Convención de nombres ● Cómo estructurar un proyecto con BEM <p>Metodología OOCSS</p>

		<ul style="list-style-type: none"> • En qué consiste esta metodología • Principios básicos de OOCSS <p>Metodología SMACCS</p> <ul style="list-style-type: none"> • En qué consiste esta metodología • Categorías de elementos SMACCS
7. Implementar un sitio web básico responsivo utilizando framework Bootstrap para organizar la presentación de un documento html	<p>7.1 Describe las características y beneficios de utilización del framework Bootstrap para el manejo de estilos en una página web</p> <p>7.2 Reconoce los elementos y estilos principales de Bootstrap para el manejo de estilos en una página web</p> <p>7.3 Utiliza estilos disponibles en el framework Bootstrap para organizar los elementos visuales en un documento HTML y resolver el problema planteado</p>	<p>Frameworks CSS</p> <ul style="list-style-type: none"> • Qué es un Framework CSS, por qué, cuándo utilizarlo • Alternativas de frameworks basados en CSS (Bootstrap, Foundation, Pure CSS, Bulma) • Ventajas de usar un Framework CSS vs HTML tradicional • Componiendo estilos propios <p>Analizando un framework CSS: Bootstrap</p> <ul style="list-style-type: none"> • Qué es Bootstrap • Ventajas • Filosofía Bootstrap <ul style="list-style-type: none"> o Responsividad o Principio Mobile First • Modificar y extender funcionalidad de Bootstrap con Sass • Conociendo layouts, contenidos y componentes de Bootstrap.: <ul style="list-style-type: none"> o Grillas o Botones o Paneles o Tablas o Formularios o Navegación y menús

		<ul style="list-style-type: none"> • Conociendo componentes JavaScript <ul style="list-style-type: none"> o Alertas o Breadcrumb o Card o Carrusel o Tooltip o Modal o Popover
--	--	---

Estrategias metodológicas para la implementación del módulo

Para la implementación de dicho módulo, se sugiere la aplicación de una metodología de aprendizaje basado en problemas o bien una metodología de proyectos, propiciando de esta forma, el “aprender haciendo”. Asimismo, se sugiere realizar actividades en donde se propicien prácticas como la “programación de pares” o bien en donde se produzca interacción y discusión entre los participantes.

El facilitador realiza una clase expositiva en donde brevemente se exponen los objetivos de la sesión y la explicación de los conceptos de forma resumida. De la misma forma, realiza una demostración práctica de los conceptos para luego realizar actividades prácticas de trabajo basado en un problema. Para finalizar, se realiza una autoevaluación o coevaluación entre los participantes o un análisis grupal del problema elaborando conclusiones de la sesión. También se sugiere el trabajo en grupos pequeños, por ejemplo de pares (pair programming), de esta forma se facilitan los atascos y se produce el intercambio de aprendizaje entre los participantes. Es importante que las demostraciones realizadas por el facilitador no solamente considere la resolución del problema sino que también cometa errores de forma intencionada para que el participante tenga la posibilidad ver cómo enfrentar una situación de atasco y cuál es el raciocinio lógico empleado para buscar la solución, comprendiendo así que el proceso de construcción de una pieza de software no siempre resulta a la primera sino que es un proceso en donde gradualmente se acerca a una solución aceptable.

Los problemas propuestos en cada sesión deben ser contextualizados en situaciones reales, preferentemente de la industria del retail o de la banca, para de esta forma construir un aprendizaje significativo y contextualizado, es decir, que los conectan con experiencias que han vivido como clientes o consumidores. Por ejemplo, se pueden utilizar ejemplos de puntos de venta, carros de compra, back office de mantención de datos, cuentas corrientes, etc.

Desarrollo de conocimientos

Es fundamental en este módulo una debida planificación de las actividades para lograr el mejor resultado. Si bien es cierto que se declaran contenidos en extenso, los criterios de evaluación son más bien genéricos, permitiendo de esta manera, al facilitador ir tomando decisiones sobre el nivel de profundidad en que se abordarán los contenidos de acuerdo al desempeño que va observando en el grupo de participantes. Sí es importante, que al menos ciertos contenidos sean mencionados para que al menos el participante conozca de su existencia y tenga la posibilidad de profundizarlos por iniciativa personal.

Desarrollo de habilidades de empleabilidad

Para promover las habilidades conductuales y/o actitudes de tolerancia a la frustración y de autoaprendizaje, se debe integrar como parte de la dinámica de trabajo o de aprendizaje, que los participantes busquen información en internet o bien discutan con sus compañeros cuando se encuentren con trabas para solucionar el problema planteado. Asimismo, propiciar que el participante utilice el método de prueba y error al enfrentar trabas, antes de entregarles la solución.

Estrategia Evaluativa del módulo

La evaluación del módulo considerará una o varias ejecuciones prácticas, individuales o grupales, durante o al final éste, en donde el participante deberá entregar un producto a partir de un caso planteado. Se deben diseñar las respectivas escalas de valoración o rúbricas que incluya los aspectos relevantes a medir y consistentes con los criterios de evaluación definidos en el módulo.

También se sugiere incorporar autoevaluaciones y coevaluaciones formativas durante el transcurso del módulo.

Facilitador

Características Transversales

El instructor debe poseer competencias pedagógicas tales como:

- Planificar la clase
- Demostrar dominio de los temas tratados
- Capacidad de motivar
- Capacidad de contextualizar los temas en el área laboral
- Capacidad de cumplir un rol de facilitador
- Habilidades comunicacionales acordes a la función

Características Específicas

- Formación académica en el área de las ingenierías, con título
- Experiencia laboral de mínimo 3 años desempeñando funciones de desarrollo web, demostrable
- Experiencia como facilitador/a de capacitación para personas adultas, y/o bien como docente en instituciones de educación superior, sean universidades, institutos profesionales o centros de formación técnica, de mínimo 100 horas cronológicas, demostrable

Recursos materiales para la implementación del módulo formativo

Infraestructura	Equipos y Herramientas	Materiales e Insumos
<ul style="list-style-type: none"> • Sala de clases implementada con: Puestos de trabajo individuales que considere mesa y silla. Escritorio y silla para el facilitador. • Conexiones para utilizar medios didácticos tales como data y salida a internet y salida de audio • Sistema de iluminación y ventilación adecuados • Servicios higiénicos separados para hombres y mujeres en recintos de aulas y de actividades prácticas. 	<ul style="list-style-type: none"> • Proyector multimedia • Notebook o PC, para el facilitador • Telón • Pizarra • Equipo computacional para cada participante: Mac o PC con 8 de ram mínimo, procesador Intel i5 mínimo • Software instalado: <ul style="list-style-type: none"> ○ Navegador ○ Visual Studio Code https://code.visualstudio.com/download ○ Sass https://sass-lang.com/install ○ NPM https://www.npmjs.com/get-npm 	<ul style="list-style-type: none"> • Plumones para pizarrón. • Lista de participantes. • Carpeta de registro de evidencias para el facilitador • Material audiovisual que contenga información del curso: • Manual didáctico que contemple todos los contenidos especificados para este módulo, por cada participante.

Especialidad Desarrollo Front-End

Módulo Formativo N° 02	
Nombre	Programación Avanzada Javascript
N° de horas	24 horas
Requisitos de ingreso	Carrera Técnica o Profesional; Experiencia Laboral TI; Test de Selección, en donde se mide: <ul style="list-style-type: none"> Conocimientos de Fundamentos de Programación en Javascript
Competencia del módulo	<ul style="list-style-type: none"> Codificar piezas de software de mediana complejidad en lenguaje Javascript utilizando paradigmas de orientación a objetos, orientación a eventos y programación asíncrona para resolver problemáticas de acuerdo a las necesidades de la industria

Aprendizajes esperados	Criterios de evaluación	Contenidos
1. Describir las características principales del lenguaje Javascript y sus particularidades para el desarrollo de aplicaciones web	1.1 Reconoce conceptos generales Javascript distinguiendo características propias del lenguaje versus otros 1.2 Reconoce los stacks y herramientas más utilizadas basados en Javascript para el desarrollo de aplicaciones 1.3 Describe la evolución del lenguaje Javascript, las distintas versiones ECMA, sus características y potencialidades	Generalidades del lenguaje Javascript <ul style="list-style-type: none"> Historia de Javascript Uso en navegadores web Entornos virtuales de JS Diferencia entre Javascript y otros lenguajes Paradigmas de programación soportados por JavaScript Fortalezas del lenguaje y forma correcta de uso Javascript como lenguaje asíncrono Sintaxis e Indentación

	<p>1.4 Reconoce las características fundamentales de TypeScript y su uso en el desarrollo de aplicaciones</p>	<ul style="list-style-type: none"> • Lenguaje interpretado v/s compilado <p>Evolución del lenguaje</p> <ul style="list-style-type: none"> • El estándar ECMA Script • Javascript v/s ECMA Script • Evolución de ECMA Script: desde ES3 a ES9 • Acerca de TypeScript y sus características • ¿Dónde y cómo se aplica TypeScript? <p>El stack Javascript</p> <ul style="list-style-type: none"> • Qué es un stack de desarrollo • Stacks basados en Javascript (MEAN, MERN) • Aplicaciones para Frontend • Aplicaciones para Backend • Frameworks basados en Javascript • Acceso a datos con Javascript • Lenguajes derivados de Javascript • Utilitarios • El GC en javascript
<p>2. Codificar algoritmos utilizando la sintaxis del lenguaje Javascript para la resolución de un problema de baja complejidad</p>	<p>2.1 Reconoce las características generales del entorno de ejecución Javascript para el desarrollo de aplicaciones</p> <p>2.2 Reconoce tipos de datos, asignación y scope de variables para la codificación de una rutina en lenguaje Javascript</p>	<p>Entorno de ejecución Javascript</p> <ul style="list-style-type: none"> • Entorno de ejecución • La consola de comandos • Setting de variables • console.log() • Estructura de un programa javascript <ul style="list-style-type: none"> ○ Import ○ Require • El event loop <ul style="list-style-type: none"> ○ Stack

	<p>2.3 Utiliza expresiones lógicas, sentencias condicionales y ciclos y anidaciones en un programa estructurado</p> <p>2.4 Codifica algoritmos utilizando arreglos para dar solución a un problema acorde al lenguaje Javascript</p>	<ul style="list-style-type: none"> ○ Heap ○ Queue <p>Desarrollo Aplicaciones Javascript</p> <ul style="list-style-type: none"> ● Variables ● Scope de variables ● Hoisting ● var v/s let v/s const ● Tipos de datos primitivos ● Operadores ● Diferencia entre null y undefined ● El operador typeof ● El comparador == v/s === ● El tipado en javascript ● Instrucciones condicionales ● Ciclos e iteraciones ● Ciclos y condiciones anidados <ul style="list-style-type: none"> ○ Break ○ Continue ● try/catch y manejo de errores ● Expresiones regulares ● Arreglos ● Debugger ● Manipulación de archivos
3. Reconocer las características de la Notación de Objetos JavaScript (JSON) como estructura de datos y	3.1 Reconoce las características de la notación de objetos JSON y su utilización como formato de intercambio de información	<p>Notación de Objetos Javascript</p> <ul style="list-style-type: none"> ● Objetos JSON ● Sintaxis y Notación ● Tipos de datos en objetos JSON

