

Programa

DESARROLLO FRONT-END

N° créditos: 13

Unidad de
competencia

Crear desde cero una aplicación web haciendo énfasis en las interfaces del usuario. Desarrollar aplicaciones web en JavaScript, utilizando una librería de código abierto (React JS), que actualmente es la más competitiva en el mercado. Diseñar soluciones web rápidas y flexibles, de gran demanda en el mercado para su propia tienda e-commerce.

Sumilla del programa

Este programa de especialización, de naturaleza teórico – práctico, le permitirá al estudiante aprender a construir interfaces web adaptables a distintos dispositivos, analizando previamente un diseño definido con un software de diseño visual. Además, definirán soluciones desde las bases de HTML, CSS y JS, utilizando como complemento la librería de React JS. Por lo tanto, el estudiante aprenderá la lógica de programación, patrones de diseño, buenas prácticas de desarrollo, conocerá a fondo el lenguaje JavaScript, para luego utilizar React JS orientado a componentes con una arquitectura SPA con definición de rutas, manejo de estado y manejo de datos síncronos y asíncronos.

Logro del
programa

Al finalizar el curso, el estudiante desarrolla su propia tienda e-commerce utilizando los componentes de HTML, CSS, JavaScript y React JS.

Programación

MÓDULOS	SEMANAS	LECCIONES	CONTENIDOS
MÓDULO 1: INTRODUCCIÓN A LA WEB Reconoce los conceptos básicos y el funcionamiento del desarrollo web para la creación de un sitio web.	Semana 1: Introducción al desarrollo web Reconoce las bases y la configuración de su entorno de trabajo incluyendo el uso de Visual Studio Code para la elaboración de páginas web.	1. ¿Qué es Web y cómo funciona?	<ul style="list-style-type: none"> • ¿Cómo funciona la web? • Web vs Internet • Clientes y Servidores • ¿Cómo están construidas las páginas web? • Front-End y Back-End.
		2. Herramientas de desarrollo y configuración de entorno de trabajo.	<ul style="list-style-type: none"> • Editor de Código • Instalación y entorno de trabajo del editor VSCODE • Extensión para VSCODE. • Creando un proyecto en VSCODE • Herramienta de Desarrollador de Chrome
		3. Uso de la terminal y línea de comandos.	<ul style="list-style-type: none"> • ¿Qué es la Línea de comandos? • Comandos básicos de Linux. • Anatomía de un comando. • Trabajando con Directorios y Carpetas. • Creando Archivos y Carpetas
	Hackathon 1: El estudiante podrá configurar su entorno de trabajo. Además establecerá, mediante línea de comando, la creación de las carpetas en donde se va a ejecutar el proyecto.		
MÓDULO 2: SISTEMA DE CONTROL DE VERSIONES Y FUNDAMENTOS DE HTML Emplea el control de versiones a través del Git, estructuración de	Semana 2: CONTROL DE VERSIONES Y HTML Emplea el sistema de control de versiones para la creación de una web con HTML.	1. Git	<ul style="list-style-type: none"> • ¿Qué es Git? • Instalación y configuración. • Comandos básicos de Git • Creación de repositorios y ramas. • Flujos de cambios • Archivo gitignore

proyecto y las etiquetas HTML.		2. GitHub y GitHub Pages	<ul style="list-style-type: none"> ¿Qué es un repositorio remoto? Configuración de repositorio remoto Flujo de trabajo de repositorio remoto GitHub Pages
		3. Prototipado y conceptos básicos de HTML	<ul style="list-style-type: none"> Proyecto web: ¿qué es? Prototipado y estructura de una Web. ¿Qué es HTML? DOM. Etiquetas y Atributos HTML Semántico. Listas, Tablas. Enlaces y Rutas
		Hackathon 2: El estudiante podrá establecer el diseño, aplicará la estructura HTML de su Proyecto y reforzará sus conocimientos en HTML y las principales etiquetas.	
MÓDULO 3: MAQUETACIÓN WEB Aplica una interfaz adaptable para visualizarlo desde cualquier dispositivo.	Semana 3: MANEJO DE FORMULARIOS E INCLUYENDO CSS EN NUESTRO PROYECTO Emplea formularios, estilos y reglas de apariencia para cada elemento HTML.	1. Formularios e Input	<ul style="list-style-type: none"> ¿Qué es un formulario HTML? Etiquetas de formulario Tipos de input
		2. Validación de Formularios	<ul style="list-style-type: none"> Validaciones de formulario y atributos de validación Patrones de validación Pseudo clases de validación
		3. Fundamentos de CSS	<ul style="list-style-type: none"> ¿Qué es CSS? Agregando CSS Reglas de CSS Selectores Layout Display – Flujo Elementos en Línea y Bloque
		Hackathon 3: El estudiante aplicará estilos a la interfaz o visualización de su proyecto. Además de generar una interfaz amigable con ayuda de formulario para mejorar la interacción con el usuario.	