<p>estándar de intercambio de información en las aplicaciones</p>	<p>3.2 Estructura un objeto JSON utilizando la sintaxis y notación apropiada para la representación de datos que resuelven un problema</p> <p>3.3 Realiza operaciones de manipulación de objetos JSON acorde al lenguaje Javascript para resolver un problema</p>	<ul style="list-style-type: none"> • Objetos anidados • Funciones dentro de objetos • Asignación por deestructuración • Operaciones con objetos JSON <ul style="list-style-type: none"> ○ Clonación ○ Merge ○ Recorrido, Parse y Stringify • JSON como base para el protocolo API REST • Otros formatos de intercambio de información <ul style="list-style-type: none"> ○ El lenguaje YAML ○ Comparación con XML
<p>4. Distinguir estructuras de datos requeridas para la resolución de un problema y su implementación con Javascript</p>	<p>4.1 Describe las características de las estructuras de datos en la resolución de problemas de programación</p> <p>4.2 Selecciona las estructuras de dato adecuadas para la resolución de un problema de programación</p> <p>4.3 Implementa estructuras de datos utilizando lenguaje Javascript para la resolución de un problema</p>	<p>Estructuras de Dato en Java</p> <ul style="list-style-type: none"> • Qué es una estructura de datos • Arreglos de Datos <ul style="list-style-type: none"> ○ Qué es un arreglo y sus características ○ Tipos (vectores, matrices) ○ Operaciones sobre arreglos ○ Implementación Javascript • Listas <ul style="list-style-type: none"> ○ Qué es una Lista y sus características ○ Tipos (simples, dobles, circulares) ○ Operaciones sobre listas ○ Implementación Javascript • Pilas <ul style="list-style-type: none"> ○ Qué es una Pila y sus características ○ Operaciones sobre pilas ○ Implementación Javascript • Colas <ul style="list-style-type: none"> ○ Qué es una Cola y sus características

		<ul style="list-style-type: none"> ○ Operaciones sobre colas ○ Implementación Javascript ● Árboles <ul style="list-style-type: none"> ○ Qué es un Árbol y sus características ○ Tipos (binarios, ordenados, multicamino) ○ Operaciones sobre árboles ○ Implementación en Javascript ● Conjuntos <ul style="list-style-type: none"> ○ Qué es un Conjunto y sus características ○ Operaciones sobre conjuntos ○ Implementación en Javascript ● Grafos <ul style="list-style-type: none"> ○ Qué es un Grafo y sus características ○ Operaciones sobre grafos ○ Implementación en Javascript
5. Codificar algoritmos de mediana complejidad utilizando estructuras de datos para dar solución a un problema acorde al lenguaje Javascript	<p>5.1 Reconoce el concepto de eficiencia y complejidad ciclométrica que posee un algoritmo y sus implicancias en el desarrollo de piezas de software</p> <p>5.2 Reconoce problemas comunes de desarrollo de software y aplicar algoritmos adecuados para su resolución en lenguaje Javascript</p> <p>5.3 Codifica un algoritmo utilizando estructuras de datos apropiadas para resolver un problema de mediana complejidad</p>	<p>Programación de Algoritmos</p> <ul style="list-style-type: none"> ● Por qué estudiar algoritmos ● Eficiencia y Complejidad de un algoritmo <ul style="list-style-type: none"> ○ Complejidad Ciclométrica ○ Big-O ● Algoritmos Recursivos ● Algoritmos de Ordenamiento ● Algoritmos de Búsqueda ● Implementación en Javascript ● Técnicas de Debugging ● Librerías que facilitan el trabajo con estructuras de datos (underscore, lodash, moment) <p>Utilitarios basados en javascript</p> <ul style="list-style-type: none"> ● moment

	5.4 Incorporar y aplicar conceptos básicos de algoritmos y estructuras de datos en programas Javascript	<ul style="list-style-type: none"> ● underscore ● socket ● axios ● lodash
6. Reconocer los principales patrones de diseño utilizados en lenguaje Javascript para la resolución de problemas habituales	<p>6.1 Reconoce la utilidad de los patrones de diseño para la construcción de software mantenible y de calidad</p> <p>6.2 Reconoce las características de los patrones creacionales, estructurales y de comportamiento utilizados comúnmente en Javascript</p> <p>6.3 Distingue los principales patrones creacionales y sus características para el desarrollo de piezas en lenguaje Javascript</p> <p>6.4 Distingue los principales patrones estructurales y sus características para el desarrollo de piezas en lenguaje Javascript</p> <p>6.5 Distingue los principales patrones comportamiento y sus características para el desarrollo de piezas en lenguaje Javascript</p>	<p>Patrones de diseño</p> <ul style="list-style-type: none"> ● Qué es un patrón de diseño ● Utilidad de los patrones de diseño ● Patrones creacionales <ul style="list-style-type: none"> ○ Patrón Constructor ○ Patrón Módulo ○ Patrón Módulo revelador ○ Patrón Prototipo ○ Patrón Singleton ● Patrones estructurales <ul style="list-style-type: none"> ○ Patrón Mixin ○ Patrón Decorador ○ Patrón Facade ○ Patrón adaptador ○ Patrón proxy ● Patrones de comportamiento <ul style="list-style-type: none"> ○ Patrón observador ○ Patrón mediado ○ Patrón comando ○ Patrón cadena de responsabilidad ○ Patrón iterador ○ Patrón state

<p>7. Utilizar aspectos de la programación funcional en el entorno Javascript para la resolución de un problema</p>	<p>7.1 Explica el concepto de programación funcional, sus principales características y la forma en que es implementado en Javascript</p> <p>7.2 Codifica rutinas de Javascript utilizando elementos de la programación funcional para dar solución a un problema</p>	<p>Programación funcional</p> <ul style="list-style-type: none"> ● El paradigma de programación Funcional ● Enfoque imperativo v/s declarativo ● Funciones como Objetos de primera clase ● Funciones y procedimientos <ul style="list-style-type: none"> ○ Invocación de funciones <ul style="list-style-type: none"> ○ El paso de argumentos y retorno ○ La función flecha () => {} ○ Funciones dentro de objetos ○ Variables funciones ● Recursión en programación funcional ● "Currying" en programación funcional ● Composición de funciones
<p>8. Aplicar el paradigma de la orientación a Eventos y programación asíncrona del lenguaje Javascript para la codificación de rutinas que resuelven un problema</p>	<p>8.1 Explica el paradigma de programación orientada a eventos, sus principales características y su implementación en Javascript</p> <p>8.2 Codifica rutinas con comportamiento asíncrono utilizando callbacks para la resolución de un problema</p> <p>8.3 Codifica rutinas asíncronas utilizando sentencias ASYNC/AWAIT para la resolución de un problema</p> <p>8.4 Codifica rutinas asíncronas utilizando sentencias Promise/Resolve/Reject epara la resolución de un problema</p>	<p>Programación orientada a Eventos</p> <ul style="list-style-type: none"> ● ¿Qué es un evento? ● Paradigma de la orientación a eventos ● Captura eventos (Listener) ● Disparar Eventos ● Eventos en los navegadores Web <p>Programación asíncrona</p> <ul style="list-style-type: none"> ● Programación síncrona v/s asíncrona ● Blocking v/s non-blocking ● Creación de Callbacks ● Invocación de Callbacks ● Creación de funciones ASYNC/AWAIT ● Utilizar Promise/Resolve/Reject ● Utilizar Catch/Throw y errores personalizados en llamadas asíncronas

9. Reconocer los aspectos fundamentales de la programación orientada a objetos y los mecanismos de implementación en Javascript	<p>9.1 Describe las características fundamentales del paradigma de orientación a objetos y su implementación en Javascript</p> <p>9.2 Distingue los enfoques de orientación a objeto que han sido implementados con las distintas versiones de Javascript</p> <p>9.3 Codifica rutinas utilizando objetos para la resolución un problema de acuerdo al lenguaje Javascript</p> <p>9.4 Codifica rutinas definiendo clases y creando objetos para la resolución un problema de acuerdo a ECMA 2015</p>	<p>Programación orientada a objetos</p> <ul style="list-style-type: none"> • En qué consiste el paradigma de programación orientada a objetos • Principios de la POO <ul style="list-style-type: none"> ○ Encapsulación ○ Abstracción ○ Herencia ○ Polimorfismo • Implementación de la POO en Javascript <ul style="list-style-type: none"> ○ Objetos Literales ○ Método Create ○ Función Constructor ○ Herencia por Prototipos • Implementación de la POO con ECMA 2015 <ul style="list-style-type: none"> ○ Definición de clases ○ Definición de subclases ○ Creación de instancias

Estrategias metodológicas para la implementación del módulo

El objetivo del presente módulo es dotar al participante de una base sólida en lenguaje Javascript que le permita resolver problemas de mediana complejidad bajo el paradigma de la programación asíncrona, orientada a eventos y orientada a objetos y así, brindar una visión general del proceso de desarrollo de algoritmos y estructuras de datos usando al máximo las potencialidades del lenguaje. Bajo un aprendizaje práctico el estudiante diseñará y desarrollará algoritmos complejos y utilizará características propias del lenguaje Javascript para resolver problemas de captura procesamiento datos y eventos. El alumno utilizará todas las potencialidades y características de los objetos JSON para dar forma a soluciones que le permitirán resolver problemas empresariales complejos, de manera eficiente y de acuerdo a los requerimientos pedidos. Con todo esto, el alumno adquiere una visión general del proceso de desarrollo bajo plataforma Javascript que sentará las bases de su conocimiento

del lenguaje y sus objetos para la posterior utilización de herramientas más complejas basadas en esta tecnología como Node y MongoDB, módulos siguientes donde deberá aplicar todo el conocimiento adquirido en esta sección.

Para la implementación de este módulo, se sugiere la aplicación de una metodología de aprendizaje basado en ejemplos a seguir y trabajo práctico del alumno en función a retos, idealmente en cada clase práctica, se recomienda tener a la vista la consola de comandos e ir desplegando constantemente el resultado de lo realizado a la línea de comandos de la consola para ir viendo en tiempo real el resultado de lo aplicado

Para cada clase se sugiere dividir el proceso en 3 etapas. Durante la primera etapa, eminentemente teórica, el instructor podrá realizar sus clases generando una cadencia entre diversas técnicas, tanto centradas en el instructor, como en el alumno, o en el grupo. Luego el instructor procederá mostrar el proceso de construcción de un sitio con las herramientas enseñadas en la unidad y finalmente dejará lugar a trabajo práctico de los alumnos para fomentar las competencias declaradas.

Los retos propuestos en cada sesión deben ser contextualizados en situaciones reales, provenientes de diversas industrias, tales como retail, banca, seguros, utilities, salud, educación, entre otros, para de esta forma construir un aprendizaje significativo y contextualizado.

Desarrollo de conocimientos

Para el desarrollo de los conocimientos clave esperados por los estudiantes, se recomienda apoyar los ejemplos y casos de estudio con literatura formal y bibliografía de tendencia, además de variados ejercicios de auto aplicación, como experimentación controlada, gamificación y/o dinámicas de grupo, donde se puedan aplicar con el grupo de alumnos ciertos experimentos desarrollados por la ciencia y que fundamentan las bases del comportamiento.

Estrategia Evaluativa del módulo

La evaluación del módulo considerará una prueba de selección múltiple, donde pueda reconocer los elementos más significativos del aprendizaje, dándole una estructura a lo largo del proceso evaluativo.

También se sugiere incorporar autoevaluaciones y coevaluaciones formativas durante el transcurso del módulo.

Perfil del Facilitador

Características Transversales

El instructor debe poseer competencias pedagógicas tales como:

- Planificar la clase

- Demostrar dominio de los temas tratados
- Capacidad de motivar
- Capacidad de contextualizar los temas en el área laboral
- Capacidad de cumplir un rol de facilitador
- Habilidades comunicacionales acordes a la función

Características Específicas

- Formación académica en el área de las ingenierías, o como técnico de nivel superior en el área informática, con título
- Experiencia laboral de mínimo 2 años desempeñando funciones de desarrollo de aplicaciones backend con Node
- Experiencia como facilitador/a de capacitación para personas adultas, y/o bien como docente en instituciones de educación superior, sean universidades, institutos profesionales o centros de formación técnica, de mínimo 100 horas cronológicas, demostrable

Recursos materiales para la implementación del módulo formativo

Infraestructura	Equipos y Herramientas	Materiales e Insumos
<ul style="list-style-type: none"> • Sala de clases implementada con: Puestos de trabajo individuales que considere mesa y silla. Escritorio y silla para el facilitador. • Conexiones para utilizar medios didácticos tales como data y salida a internet y salida de audio • Sistema de iluminación y ventilación adecuados 	<ul style="list-style-type: none"> • Proyector multimedia • Notebook o PC, para el facilitador • Telón • Pizarra • Equipo computacional, uno para cada participante, al menos Pentium Core 5, 8 GB RAM y 500 GB HD, y conexión a internet, con el siguiente software instalado: <ul style="list-style-type: none"> • Node 10.X o superior compatible 	<ul style="list-style-type: none"> • Plumones para pizarrón. • Lista de participantes. • Carpeta de registro de evidencias para el facilitador • Material audiovisual que contenga información del curso: • Manual didáctico que contemple todos los contenidos especificados para este módulo, por cada participante.

<ul style="list-style-type: none">• Servicios higiénicos separados para hombres y mujeres en recintos de aulas y de actividades prácticas.	<ul style="list-style-type: none">• Visual Studio Code Editor 1.3 o superior• MongoDB 4.X• Robo 3T 1.3 o superior	
--	---	--

Especialización Desarrollo Front-End

Módulo Formativo N° 03	
Nombre	Desarrollo de Interfaces Interactivas con Framework React
N° de horas	18 horas
Requisitos de ingreso	Carrera Técnica o Profesional; Experiencia Laboral TI; Test de Selección, en donde se mide: <ul style="list-style-type: none"> • Conocimientos de Fundamentos de Programación en Javascript • HTML y CSS
Competencia del módulo	<ul style="list-style-type: none"> • Implementar una interfaz de usuario web con elementos interactivos utilizando el framework Vue.js para dar solución a un requerimiento

Aprendizajes esperados	Criterios de evaluación	Contenidos
1. Reconocer origen e historia de la biblioteca de código de ReactJS identificando las tecnologías web que utiliza y comparando con otras tecnologías para el desarrollo de aplicaciones front-end web	1.1 Compara ReactJS con otros lenguajes y frameworks que se utilizan en la actualidad en el desarrollo web 1.2 Reconoce las características y prerequisites de ReactJS para el desarrollo de aplicaciones web 1.3 Identifica las tecnologías web base que utiliza ReactJS para el desarrollo de aplicaciones web	Introducción a ReactJs <ul style="list-style-type: none"> • ¿Qué es ReactJs y cuál es su origen? • ¿Por qué usar ReactJs? • Acerca de las aplicaciones SPA • Renderización de elementos del DOM • ReactJs y ReactJs Native • Descripción y uso de ReactJs en aplicaciones web • ¿Es ReactJs un framework? • ReactJs versus otros Frameworks basados en Javascript • ¿Qué NO es ReactJs?

		<p>Características de ReactJs</p> <ul style="list-style-type: none"> • Librería basada en Javascript • Lenguaje declarativo • Basado en componentes • Manejo del DOM • Isomorfismo (renderización en cliente y servidor) • Sintaxis JSX • ReactJs sin JSX • Flujo de datos unidireccional <p>Entornos de ejecución</p> <ul style="list-style-type: none"> • Prerrequisitos para la utilización de ReactJS • Versiones de Javascript compatibles con ReactJs • Ejecución desde CDN • Ejecución desde un ambiente local • Librerías usadas por ReactJs (Webpack, Babel, Next.js, Redux)
2. Describir los elementos fundamentales del entorno ReactJS y su utilidad para el desarrollo de aplicaciones front-end	<p>2.1 Identifica los elementos fundamentales de la arquitectura del entorno ReactJS para el desarrollo de aplicaciones front-end</p> <p>2.2 Describe el rol de los elementos fundamentales de la arquitectura del entorno ReactJS para el desarrollo de aplicaciones front-end</p> <p>2.3 Distingue las diferencias del código declarativo versus el imperativo imperativo como principio de codificación</p>	<p>Elementos fundamentales de ReactJS</p> <ul style="list-style-type: none"> • Introducción a JSX • Pensando en ReactJs • Componentes en ReactJs • Composición versus herencia • Paso de datos con Props • Listas y keys • Formularios • El ciclo de vida ReactJs • Desplegando datos en la interfaz (Renderización) • Uso de extensión browser “React Developer Tools” • Hojas de estilo en ReactJs • Introducción a Hooks

		<ul style="list-style-type: none"> ● Introducción a uso de API's
3. Utilizar JSX para la representación simplificada de elementos html en el entorno React que dan solución a un problema	<p>3.1 Describe las características generales de JSX para la simplificación de código React</p> <p>3.2 Reconoce la sintaxis, notación y palabras reservadas de expresiones JSX en un entorno React</p> <p>3.3 Implementa código JSX para la resolución de un problema utilizando el framework React</p>	<p>JSX, la simplificación de ReactJS</p> <ul style="list-style-type: none"> ● Por qué usar JSX ● Insertando expresiones en JSX ● Notación y palabras reservadas ● JSX también es una expresión ● Especificando atributos con JSX ● Especificando hijos con JSX ● Previendo ataques de inyección con JSX ● Representación de objetos en JSX ● El rol de Babel en la conversión de objetos ● Comentarios en JSX
4. Implementar componentes reutilizables de acuerdo al entorno React para dar solución a un problema	<p>4.1 Describe los aspectos fundamentales y beneficios de la componentización en el desarrollo de aplicaciones front-end</p> <p>4.2 Reconoce los elementos y sintaxis requerida para la especificación de un componente</p> <p>4.3 Utiliza componentes previamente construidos utilizando sus propiedades para su personalización</p> <p>4.4 Implementa componentes definiendo parámetros de propiedades para facilitar la reutilización de dicho componente</p>	<p>Componentización</p> <ul style="list-style-type: none"> ● Por qué usar componentes ● Ventajas de usar componentes (Reutilización de código, escalabilidad, facilidad de uso, mantenimiento) ● Especificación de componentes <ul style="list-style-type: none"> ○ Declaración ○ Extensiones ○ Propiedades ○ Manejo de eventos ○ Componentes basadas en funciones ○ Componentes basadas en clases <ul style="list-style-type: none"> ▪ Estructura de una clase ▪ Constructores

	4.5 Implementa componente definiendo estados para dar solución a un problema	<ul style="list-style-type: none"> ▪ Renderización y renderización condicional <ul style="list-style-type: none"> ○ Convertir una función en una clase ○ Instanciación de componentes ● Enviando datos a componentes usando propiedades <ul style="list-style-type: none"> ○ ¿Qué son las props? ○ ¿Para qué sirven las props? ○ Cómo se utilizan los props dentro de las componentes ● Cambiando el comportamiento de los componentes (uso de State) ● Utilizando componentes ● Navegación entre componentes <p>Manejo de estados</p> <ul style="list-style-type: none"> ● Qué son los estados ● Tipos de estados ● Manejo de estados en un componente ● Asignar estado inicial en el constructor ● Actualizar estado ● ¿Qué ocurre cuando hay un cambio de estado? ● El callback de setState()
5. Implementar un prototipo de una interfaz web navegable y accesible utilizando framework React para dar solución a una necesidad	<p>5.1 Reconoce buenas prácticas de accesibilidad para la implementación de un prototipo navegable</p> <p>5.2 Distingue las diferencias entre enrutamiento estático y dinámico para</p>	<p>Enrutamiento en ReactJs</p> <ul style="list-style-type: none"> ● Acerca de React Router ● Enrutamiento estático v/s enrutamiento dinámico ● Browser Router ● Switch ● Route ● Redirect

	<p>darle navegabilidad a un prototipo de una aplicación web React</p> <p>5.3 Utiliza directrices para dar accesibilidad a un prototipo web con React</p> <p>5.4 Implementa un prototipo navegable utilizando React Router para dar solución a una necesidad</p>	<ul style="list-style-type: none"> • Link <p>Accesibilidad</p> <ul style="list-style-type: none"> • Normas y Directrices • HTML semántico • Formularios Accesibles • Control de enfoque • Eventos de mouse y puntero • Widgets más complejos
<p>6. Construir una aplicación web de interfaz de usuario navegable utilizando el framework React para ser desplegada en un servicio host</p>	<p>6.1 Reconoce el entorno de desarrollo requerido para el desarrollo de un proyecto React</p> <p>6.2 Utiliza la interfaz CLI para la realización de operaciones básicas en el desarrollo de un proyecto React</p> <p>6.3 Genera código compilado del proyecto para el despliegue del aplicativo en un servicio host</p>	<p>Creación de un proyecto básico con ReactJs</p> <ul style="list-style-type: none"> • Prerrequisitos para la utilización de ReactJs • Entorno de ejecución Node.Js <ul style="list-style-type: none"> ○ Qué es Node.js ○ Para que se utiliza ○ Manejo de paquetes con npm • Interfaz CLI de ReactJs <ul style="list-style-type: none"> ○ Qué es la interfaz CLI ○ Para qué se utiliza ○ Principales comandos • Crear aplicación vía Comandos CLI <ul style="list-style-type: none"> ○ “Create react app” ○ Creación a partir de templates ○ Estructura de directorio de un proyecto ReactJs • Ejecutar una aplicación ReactJs • Compilar y desplegar una aplicación ReactJS <p>Navegar un prototipo funcional básico</p> <ul style="list-style-type: none"> • Crear componentes utilizando JSX • Renderizar componentes aplicando el uso de Props • Enrutar componentes

		<ul style="list-style-type: none"> • Manipulando elementos del DOM • Aplicando estilos al componente • Manejando eventos • Hacer uso del State
--	--	--

Estrategias metodológicas para la implementación del módulo

Para la implementación de dicho módulo, se sugiere la aplicación de una metodología de aprendizaje basado en problemas o bien una metodología de proyectos, propiciando de esta forma, el “aprender haciendo”. Asimismo, se sugiere realizar actividades en donde se propicien prácticas como la “programación de pares” o bien en donde se produzca interacción y discusión entre los participantes.

El facilitador realiza una clase expositiva en donde brevemente se exponen los objetivos de la sesión y la explicación de los conceptos de forma resumida. De la misma forma, realiza una demostración práctica de los conceptos para luego realizar actividades prácticas de trabajo basado en un problema. Para finalizar, se realiza una autoevaluación o coevaluación entre los participantes o un análisis grupal del problema elaborando conclusiones de la sesión. También se sugiere el trabajo en grupos pequeños, por ejemplo de pares (pair programming), de esta forma se facilitan los atascos y se produce el intercambio de aprendizaje entre los participantes. Es importante que las demostraciones realizadas por el facilitador no solamente considere la resolución del problema sino que también cometa errores de forma intencionada para que el participante tenga la posibilidad ver cómo enfrentar una situación de atasco y cuál es el raciocinio lógico empleado para buscar la solución, comprendiendo así que el proceso de construcción de una pieza de software no siempre resulta a la primera sino que es un proceso en donde gradualmente se acerca a una solución aceptable.

Los problemas propuestos en cada sesión deben ser contextualizados en situaciones reales, preferentemente de la industria del retail o de la banca, para de esta forma construir un aprendizaje significativo y contextualizado, es decir, que los conectan con experiencias que han vivido como clientes o consumidores. Por ejemplo, se pueden utilizar ejemplos de puntos de venta, carros de compra, back office de mantención de datos, cuentas corrientes, etc.

Desarrollo de conocimientos

Es fundamental en este módulo una debida planificación de las actividades para lograr el mejor resultado. Si bien es cierto que se declaran contenidos en extenso, los criterios de evaluación son más bien genéricos, permitiendo de esta manera, al facilitador ir tomando decisiones sobre el nivel de profundidad en que se abordarán los contenidos de acuerdo al desempeño que va observando en el grupo de participantes. Sí es

importante, que al menos ciertos contenidos sean mencionados para que al menos el participante conozca de su existencia y tenga la posibilidad de profundizarlos por iniciativa personal.

Desarrollo de habilidades de empleabilidad

Para promover las habilidades conductuales y/o actitudes de tolerancia a la frustración y de autoaprendizaje, se debe integrar como parte de la dinámica de trabajo o de aprendizaje, que los participantes busquen información en internet o bien discutan con sus compañeros cuando se encuentren con trabas para solucionar el problema planteado. Asimismo, propiciar que el participante utilice el método de prueba y error al enfrentar trabas, antes de entregarles la solución.

Estrategia Evaluativa del módulo

La evaluación del módulo considerará una prueba de selección múltiple, donde pueda reconocer los elementos más significativos del aprendizaje, dándole una estructura a lo largo del proceso evaluativo.

También se sugiere incorporar autoevaluaciones y coevaluaciones formativas durante el transcurso del módulo.

Perfil del Facilitador

Características Transversales

El instructor debe poseer competencias pedagógicas tales como:

- Planificar la clase
- Demostrar dominio de los temas tratados
- Capacidad de motivar
- Capacidad de contextualizar los temas en el área laboral
- Capacidad de cumplir un rol de facilitador
- Habilidades comunicacionales acordes a la función

Características Específicas

- Formación académica en el área de las ingenierías, con título
- Experiencia laboral de mínimo 3 años desempeñando funciones de desarrollo web, demostrable.

- Experiencia como facilitador/a de capacitación para personas adultas, y/o bien como docente en instituciones de educación superior, sean universidades, institutos profesionales o centros de formación técnica, de mínimo 100 horas cronológicas, demostrable

Recursos materiales para la implementación del módulo formativo

Infraestructura	Equipos y Herramientas	Materiales e Insumos
<ul style="list-style-type: none"> • Sala de clases implementada con: Puestos de trabajo individuales que considere mesa y silla. Escritorio y silla para el facilitador. • Conexiones para utilizar medios didácticos tales como data y salida a internet y salida de audio • Sistema de iluminación y ventilación adecuados • Servicios higiénicos separados para hombres y mujeres en recintos de aulas y de actividades prácticas. 	<ul style="list-style-type: none"> • Proyector multimedia • Notebook o PC, para el facilitador • Telón • Pizarra • Equipo computacional para cada participante: Mac o PC con 8 de ram mínimo, procesador Intel i5 mínimo • Software instalado: <ul style="list-style-type: none"> ○ Navegador ○ Visual Studio Code https://code.visualstudio.com/download ○ Node JS https://nodejs.org/es/download/ 	<ul style="list-style-type: none"> • Plumones para pizarrón. • Lista de participantes. • Carpeta de registro de evidencias para el facilitador • Material audiovisual que contenga información del curso: • Manual didáctico que contemple todos los contenidos especificados para este módulo, por cada participante.

Especialización Desarrollo Front-End

Módulo Formativo N° 04	
Nombre	Desarrollo de Aplicaciones Front-End con Framework React
N° de horas	24 horas
Requisitos de ingreso	Carrera Técnica o Profesional; Experiencia Laboral TI; Test de Selección, en donde se mide: <ul style="list-style-type: none"> • Conocimientos de Fundamentos de Programación en Javascript • HTML y CSS
Competencia del módulo	<ul style="list-style-type: none"> • Implementar una aplicación Front-End utilizando un framework orientado a componentes para dar solución a un requerimiento

Aprendizajes esperados	Criterios de evaluación	Contenidos
1. Reconocer los aspectos fundamentales de la manipulación del DOM en el entorno React para el desarrollo de aplicativos web	1.1 Describe las características de React y del DOM Virtual 1.2 Describe las características de las referencias indicando cuándo es conveniente utilizarlas y cómo acceder a ellas 1.3 Reconoce las características principales del paquete React-DOM facilitar la manipulación del DOM	Manejo del DOM <ul style="list-style-type: none"> • Elementos DOM en ReactJs • Acerca de React.Component • El DOM Virtual en ReactJs <ul style="list-style-type: none"> ○ Qué es el DOM Virtual ○ Qué es el Shadow DOM ○ Qué es React Fiber • Uso de referencias <ul style="list-style-type: none"> ○ Qué son las referencias ○ Cuando usar referencias ○ Creación de referencias

		<ul style="list-style-type: none"> ○ Accediendo a las referencias ○ Agregando referencias al DOM ○ Referencias mediante Callback ● El paquete React-DOM <ul style="list-style-type: none"> ○ React DOM en el cliente ○ React DOM en el servidor ○ Soporte en navegadores ● Diferencias en atributos HTML que funcionan diferente en ReactJs (checked, className, onChange, selected) ● Utilidades para pruebas ● Renderizador de prueba ● Requerimientos del entorno de JS
2. Reconoce características avanzadas del entorno React para el desarrollo de aplicaciones front-end	<p>2.1 Describe el concepto de Contexto distinguiendo los casos en que es conveniente su utilización</p> <p>2.2 Describe el concepto de Fragmento en React reconociendo su utilidad</p> <p>2.3 Describe el concepto de Reconciliación en React reconociendo su utilidad</p> <p>2.4 Describe el concepto de Portales en React reconociendo su utilidad</p> <p>2.5 Describe las características del Modo Estricto en React y su utilización</p>	<p>Elementos avanzados de ReactJS</p> <ul style="list-style-type: none"> ● Manejo de socket.io ● División de código ● Transformar elementos ● Contexto ● Fragmentos ● Transitions ● Componentes de orden superior ● Optimizando el rendimiento ● Portales ● Profiler ● Reconciliación ● Comprobación de tipos estáticos ● Modo estricto ● Verificación de tipos con PropTypes ● Componentes no controlados ● Web Components

3. Implementar un aplicativo React que consume datos desde una API REST utilizando la librerías Fetch y Axios para dar solución a una problemática	<p>1.4 Verifica el funcionamiento de una API Rest publicada en un servidor utilizando una herramienta cliente</p> <p>1.5 Implementa un componente React que consume datos de una API Rest utilizando la librería Axios para dar solución a un problema</p> <p>1.6 Implementa un componente React que consume datos de una API Rest utilizando la librería Fetch para dar solución a un problema</p>	<p>Consumo de API</p> <ul style="list-style-type: none"> • El rol del front en una aplicación Cliente/Servidor • Interacción a través de APIs • Usando el Hook useEffect <ul style="list-style-type: none"> ○ Qué hace useEffect ○ Omite efectos para optimizar el rendimiento ○ Cómo realizar peticiones en React con useEffect ○ Cómo realizar peticiones a partir de eventos del usuario ○ Manejando errores • Usando el Hook useState • Usando Fetch API <ul style="list-style-type: none"> ○ Qué es Fetch API ○ Cómo utilizar Fetch API ○ Cómo manejar errores con Fetch API • Usando Axios <ul style="list-style-type: none"> ○ Sintaxis ○ Invocación ○ Manejo de errores ○ Usando Async/Await • Principales diferencias entre Fetch API y Axios
4. Reconocer los aspectos fundamentales de la programación TypeScript reconociendo sus diferencias con JavaScript e identificando sus beneficios en un proyecto React	<p>1.7 Reconoce los aspectos fundamentales de TypeScript y su utilidad en un proyecto React</p> <p>1.8 Reconoce la sintaxis básica de TypeScript para la codificación de una pieza de software</p>	<p>Introducción a TypeScript</p> <ul style="list-style-type: none"> • Qué es TypeScript • Para qué se utiliza TypeScript • TypeScript en React.js • TypeScript v/s Javascript • TypeScript y Webpack • Definiendo tipos

	<p>1.9 Implementa rutinas TypeScript utilizando definición de tipos para resolver un problema simple</p> <p>1.10 Implementa rutinas TypeScript utilizando clases e interfaces para resolver un problema simple</p>	<ul style="list-style-type: none"> • Tipos por inferencia • Componiendo tipos • Sistema de tipo estructural • Interfaces y Clases • Algunos Frameworks que soportan TypeScript (Next.js, Gatsby.js)
<p>5. Describe los aspectos fundamentales de seguridad a considerar en el desarrollo de un aplicativo Front-End bajo el entorno React</p>	<p>5.1 Identifica las principales vulnerabilidades al cual está expuesto un aplicativo Front-End</p> <p>5.2 Describe recomendaciones de seguridad en un aplicativo Fron-End implementado en el entorno React</p> <p>5.3 Reconoce prácticas para la securización de rutas en React</p> <p>5.4 Implementa mecanismos de seguridad para el consumo de API tales como Api Key y JWT bajo el entorno React</p>	<p>Seguridad en un aplicativo Front-End</p> <ul style="list-style-type: none"> • Conceptos básicos de seguridad en aplicaciones Web (Clickjacking, Ataque XSS, SQL Injection, Ataque DoS) • Recomendaciones de seguridad en una aplicación Web • Recomendaciones de seguridad en una aplicación ReactJs • Identificando vulnerabilidades en una aplicación ReactJs • Consumiendo servicios REST con Api Key y JWT • Cómo proteger rutas con React Router DOM • Implementando seguridad por Roles en React • La seguridad en la autenticación de usuarios • Encriptación de datos en el front
<p>6. Utilizar mecanismos de Hooks y Manejo de Errores en un aplicativo React para dar solución a un problema planteado</p>	<p>6.1 Describe el concepto de hook y sus utilidades de acuerdo al entorno React</p> <p>6.2 Identifica los conceptos de Hook de Estado y de Hook de Efecto en un aplicativo React</p>	<p>Hooks</p> <ul style="list-style-type: none"> • Qué son los Hooks • Un vistazo en detalle a los Hooks • Usando el Hook de estado • Usando el Hook de efecto • Reglas de los Hooks • Construyendo Hooks personalizados

	<p>6.3 Reconoce los mecanismos para el manejo de errores en un aplicativo React</p> <p>6.4 Utiliza hooks en un aplicativo React para resolver un problema</p> <p>6.5 Utiliza mecanismos disponibles en el entorno React para el manejo y control de los errores y excepciones</p>	<p>Manejo y límite de errores</p> <ul style="list-style-type: none"> ● React Error Handling ● Creación de límite de errores ● ¿Qué método usar? ● Identificación de límites de error ● Limitaciones de los límites de error ● Error Boundaries ● Nuevo comportamiento para errores no detectados ● Manejando el Stack Trace ● Try/catch ● Gatillando errores personalizados ● Atrapando errores ● Interpretando los códigos de error
<p>7. Implementa pruebas unitarias utilizando las herramientas provistas por React para verificar el correcto funcionamiento del aplicativo</p>	<p>7.1 Reconoce los conceptos fundamentales del testing distinguiendo buenas prácticas en las pruebas de un aplicativo front-end</p> <p>7.2 Identifica herramientas utilizadas para la realización de pruebas unitarias en un entorno React</p> <p>7.3 Implementa pruebas unitarias utilizando las herramientas provistas por React para verificar el funcionamiento de un componente</p>	<p>Testing</p> <ul style="list-style-type: none"> ● Visión general del proceso de pruebas ● Por qué hacer pruebas ● Mejores prácticas en proceso de pruebas ● Entornos de prueba <ul style="list-style-type: none"> ○ Instalación de librerías ● Herramientas para pruebas <ul style="list-style-type: none"> ○ Jest ○ React Testing Library ● Creación de unidades pruebas ● Ejecutando unidades de prueba ● Instrucciones más utilizadas en unidades de prueba

<p>8. Implementar un aplicativo utilizando el entorno React para dar solución a un problema empleando buenas prácticas</p>	<p>8.1 Reconoce buenas prácticas para el desarrollo de un aplicativo front-end web React</p> <p>8.2 Utiliza mecanismos tales como PropTypes o TypeScript para la verificación de tipos en un aplicativo React</p> <p>8.3 Utiliza herramientas de Linting para la detección de errores en el código</p> <p>8.4 Utiliza conceptos de componentización en distintos niveles de jerarquía para facilitar la reutilización de código</p> <p>8.5 Utiliza convenciones de nombramiento de archivos, carpetas y componentes</p>	<p>Creación de una aplicación frontend ReactJs</p> <ul style="list-style-type: none"> ● Seguridad y comunicación cliente/servidor ● Buenas prácticas en el desarrollo de aplicaciones Web ReactJs <ul style="list-style-type: none"> ○ Organización de la estructura de directorios ○ Nombramiento de archivos y carpetas ○ Diseño de Componentes <ul style="list-style-type: none"> ▪ Compactas ▪ Simples ▪ Reusables ▪ Única responsabilidad ○ Buen nombramiento de las componentes ○ Reutilización de código ○ Organización de los estados ○ Organización de las hojas de estilo ○ Lógica de representación y de estado independientes ○ Utilización PropTypes (o TypeScript) para la verificación de tipos ○ De preferencia uso de Hooks ○ Utilización herramientas de linting ○ Escritura de casos de prueba ● Creando una aplicación ReactJs usando componentes en distintos niveles de jerarquía ● Qué consideraciones tener al desplegar una aplicación en producción ● Incorporar el consumo de API Rest ● Incorporar conceptos de seguridad ● Incorporar unidades de prueba
--	---	---

--	--	--

Estrategias metodológicas para la implementación del módulo

Para la implementación de dicho módulo, se sugiere la aplicación de una metodología de aprendizaje basado en problemas o bien una metodología de proyectos, propiciando de esta forma, el “aprender haciendo”. Asimismo, se sugiere realizar actividades en donde se propicien prácticas como la “programación de pares” o bien en donde se produzca interacción y discusión entre los participantes.

El facilitador realiza una clase expositiva en donde brevemente se exponen los objetivos de la sesión y la explicación de los conceptos de forma resumida. De la misma forma, realiza una demostración práctica de los conceptos para luego realizar actividades prácticas de trabajo basado en un problema. Para finalizar, se realiza una autoevaluación o coevaluación entre los participantes o un análisis grupal del problema elaborando conclusiones de la sesión. También se sugiere el trabajo en grupos pequeños, por ejemplo de pares (pair programming), de esta forma se facilitan los atascos y se produce el intercambio de aprendizaje entre los participantes. Es importante que las demostraciones realizadas por el facilitador no solamente considere la resolución del problema sino que también cometa errores de forma intencionada para que el participante tenga la posibilidad ver cómo enfrentar una situación de atasco y cuál es el raciocinio lógico empleado para buscar la solución, comprendiendo así que el proceso de construcción de una pieza de software no siempre resulta a la primera sino que es un proceso en donde gradualmente se acerca a una solución aceptable.

Los problemas propuestos en cada sesión deben ser contextualizados en situaciones reales, preferentemente de la industria del retail o de la banca, para de esta forma construir un aprendizaje significativo y contextualizado, es decir, que los conectan con experiencias que han vivido como clientes o consumidores. Por ejemplo, se pueden utilizar ejemplos de puntos de venta, carros de compra, back office de mantención de datos, cuentas corrientes, etc.

Desarrollo de conocimientos

Es fundamental en este módulo una debida planificación de las actividades para lograr el mejor resultado. Si bien es cierto que se declaran contenidos en extenso, los criterios de evaluación son más bien genéricos, permitiendo de esta manera, al facilitador ir tomando decisiones sobre el nivel de profundidad en que se abordarán los contenidos de acuerdo al desempeño que va observando en el grupo de participantes. Sí es importante, que al menos ciertos contenidos sean mencionados para que al menos el participante conozca de su existencia y tenga la posibilidad de profundizarlos por iniciativa personal.

Desarrollo de habilidades de empleabilidad

Para promover las habilidades conductuales y/o actitudes de tolerancia a la frustración y de autoaprendizaje, se debe integrar como parte de la dinámica de trabajo o de aprendizaje, que los participantes busquen información en internet o bien discutan con sus compañeros cuando se encuentren con trabas para solucionar el problema planteado. Asimismo, propiciar que el participante utilice el método de prueba y error al enfrentar trabas, antes de entregarles la solución.

Estrategia Evaluativa del módulo

La evaluación del módulo considerará una prueba de selección múltiple, donde pueda reconocer los elementos más significativos del aprendizaje, dándole una estructura a lo largo del proceso evaluativo.

También se sugiere incorporar autoevaluaciones y coevaluaciones formativas durante el transcurso del módulo.

Perfil del Facilitador

Características Transversales

El instructor debe poseer competencias pedagógicas tales como:

- Planificar la clase
- Demostrar dominio de los temas tratados
- Capacidad de motivar
- Capacidad de contextualizar los temas en el área laboral
- Capacidad de cumplir un rol de facilitador
- Habilidades comunicacionales acordes a la función

Características Específicas

- Formación académica en el área de las ingenierías, con título
- Experiencia laboral de mínimo 3 años desempeñando funciones de desarrollo web, demostrable.
- Experiencia como facilitador/a de capacitación para personas adultas, y/o bien como docente en instituciones de educación superior, sean universidades, institutos profesionales o centros de formación técnica, de mínimo 100 horas cronológicas, demostrable

Recursos materiales para la implementación del módulo formativo

Infraestructura	Equipos y Herramientas	Materiales e Insumos
<ul style="list-style-type: none"> • Sala de clases implementada con: Puestos de trabajo individuales que considere mesa y silla. Escritorio y silla para el facilitador. • Conexiones para utilizar medios didácticos tales como data y salida a internet y salida de audio • Sistema de iluminación y ventilación adecuados • Servicios higiénicos separados para hombres y mujeres en recintos de aulas y de actividades prácticas. 	<ul style="list-style-type: none"> • Proyector multimedia • Notebook o PC, para el facilitador • Telón • Pizarra • Equipo computacional para cada participante: Mac o PC con 8 de ram mínimo, procesador Intel i5 mínimo • Software instalado: <ul style="list-style-type: none"> ○ Navegador ○ Visual Studio Code https://code.visualstudio.com/download ○ Node JS https://nodejs.org/es/download/ 	<ul style="list-style-type: none"> • Plumones para pizarrón. • Lista de participantes. • Carpeta de registro de evidencias para el facilitador • Material audiovisual que contenga información del curso: • Manual didáctico que contemple todos los contenidos especificados para este módulo, por cada participante.

Especialización Desarrollo Front-End

Módulo Formativo N° 05	
Nombre	Desarrollo de Aplicaciones Web Progresivas
N° de horas	12 horas
Requisitos de ingreso	Título técnico o profesional; Experiencia laboral. Test de conocimiento: <ul style="list-style-type: none"> • Programación Javascript • HTML y CSS
Competencia del módulo	<ul style="list-style-type: none"> • Implementar una aplicación web progresiva utilizando el entorno React de acuerdo a los requerimientos

Aprendizajes esperados	Criterios de evaluación	Contenidos
1. Reconocer las características de una PWA, identificando beneficios y limitaciones, arquitectura y componentes, y frameworks para el desarrollo	1.1 Describe las características de una PWA identificando sus beneficios y limitaciones 1.2 Reconoce los elementos que componen una arquitectura PWA 1.3 Distingue alternativas de frameworks para el desarrollo de aplicaciones PWA	Introducción a las aplicaciones PWA <ul style="list-style-type: none"> • Acerca de las aplicaciones Nativas • Diferencia entre aplicaciones nativas y aplicaciones Web • Qué es una PWA (Progressive Web Application) • Por qué crear una PWA • Características de una PWA (Progresiva, responsiva, adaptable, segura, independiente) • Beneficios de una PWA • Limitaciones de una PWA • Aplicaciones PWA en móviles y PC de escritorio

		<ul style="list-style-type: none"> ● El uso de funciones nativas del sistema operativo ● Consideraciones de diseño de una PWA ● Operación “fuera de línea” de una PWA ● Diferencias entre una PWA, una aplicación Web tradicional y una aplicación Nativa ● Arquitectura y componentes de una PWA (service workers, manifiesto, Shell aplicación) ● Frameworks que soportan el desarrollo de PWA’s
2. Reconoce las características principales de los elementos que componen la arquitectura de una aplicación PWA	<p>2.1 Describe el rol del manifiesto dentro de una PWA identificando los principales elementos de su estructura</p> <p>2.2 Describe los elementos principales del manifiesto para la implementación de una PWA</p> <p>2.3 Describe el rol del Service Worker como componente de la arquitectura de una PWA</p> <p>2.4 Describe los elementos que componen el ciclo de vida de un Service Worker</p> <p>2.5 Reconoce estrategias de almacenamiento en caché de Service Worker</p>	<p>El Manifiesto</p> <ul style="list-style-type: none"> ● ¿Qué es el Manifiesto? ● ¿Para qué se usa el Manifiesto? ● Estructura de un archivo de Manifiesto <p>El Service Worker</p> <ul style="list-style-type: none"> ● ¿Qué es el Service Worker? ● ¿Para qué se usa el Service Worker? ● ¿Qué ventajas tiene usar un Service Worker? ● Descripción general de un Service Worker (API asíncrona, API basada en eventos, precaching, aislamiento del hilo principal) ● Ciclo de vida de un Service Worker ● Cómo interactúa el Service Worker con el caché y el acceso a la red ● Cómo configurar Service Worker para ReactJs ● Funcionamiento de una PWA con HTTPS ● Estrategias de almacenamiento en caché de Service Worker <ul style="list-style-type: none"> ○ Stale-While-Revalidate ○ Cache-first ○ Network First ○ Cache Only

		<ul style="list-style-type: none"> ○ Network Only ○ Cache And Network
3. Reconoce los mecanismos para el almacenamiento local de datos de una PWA	<p>3.1 Describe los mecanismos de almacenamiento de datos en un entorno local PWA</p> <p>3.2 Reconoce bibliotecas de bases de datos con soporte PWA</p>	<p>Almacenamiento en una PWA</p> <ul style="list-style-type: none"> ● Cómo se administra el Almacenamiento Web en una PWA ● El uso de LocalStorage y SessionStorage ● Acerca de WebAssembly y el código precompilado ● Bibliotecas de bases de datos con soporte para PWA (IndexedDB, PouchDB, RxDB, GunDB)
4. Implementar aplicativos PWA utilizando el entorno React para dar solución a un requerimiento	<p>4.1 Implementa un aplicativo PWA utilizando el entorno React para dar solución a un requerimiento de baja complejidad</p> <p>4.2 Optimiza un aplicativo PWA utilizando Lighthouse</p> <p>4.3 Despliega un aplicativo PWA en un dispositivo móvil para su posterior funcionamiento</p>	<p>Creando una PWA</p> <ul style="list-style-type: none"> ● Creando un proyecto PWA en ReactJs ● Registrando un Service Worker ● Personalizando un Service Worker ● Navegando a través de una PWA ● Implementando las distintas estrategias de almacenamiento en caché de Service Worker ● Accediendo a periféricos del Sistema Operativo ● Despliegue de una PWA <p>Explorando el estado de una PWA con Lighthouse</p> <ul style="list-style-type: none"> ● Instalando la extensión lighthouse en el navegador ● Algunos aspectos relevantes de lighthouse (Rápido, instalable, optimizado para PWA) ● Ejecutando lighthouse (Análisis de carga de página) ● Revisando el informe de lighthouse

Estrategias metodológicas para la implementación del módulo

Para la implementación de dicho módulo, se sugiere la aplicación de una metodología de aprendizaje basado en problemas o bien una metodología de proyectos, propiciando de esta forma, el “aprender haciendo”. Asimismo, se sugiere realizar actividades en donde se propicien prácticas como la “programación de pares” o bien en donde se produzca interacción y discusión entre los participantes.

El facilitador realiza una clase expositiva en donde brevemente se exponen los objetivos de la sesión y la explicación de los conceptos de forma resumida. De la misma forma, realiza una demostración práctica de los conceptos para luego realizar actividades prácticas de trabajo basado en un problema. Para finalizar, se realiza una autoevaluación o coevaluación entre los participantes o un análisis grupal del problema elaborando conclusiones de la sesión. También se sugiere el trabajo en grupos pequeños, por ejemplo de pares (pair programming), de esta forma se facilitan los atascos y se produce el intercambio de aprendizaje entre los participantes. Es importante que las demostraciones realizadas por el facilitador no solamente considere la resolución del problema sino que también cometa errores de forma intencionada para que el participante tenga la posibilidad ver cómo enfrentar una situación de atasco y cuál es el raciocinio lógico empleado para buscar la solución, comprendiendo así que el proceso de construcción de una pieza de software no siempre resulta a la primera sino que es un proceso en donde gradualmente se acerca a una solución aceptable.

Los problemas propuestos en cada sesión deben ser contextualizados en situaciones reales, preferentemente de la industria del retail o de la banca, para de esta forma construir un aprendizaje significativo y contextualizado, es decir, que los conectan con experiencias que han vivido como clientes o consumidores. Por ejemplo, se pueden utilizar ejemplos de puntos de venta, carros de compra, back office de mantención de datos, cuentas corrientes, etc.

Desarrollo de conocimientos

Es fundamental en este módulo una debida planificación de las actividades para lograr el mejor resultado. Si bien es cierto que se declaran contenidos en extenso, los criterios de evaluación son más bien genéricos, permitiendo de esta manera, al facilitador ir tomando decisiones sobre el nivel de profundidad en que se abordarán los contenidos de acuerdo al desempeño que va observando en el grupo de participantes. Sí es importante, que al menos ciertos contenidos sean mencionados para que al menos el participante conozca de su existencia y tenga la posibilidad de profundizarlos por iniciativa personal.

Desarrollo de habilidades de empleabilidad

Para promover las habilidades conductuales y/o actitudes de tolerancia a la frustración y de autoaprendizaje, se debe integrar como parte de la dinámica de trabajo o de aprendizaje, que los participantes busquen información en internet o bien discutan con sus compañeros cuando se

encuentren con trabas para solucionar el problema planteado. Asimismo, propiciar que el participante utilice el método de prueba y error al enfrentar trabas, antes de entregarles la solución.

Estrategia Evaluativa del módulo

La evaluación del módulo considerará una prueba de selección múltiple, donde pueda reconocer los elementos más significativos del aprendizaje, dándole una estructura a lo largo del proceso evaluativo.

También se sugiere incorporar autoevaluaciones y coevaluaciones formativas durante el transcurso del módulo.

Perfil del Facilitador

Características Transversales

El instructor debe poseer competencias pedagógicas tales como:

- Planificar la clase
- Demostrar dominio de los temas tratados
- Capacidad de motivar
- Capacidad de contextualizar los temas en el área laboral
- Capacidad de cumplir un rol de facilitador
- Habilidades comunicacionales acordes a la función

Características Específicas

- Formación académica en el área de las ingenierías, con título
- Experiencia laboral de mínimo 3 años desempeñando funciones de desarrollo web, demostrable.
- Experiencia como facilitador/a de capacitación para personas adultas, y/o bien como docente en instituciones de educación superior, sean universidades, institutos profesionales o centros de formación técnica, de mínimo 100 horas cronológicas, demostrable

Recursos materiales para la implementación del módulo formativo

Infraestructura	Equipos y Herramientas	Materiales e Insumos
-----------------	------------------------	----------------------

<ul style="list-style-type: none"> ● Sala de clases implementada con: Puestos de trabajo individuales que considere mesa y silla. Escritorio y silla para el facilitador. ● Conexiones para utilizar medios didácticos tales como data y salida a internet y salida de audio ● Sistema de iluminación y ventilación adecuados ● Servicios higiénicos separados para hombres y mujeres en recintos de aulas y de actividades prácticas. 	<ul style="list-style-type: none"> ● Proyector multimedia ● Notebook o PC, para el facilitador ● Telón ● Pizarra ● Equipo computacional para cada participante: Mac o PC con 8 de ram mínimo, procesador Intel i5 mínimo ● Software instalado: <ul style="list-style-type: none"> ○ Navegador ○ Visual Studio Code https://code.visualstudio.com/download ○ Node JS https://nodejs.org/es/download/ 	<ul style="list-style-type: none"> ● Plumones para pizarrón. ● Lista de participantes. ● Carpeta de registro de evidencias para el facilitador ● Material audiovisual que contenga información del curso: ● Manual didáctico que contemple todos los contenidos especificados para este módulo, por cada participante.
--	--	---

Módulos Transversales Disciplinarios

Módulo Formativo N°5	
Nombre	Fundamentos de Desarrollo Agile
N° de horas	12 horas
Requisitos de ingreso	Título Técnico / Profesional;
Competencia del módulo	<ul style="list-style-type: none"> Reconocer los conceptos fundamentales, principios y valores de las metodologías ágiles así como también los valores, roles, prácticas y artefactos asociados al marco de trabajo Scrum para el desarrollo de un producto digital

Aprendizajes esperados	Criterios de evaluación	Contenidos
1. Describir el concepto de Metodología Ágil y sus orígenes para reconocer el contexto en el que se desempeña el Product Owner	1.1 Distingue las diferencias entre los conceptos de “Agile”, “Agilidad” y “Agilismo” 1.2 Reconoce las características del Manifiesto Ágil mediante la asociación de sus 4 valores con sus 12 principios 1.3 Identifica el abanico de Métodos, Metodologías o marcos de trabajo asociados a Agile	Metodologías Ágiles <ul style="list-style-type: none"> Orígenes Diferencias con Modelos Tradicionales de Desarrollo <ul style="list-style-type: none"> Waterfall RUP Modelo PMBOK Diferencias entre “Agile”, “Agilidad” y “Agilismo” Manifiesto Ágil Valores y Principios Abanico de Metodologías ágiles <ul style="list-style-type: none"> DSDM - Atern Lean Software Development Extreme Programming (XP) Kanban

<p>2. Describe los principios, prácticas, roles y artefactos del marco de trabajo Scrum para el desarrollo de productos en un entorno ágil de desarrollo</p>	<p>2.1 Reconoce los principios y elementos fundamentales del marco de trabajo Scrum para el desarrollo de productos tecnológicos</p> <p>2.2 Describe los roles y sus características dentro del marco de trabajo Scrum para el desarrollo de productos tecnológicos</p> <p>2.3 Explica las principales prácticas del marco de trabajo Scrum para el desarrollo de productos tecnológicos</p> <p>2.4 Describe los principales artefactos utilizados en el marco de trabajo Scrum para el desarrollo de productos tecnológicos</p>	<p>Fundamentos de Scrum</p> <ul style="list-style-type: none"> ● Qué es Scrum ● Principios y Valores de Scrum ● Roles <ul style="list-style-type: none"> ○ Scrum Master ○ Scrum Developers ○ Product Owner ● Prácticas <ul style="list-style-type: none"> ○ Sprint Planning ○ Daily Scrum ○ Review ○ Retrospective ● Artefactos <ul style="list-style-type: none"> ○ Product Backlog ○ Sprint Backlog ○ Incremento
<p>3. Reconocer las características más importantes en la definición de productos y en la construcción de una Visión de Producto de acuerdo al marco de trabajo Scrum</p>	<p>3.1 Reconoce los elementos asociados al producto y a la visión de acuerdo al marco de trabajo Scrum</p> <p>3.2 Reconoce las formas de definir alcance variable y controlar modelos de gestión de entregas.</p> <p>3.3 Describe las Características fundamentales de un MVP correcto</p>	<p>Visión del Producto en Scrum</p> <ul style="list-style-type: none"> ● Producto y su Visión ● Cómo Estimar ● Definición de Alcance ● Release Management ● Scrum Como modelo Iterativo y Evolutivo ● Mínimo Producto Viable
<p>4. Redactar historias de usuario utilizando técnicas para la creación de historias eficientes de acuerdo al marco de trabajo scrum</p>	<p>4.1 Identifica los elementos fundamentales de una historia de usuario de acuerdo al marco de trabajo scrum</p> <p>4.2 Redacta Historia de Usuario utilizando el Modelo INVEST para la creación de historias de usuario efectivas</p>	<p>Taller de Historias de Usuario</p> <ul style="list-style-type: none"> ● Qué es una historia de usuario ● Utilidad de las historias de usuario ● Escribiendo Historias de Usuario ● Modelo INVEST ● Modelo SMART

	4.3 Define Objetivos de negocio utilizando el modelo SMART	
5. Realizar estimación de esfuerzo a historias de usuario utilizando técnicas ágiles para reducir la incertidumbre respecto de modelos de estimación tradicionales	<p>5.1 Describe las principales características de la estimación de acuerdo al marco de trabajo scrum</p> <p>5.2 Explica por qué en modelos ágiles no se estima en HH</p> <p>5.3 Distingue diversas técnicas de estimación de esfuerzo identificando las ventajas y desventajas de cada una de ellas</p> <p>5.4 Realiza estimación aplicando técnicas de estimación ágil de acuerdo al marco de trabajo scrum</p>	<p>Taller de Estimación</p> <ul style="list-style-type: none"> ● Estimación en scrum ● Desechando la estimación en HH ● Modelos de Estimación Ágiles <ul style="list-style-type: none"> ○ Puntos de Historia ○ Tallas ○ Matriz de Incertidumbre ● Movimiento No Estimates
6. Describir técnicas de priorización para la gestión del backlog de historias de usuario permitiendo alinear el avance del producto con las necesidades del negocio	<p>6.1 Explica la importancia de la priorización de acuerdo al marco de trabajo scrum para alinear el avance del producto con la estrategia de negocio</p> <p>6.2 Distingue la diferencia entre cosas urgentes e importantes, así como aquellas que no lo son para la realización de una correcta priorización</p> <p>6.3 Describe los principales modelos de priorización de acuerdo al marco de trabajo scrum</p>	<p>Priorización</p> <ul style="list-style-type: none"> ● En qué consiste la práctica de priorización ● Diferenciar lo Urgente de lo Importante ● Modelos de Priorización <ul style="list-style-type: none"> ○ Visión de Negocio ○ Triage ○ Moscow ● Cómo Negociar prioridades
7. Describir las prácticas de refinamiento de Historias de Usuario para la generación de un entendimiento común	<p>7.1 Explica la importancia del proceso de refinamiento de historias dentro de la dinámica de trabajo ágil</p> <p>7.2 Describe las etapas del refinamiento y practicar la ejecución de una sesión</p>	<p>Refinamiento del backlog</p> <ul style="list-style-type: none"> ● Para qué sirve ● Importancia de Refinar Historias ● Cómo llevar sesiones de Refinamiento ● Malas prácticas

--	--	--

Estrategias Metodológicas para la implementación del módulo

El objetivo del presente módulo, es desarrollar en los participantes los aprendizajes relacionados con los aspectos fundamentales del marco de trabajo Scrum y de las metodologías ágiles, de forma que el participante pueda participar en equipos de desarrollo compartiendo los principios y valores ágiles.

Para la implementación de este módulo, se sugiere la aplicación de una metodología de aprendizaje activa, en donde el participante sea el principal actor en su proceso de aprendizaje. Es recomendable la utilización del método de casos, talleres prácticos y juegos de roles para simular las condiciones reales de un equipo de desarrollo, y de esta manera llevar a cabo las prácticas propuestas por el marco de trabajo.

Importante también es la interacción y el debate entre los mismos participantes dando espacio a la reflexión, la utilización de casos de estudios y discusión a partir de la experiencia de los mismos participantes, quienes al contar con experiencia profesional relevante, tendrán mucho que aportar y contribuir a las sesiones.

Finalmente, se realiza una autoevaluación o coevaluación entre los estudiantes, con el apoyo del instructor, basándose en los Insights obtenidos de los casos estudiados anteriormente. Los problemas propuestos en cada sesión deben ser contextualizados en situaciones reales, provenientes de diversas industrias, tales como retail, banca, seguros, utilities, salud, educación, entre otros, para de esta forma construir un aprendizaje significativo y contextualizado.

Desarrollo de conocimientos

Para el desarrollo de los conocimientos clave esperados por los estudiantes, se recomienda apoyar los ejemplos y casos de estudio con literatura formal y bibliografía de tendencia, además de variados ejercicios de auto aplicación, como experimentación controlada, gamificación y/o dinámicas de grupo, donde se puedan aplicar con el grupo de alumnos ciertos experimentos desarrollados por la ciencia y que fundamentan las bases del comportamiento.

Estrategia Evaluativa del módulo

La evaluación del módulo podrá ser durante y/o bien al final de éste. Para medir los conocimientos fundamentales, se sugiere la aplicación de una prueba de selección múltiple, donde pueda reconocer los elementos más significativos del aprendizaje, dándole una estructura a lo largo del proceso evaluativo. Asimismo, se pueden considerar instancias de notas por participación a lo largo de las sesiones y los talleres prácticos.

También se sugiere incorporar autoevaluaciones y coevaluaciones formativas durante el transcurso del módulo.

Perfil del Facilitador

Características Transversales

El instructor debe poseer competencias pedagógicas tales como:

- Planificar la clase
- Demostrar dominio de los temas tratados
- Capacidad de motivar
- Capacidad de contextualizar los temas en el área laboral
- Capacidad de cumplir un rol de facilitador
- Habilidades comunicacionales acordes a la función

Características Específicas

- Certificaciones atinentes al contexto educativo, idealmente relacionado al Coaching Agile.
- Experiencia como Docente.
- Experiencia como Facilitador de equipos u organizaciones.
- Experiencia como Charlista o Conferencista.
- Experiencia laboral de al menos 3 años en el área de proyectos ágiles.

Recursos materiales para la implementación del módulo formativo

Infraestructura	Equipos y Herramientas	Materiales e Insumos
<ul style="list-style-type: none"> • Sala de clases implementada con: Puestos de trabajo individuales que considere mesa y silla. Escritorio y silla para el facilitador. 	<ul style="list-style-type: none"> • Proyector multimedia • Notebook o PC, para el facilitador • Telón • Pizarra 	<ul style="list-style-type: none"> • Set de oficina, uno por participante, compuesto por: Carpeta o archivador. Cuaderno o croquera. Lápiz pasta. Lápiz grafito. Goma de borrar. Liquido corrector. Regla. • Plumones para pizarrón.

<ul style="list-style-type: none">• Conexiones para utilizar medios didácticos tales como data y salida a internet y salida de audio• Sistema de iluminación y ventilación adecuados• Servicios higiénicos separados para hombres y mujeres en recintos de aulas y de actividades prácticas.		<ul style="list-style-type: none">• Lista de participantes.• Carpeta de registro de evidencias para el facilitador• Material audiovisual que contenga información del curso:• Manual didáctico que contemple todos los contenidos especificados para este módulo, por cada participante.
--	--	---

Módulos Transversales Disciplinarios

Módulo Formativo N°5	
Nombre	Fundamentos de Integración Continua
N° de horas	15 horas
Requisitos de ingreso	Título Técnico / Profesional;
Competencia del módulo	<ul style="list-style-type: none"> Reconocer los conceptos fundamentales, principios y prácticas de la cultura DevOps distinguiendo el rol del desarrollador en un equipo ágil utilizando un pipeline de integración continua para la entrega de un producto

Aprendizajes esperados	Criterios de evaluación	Contenidos
1. Explicar los aspectos fundamentales del pensamiento DevOps y su impacto en el desarrollo de productos digitales	1.1 Explica los fundamentos del pensamiento DevOps, propósito, cultura y beneficios de su adopción en la organización 1.2 Explica los principios fundamentales de las prácticas DevOps para el desarrollo de un producto digital a lo largo de su ciclo de vida 1.3 Identifica las etapas del ciclo de vida de un producto tecnológico de acuerdo a las prácticas DevOps	Fundamentos de DevOps <ul style="list-style-type: none"> Qué es DevOps y cuál es su propósito Origen y Evolución Cultura y principios DevOps Beneficios de DevOps Modelo CAMS (Culture, Automation, Measurement, Sharing) DevOps y DevSecOps Ciclo de vida DevOps
2. Distinguir los conceptos de Integración Continua, Despliegue Continuo y Entrega Continua	2.1 Describe el concepto de Integración Continua dentro del ciclo DevOps distinguiendo sus diferencias con el	Integración Continua / Entrega Continua <ul style="list-style-type: none"> Qué es Integración Continua Despliegue Continuo v/s Entrega Continua

reconociendo sus características, prácticas y herramientas	<p>concepto de Despliegue Continuo y Entrega Continua</p> <p>2.2 Describe las prácticas DevOps utilizadas en el proceso de integración continua para dar soporte al proceso de desarrollo</p> <p>2.3 Explica las características de un sistema de integración continua reconociendo alternativas disponibles para su utilización</p>	<ul style="list-style-type: none"> Flujo del proceso de Integración Continua <ul style="list-style-type: none"> Control de versiones Compilación Testing Sistemas de integración Continua <ul style="list-style-type: none"> Qué es un sistema de integración continua Alternativas (Jenkins, Circle CI, GitLab CI, Bamboo)
3. Distinguir las prácticas relacionadas con Infraestructura y Operaciones de acuerdo al movimiento DevOps	<p>3.1 Describe el concepto de infraestructura evolutiva y su importancia en el ciclo DevOps</p> <p>3.2 Identifica los modelos de servicios cloud reconociendo sus principales características</p> <p>3.3 Explica el concepto de Infraestructura como código y su importancia en el ciclo DevOps</p> <p>3.4 Explica el concepto de monitoreo continuo identificando herramientas y stacks disponibles para el monitoreo</p>	<p>Infraestructura y Operaciones</p> <ul style="list-style-type: none"> Infraestructura <ul style="list-style-type: none"> Qué se entiende por infraestructura y por plataforma Infraestructura evolutiva Diferencias entre on-premise y cloud Importancia de la infraestructura en el ciclo DevOps Infraestructura como código El modelo de infraestructura cloud Modelos de servicio en la nube (IaaS, PaaS, SaaS) Operaciones y Escalamiento <ul style="list-style-type: none"> Problemas de la operación Orquestadores de contenedores Qué es Kubernetes Monitoreo <ul style="list-style-type: none"> Qué es el monitoreo Continuo Importancia del monitoreo continuo durante el ciclo DevOps Importancia del manejo de logs y eventos Stacks de monitoreo (ELK, GFG)

<p>4. Utilizar sistemas de control de versiones para la gestión del código fuente de acuerdo a las buenas prácticas DevOps</p>	<p>4.1 Explica el rol de un sistema de control de versiones dentro del ciclo de vida DevOps</p> <p>4.2 Utiliza Git para la gestión del código fuente en un repositorio local</p> <p>4.3 Identifica distintas modalidades de flujo de trabajo grupal utilizando GitHub</p> <p>4.4 Gestiona el trabajo colectivo en un repositorio remoto utilizando GitHub</p>	<p>Sistemas de Control de Versiones (SCV)</p> <ul style="list-style-type: none"> • Qué es un SCV • Problema que resuelve un SCV • Principales conceptos de un SCV (Repositorio, Diff, Commit, Branch, Merge, Clone, Fork) • Tipos de SCV y alternativas <ul style="list-style-type: none"> ○ Centralizados (SVN, CVS) ○ Distribuidos (Git, Mercurial) • Git como sistema de control de versiones <ul style="list-style-type: none"> ○ Qué es Git ○ Instalación, configuración y comandos básicos ○ Utilización de Git en repositorios locales ○ Commits y restauración de archivos ○ Ignorando archivos ○ Ramas, uniones, conflictos y tags ○ Stash y rebase • Centralización de repositorios <ul style="list-style-type: none"> ○ Servicios de centralización de repositorios (Gitlab, Github, Bitbucket) ○ Repositorios remotos, push y pull ○ Fetch v/s Pull ○ Clone y Fork de un repositorio ○ Flujos de trabajo típicos con Git ○ Pull Requests
<p>5. Utilizar tecnología de contenedores para la implementación de un ciclo de integración continuo</p>	<p>5.1 Reconoce el rol y la importancia de Docker en la infraestructura y pilar fundamental del ciclo DevOps</p>	<p>Contenedores de Aplicaciones</p> <ul style="list-style-type: none"> • Qué es un contenedor de aplicaciones • Diferencias entre un contenedor y una máquina virtual • El contenedor Docker

	<p>5.2 Reconoce conceptos y comandos básicos sobre contenedores Docker</p> <p>5.3 Reconoce la sintaxis, formato y comandos utilizados en un archivo Dockerfile para la creación de una imagen docker</p> <p>5.4 Utiliza imágenes dockerizadas para la implementación de servicios comúnmente utilizados</p>	<ul style="list-style-type: none"> ● Beneficios de utilizar un contenedor Docker ● Rol del contenedor Docker dentro del ciclo DevOps ● Conceptos básicos de Dockers <ul style="list-style-type: none"> ○ Images ○ DockerFile ○ Containers ○ Volumes ● Qué es un DockerFile y para qué sirve ● Implementación de Docker <ul style="list-style-type: none"> ○ Instalación y configuración ○ Comandos administrativos básicos ○ Utilización de imágenes ○ Monitoreo de eventos y logs asociados a contenedores
6. Implementar una batería de tests automatizados al ciclo de integración continua de acuerdo a las buenas prácticas DevOps	<p>6.1 Describe las características fundamentales del testing dentro de un entorno ágil de desarrollo</p> <p>6.2 Explica las características de los distintos tipos de prueba en un entorno de integración continua</p> <p>6.3 Describe los mecanismos y tipos de herramientas para realizar automatización de las pruebas en el ciclo de integración continua</p> <p>6.4 Implementa pruebas automatizadas en un ciclo de integración continua de acuerdo a las prácticas DevOps</p>	<p>Las pruebas en un entorno de integración continua</p> <ul style="list-style-type: none"> ● Testing en un entorno ágil <ul style="list-style-type: none"> ○ Qué es Agile Testing ○ Principios de Agile Testing ○ Prácticas relacionadas con Agile Testing ○ Rol del tester en un marco ágil ○ Qué es TDD ● Objetivo de las pruebas ● Tipos de prueba <ul style="list-style-type: none"> ○ Pruebas unitarias ○ Pruebas de integración ○ Pruebas de Sistema <ul style="list-style-type: none"> ▪ Funcionales ▪ Rendimiento ● El concepto de regresividad en las pruebas

		<ul style="list-style-type: none"> • Otros conceptos de prueba <ul style="list-style-type: none"> ○ Pruebas de aceptación ○ Pruebas de humo • Automatización de las pruebas <ul style="list-style-type: none"> ○ Objetivo de la automatización ○ Herramientas para la automatización de las pruebas ○ Incorporación de la automatización de pruebas al pipeline de integración continua
7. Implementar un pipeline de integración continua utilizando Jenkins de acuerdo a las buenas prácticas DevOps	<p>7.1 Describe las características de Jenkins, su rol y beneficios dentro del ciclo de integración continua</p> <p>7.2 Configura plugins, usuarios, roles y permisos en plataforma Jenkins para su posterior utilización en el ciclo de integración continua</p> <p>7.3 Configura Jobs y Pipelines utilizando Jenkins para su posterior utilización en el ciclo de integración continua</p> <p>7.4 Configura Pipeline de integración continua como código utilizando JenkinsFile para su posterior utilización en el ciclo CI/CD</p>	<p>Implementación de un Pipeline CI</p> <ul style="list-style-type: none"> • Qué es Jenkins y para qué sirve • Rol de Jenkins dentro de la Integración Continua • Instalación y configuración de Jenkins • Actualización de plugins • Configuración de usuarios, roles y permisos • Definición de pipelines de integración con código: JenkinsFile. Qué es y para qué se usa • Utilización de Jenkins en el ciclo CI/CD <ul style="list-style-type: none"> ○ Creación y configuración de un job en Jenkins ○ Creación y configuración de pipeline de Jenkins mediante código JenkinsFile ○ Creación y configuración de despliegue sobre el ambiente de pruebas

Estrategias Metodológicas para la implementación del módulo

El objetivo del presente módulo, es desarrollar en los participantes los aprendizajes fundamentales del pensamiento DevOps y la utilización de las prácticas DevOps relacionadas con el desarrollo de software, para lo cual se espera que al finalizar este módulo, los participantes sean capaces

también de implementar un pipeline de integración continua con los componentes básicos que se mencionan. Este módulo es común para todos los planes formativos relacionados con desarrollo, por lo cual cada especialidad debe utilizar sus entornos de desarrollo definidos en cada curso (Java, JavaScript, Python).

Se espera que, en este módulo, los participantes sean capaces de desplegar sus soluciones construidas a lo largo del curso en un entorno (por ejemplo, testing, productivo, etc). Es altamente recomendable, que los participantes cuenten con acceso a alguna nube (Amazon, Google, Azure) de tal forma que puedan utilizar las herramientas disponibles para la implementación de los conceptos de integración continua y puedan cumplir dicho propósito.

Cabe aclarar, que los aprendizajes expuestos en este documento están ordenados de acuerdo a su nivel taxonómico (describir, utilizar, implementar, diseñar) y no dan cuenta de una secuencialidad en los aprendizajes. Es desafío del diseño instruccional de definir una secuencia apropiada para el desarrollo de los aprendizajes acá mencionados.

Para la implementación de este módulo, se sugiere la aplicación de una metodología de aprendizaje activa, en donde el participante sea el principal actor en su proceso de aprendizaje. Es recomendable la utilización del método de casos, talleres prácticos y juegos de roles para simular las condiciones reales de un equipo de desarrollo, y de esta manera llevar a cabo las prácticas propuestas por el marco de trabajo.

Importante también es la interacción y el debate entre los mismos participantes dando espacio a la reflexión, la utilización de casos de estudios y discusión a partir de la experiencia de los mismos participantes, quienes al contar con experiencia profesional relevante, tendrán mucho que aportar y contribuir a las sesiones.

Finalmente, se realiza una autoevaluación o coevaluación entre los estudiantes, con el apoyo del instructor, basándose en los Insights obtenidos de los casos estudiados anteriormente. Los problemas propuestos en cada sesión deben ser contextualizados en situaciones reales, provenientes de diversas industrias, tales como retail, banca, seguros, utilities, salud, educación, entre otros, para de esta forma construir un aprendizaje significativo y contextualizado.

Estrategia Evaluativa del módulo

La evaluación del módulo podrá ser durante y/o bien al final de éste. Para medir los conocimientos fundamentales, se sugiere la aplicación de una prueba de selección múltiple, donde pueda reconocer los elementos más significativos del aprendizaje, dándole una estructura a lo largo del proceso evaluativo. Asimismo, se sugiere al finalizar el módulo la evaluación de un producto entregado por cada participante que de cuenta de los aprendizajes adquiridos a lo largo del módulo.

También se sugiere incorporar autoevaluaciones y coevaluaciones formativas durante el transcurso del módulo.

Perfil del Facilitador

Características Transversales

El instructor debe poseer competencias pedagógicas tales como:

- Planificar la clase
- Demostrar dominio de los temas tratados
- Capacidad de motivar
- Capacidad de contextualizar los temas en el área laboral
- Capacidad de cumplir un rol de facilitador
- Habilidades comunicacionales acordes a la función

Características Específicas

- Ingeniero en Informática, civil o ejecución, o similar
- Haber aprobado cursos formales de capacitación en el área DevOps o bien certificaciones de la industria
- Experiencia docente en relatoría de cursos del área tecnológica
- Experiencia laboral de al menos 2 años en el área de DevOps
-

Recursos materiales para la implementación del módulo formativo

Infraestructura	Equipos y Herramientas	Materiales e Insumos
<ul style="list-style-type: none">• Sala de clases implementada con: Puestos de trabajo individuales que considere mesa y silla. Escritorio y silla para el facilitador.	<ul style="list-style-type: none">• Proyector multimedia• Notebook o PC, para el facilitador• Telón• Pizarra	<ul style="list-style-type: none">• Set de oficina, uno por participante, compuesto por: Carpeta o archivador. Cuaderno o croquera. Lápiz pasta. Lápiz grafito. Goma de borrar. Liquido corrector. Regla.

<ul style="list-style-type: none">• Conexiones para utilizar medios didácticos tales como data y salida a internet y salida de audio• Sistema de iluminación y ventilación adecuados• Servicios higiénicos separados para hombres y mujeres en recintos de aulas y de actividades prácticas.		<ul style="list-style-type: none">• Plumones para pizarrón.• Lista de participantes.• Carpeta de registro de evidencias para el facilitador• Material audiovisual que contenga información del curso:• Manual didáctico que contemple todos los contenidos especificados para este módulo, por cada participante.
--	--	---

Módulos Transversales

Módulo Formativo N°1	
Nombre	Desarrollo de Portafolio de un Producto Digital
N° de horas	6 horas
Requisitos de ingreso	Experiencia Laboral; Test de Selección;
Competencia del módulo	Presentar un producto digital en un portafolio personal utilizando las herramientas tecnológicas y buenas prácticas disciplinares para resolver una problemática y potenciar el perfil profesional

Aprendizajes esperados	Criterios de evaluación	Contenidos
1. Reconocer las características fundamentales de un portafolio de productos digitales efectivo para potenciar el perfil profesional	1.1 Reconoce la importancia de un portafolio de productos como herramienta para potenciar el perfil profesional 1.2 Describe las buenas prácticas para la creación de un portafolio de productos 1.3 Identifica herramientas para la implementación de un portafolio de productos	El portafolio de productos <ul style="list-style-type: none"> • Qué es un portafolio de productos • Importancia de contar con un portafolio • Buenas prácticas para la creación de un portafolio de productos • Herramientas que se pueden utilizar para la creación del portafolio de productos <ul style="list-style-type: none"> ○ GitHub ○ Hosting ○ Página web personal ○ Youtube ○ Otros

<p>2. Elaborar un producto tecnológico para resolver una problemática real utilizando las buenas prácticas de la disciplina</p>	<p>2.1 Distingue conceptos y buenas prácticas para el diseño de un producto digital que resuelve un problema</p> <p>2.2 Utiliza técnicas y herramientas de la disciplina para la implementación de un producto digital Ajustes finales y cierre de entregable</p> <p>2.3 Implementa un producto funcional que resuelve un problema real utilizando las buenas prácticas de la disciplina</p>	<p>Finalización del proyecto</p> <ul style="list-style-type: none"> ● Revisión del producto construido a lo largo del curso ● Depuración y mejora del producto ● Feedback y retroalimentación ● Ajustes finales y cierre de entregable
<p>3. Implementar un portafolio de un producto digital utilizando las herramientas disponibles y las buenas prácticas de la industria</p>	<p>3.1 Distingue las competencias que serán adquiridas a lo largo de cada módulo de la currículo</p> <p>3.2 Reconoce la naturaleza del trabajo que será realizado junto con las herramientas que serán utilizadas a lo largo de cada módulo</p> <p>3.3 Reconoce la importancia de un portafolio de producto así como sus características para la formación de una identidad profesional</p> <p>3.4 Identifica los productos que serán obtenidos en cada módulo así como su contribución al portafolio de producto</p>	<p>Utilización de GitHub para crear un portafolio (*)</p> <ul style="list-style-type: none"> ● Qué es GitHub ● Características y operaciones básicas de GitHub ● Buenas prácticas para tener un portafolio atractivo en GitHub ● Buenas prácticas para la página principal de un repositorio ● Buenas prácticas para la página de perfil <p>Utilización de Behance para crear un portafolio (**)</p> <ul style="list-style-type: none"> ● Qué es Behance ● Características y operaciones básicas de Behance ● Buenas prácticas para tener un portafolio atractivo en Behance ● Behance dedicado y exclusivo a proyectos de UX <p>Alojamiento de tu producto en un servidor</p> <ul style="list-style-type: none"> ● Qué es un Hosting

		<ul style="list-style-type: none"> • Servicios gratuitos de hosting • Servicios cloud gratuitos • Cómo alojar un proyecto UX/UI • Cómo alojar un proyecto Front-End • Cómo alojar un proyecto Fullstack Java • Cómo alojar un proyecto Fullstack Javascript • Cómo alojar un proyecto Fullstack Python • Cómo alojar un proyecto Android <p>(*) Aplica sólo para cursos de desarrollo (**) Aplica sólo para curso de diseño UX/UI</p>
--	--	---

Estrategias Metodológicas para la implementación del módulo

El presente módulo tiene por objeto la finalización del producto final, que se ha venido implementando a lo largo de todos los módulos de este plan formativo, y que servirá como base para implementar el portafolio de producto.

El participante trabajará de forma activa durante este módulo en el cierre de su producto final, en donde el facilitador adoptará un rol de apoyo y de guía. Se deberá hacer una debida planificación de las sesiones de este módulo de forma de ofrecer el apoyo requerido a todos los participantes para que completen su producto final. Es altamente probable que se requiera tiempo de dedicación personal del participante fuera de las horas lectivas para completar su trabajo, por lo tanto se recomienda planificar días de receso para el avance individual y sesiones de retroalimentación, así como habilitar un canal de consulta en caso que los participantes queden estancados en su avance personal.

Asimismo, a lo largo de este módulo, se desarrollarán también aprendizajes para la implementación de un portafolio de producto, los cuales pueden llevarse a cabo mediante breves actividades de exposición y demostración, pero que tendrá un fuerte componente práctico cuando el participante implemente su propio portafolio.

Estrategia Evaluativa del módulo

La estrategia evaluativa consistirá en Evaluación de Producto, el cual se llevará a cabo mediante una rúbrica que debe ser confeccionada oportunamente de acuerdo al plan formativo en cuestión.

Perfil del Facilitador

Características Transversales

El instructor debe poseer competencias pedagógicas tales como:

- Planificar la clase
- Demostrar dominio de los temas tratados
- Capacidad de motivar
- Capacidad de contextualizar los temas en el área laboral
- Capacidad de cumplir un rol de facilitador
- Habilidades comunicacionales acordes a la función

Características Específicas

- Formación profesional en el área de las ingenierías, como diseñador para el caso del plan UX/UI. En todos los casos anteriores, con título
- Experiencia laboral de mínimo 3 años desempeñando funciones de laborales ya sea como desarrollador, científico de datos, o diseñador UX/UI, dependiendo del plan formativo, o bien gerenciando áreas en estas materias
- Experiencia como facilitador/a de capacitación para personas adultas, y/o bien como docente en instituciones de educación superior, sean universidades, institutos profesionales o centros de formación técnica, de mínimo 100 horas cronológicas, demostrable

Recursos materiales para la implementación del módulo formativo

Infraestructura	Equipos y Herramientas	Materiales e Insumos
-----------------	------------------------	----------------------

<ul style="list-style-type: none"> • Laboratorio computacional que cuente al menos con 1,5 m2 por participante, implementada con: Puestos de trabajo individuales que considere mesa y silla. Escritorio y silla para el facilitador. • Conexiones para utilizar medios didácticos tales como data y salida a internet. • Sistema de iluminación y ventilación adecuados • Servicios higiénicos separados para hombres y mujeres en recintos de aulas y de actividades prácticas. 	<ul style="list-style-type: none"> • Proyector multimedia • Notebook o PC, para el facilitador • Telón • Pizarra • Equipo computacional, uno para cada alumno, al menos Pentium Core i5, 8 GB RAM y 500 GB HD, y conexión a internet, con el siguiente software instalado: 	<ul style="list-style-type: none"> • Set de oficina, uno por participante, compuesto por: Carpeta o archivador. Cuaderno o croquera. Lápiz pasta. Lápiz grafito. Goma de borrar. Liquido corrector. Regla. • Plumones para pizarrón. • Lista de participantes. • Carpeta de registro de evidencias para el facilitador • Material audiovisual que contenga información del curso: • Manual didáctico que contemple todos los contenidos especificados para este módulo, por cada participante.
---	---	--