	Semana 4: CONCEPTOS ESENCIALES DE CSS Utiliza CSS para aplicarlos en el proyecto web.	1. CSS: Medidas, colores y fuentes	<ul style="list-style-type: none"> • Cascada, Especificidad y Herencia • Unidades de medida • Colores. • Listas • Tipografías web y fondos
		2. CSS: Aplicando estilos al contenido de nuestro sitio	<ul style="list-style-type: none"> • Box Model • Propiedades de la caja. • Posicionamiento y visualización. • Fuentes personalizadas • Google Fonts
		3. CSS: Aplicando Animaciones, transformaciones y transiciones	<ul style="list-style-type: none"> • Bordes redondeados • Box shadow • Text shadow • Degradados • Transiciones • Animaciones • Transform
	Hackathon 4: El estudiante deberá generar animaciones dentro de su proyecto para generar una mejor visualización del proyecto, así como deberá finalizar las tipografías, colores y unidades de medidas que aplicará en las Hojas de Estilo		
	Semana 5: Organización del Layout Utiliza menos código dentro del proyecto organizado para mejorar la posición de los elementos en el layout.	1. Flexbox	<ul style="list-style-type: none"> • ¿Qué es Flexbox? • Propiedades de padres e hijos. • Propiedades para el contenedor • Alineación • Flex, y los Flex ítems.
		2. CSS Grid I	<ul style="list-style-type: none"> • ¿Qué es CSS Grid? • Display • Columnas y filas • Áreas • Espacios entre columnas y filas • Implementación de Grids
		3. CSS Grid II	<ul style="list-style-type: none"> • Posicionamiento de ítems Grid explícito e implícito • Flujo de grillas • Unidades de medida, repeat, auto fit, auto fill, minmax

Hackathon 5: El estudiante va a establecer las propiedades de cada contenedor de la Web en la cual podrá utilizar Flexbox o CSS Grid.			
	Semana 6: Pre Procesadores (SASS) y Metodologías CSS Utiliza estas metodologías para la creación de hojas de estilos con características que no tiene CSS.	1. SASS I	<ul style="list-style-type: none"> ¿Qué es SASS? Configuración de compilador Variables Comments Nesting Selector Parent Partials e @import Extends / inheritance Interpolation
		2. SASS II	<ul style="list-style-type: none"> Operators Mixings Functions Content block Control de flujo Listas y Mapas Iteradores Funciones de SASS
		3. Metodologías CSS	<ul style="list-style-type: none"> ¿Qué es una metodología CSS? OOCSS BEM SMACSS ATOMIC DESIGN
	Hackathon 6: El estudiante tendrá la responsabilidad de aplicar metodologías de CSS. Además de utilizar SASS para la mejora de su Código CSS y que sea legible y mantenible.		
	Semana 7: Diseño Responsivo y Bootstrap Aplicar un framework de CSS para la creación de sitios web de una manera más rápida y estandarizada.	1. Diseño Responsivo	<ul style="list-style-type: none"> ¿Qué es RWD? Mobile First Media Queries Funciones multimedia Técnicas Responsive
		2. Frameworks CSS + Bootstrap	<ul style="list-style-type: none"> ¿Qué es un framework? Funcionalidades de Bootstrap.
		3. Bootstrap + Themes	<ul style="list-style-type: none"> Bootstrap con JS. Bootstrap Themes. Diseño responsive.
	Hackathon 7: El estudiante podrá mejorar el diseño de su proyecto haciendo uso de Bootstrap el cual le permitirá visualizar su proyecto desde cualquier dispositivo.		
	MÓDULO 4: PROGRAMACIÓN EN JAVASCRIPT Crea interfaces interactivas haciendo uso de JavaScript mediante funciones e	Semana 8: Lógica de Programación e Introducción a Javascript Desarrolla sus propios algoritmos haciendo uso de la	1. Lógica de programación
			<ul style="list-style-type: none"> ¿Qué es un algoritmo? Pseudocódigo y Simbología Estructura de Pseudocódigo Diagrama de flujo

iteraciones para la manipulación de datos.	lógica de programación para mejorar el funcionamiento de la aplicación.		<ul style="list-style-type: none"> Entrada y Salida Operador de asignación Operadores aritméticos
		2. Fundamentos de JavaScript	<ul style="list-style-type: none"> Presentación de JavaScript: ¿para qué se usa? Sintaxis Uso de la consola. Variable: ¿qué es y cómo declararla? Asignación y cambio del valor. Control de Flujo Condicional: ¿qué es? Operadores de comparación y lógicos: ¿qué son y cuál es su uso en los condicionales?
		3. Ciclos/iteraciones	<ul style="list-style-type: none"> Ciclos: ¿qué son? Tipos y diferencias entre sí. Operadores lógicos, ciclos y funciones: ¿cómo combinarlos? If Else If Else If Switch Ciclo for Ciclo con for Ciclo While
	Hackathon 8: Con ayuda de JavaScript el estudiante podrá generar la interacción del su proyecto así como crear los primeros algoritmos para el funcionamiento de su E-commerce.		
	Semana 9: Funciones con JavaScript Desarrolla un procedimiento mediante un conjunto de instrucciones para realizar una tarea o cálculo.	1. Funciones	<ul style="list-style-type: none"> ¿Qué es una función? ¿Qué son los parámetros de entrada y de salida? Scope global y local. Función anónima Función flecha
		2. Objetos	<ul style="list-style-type: none"> ¿Qué son los objetos y cómo se usan? ¿Cuáles son sus propiedades y métodos? Función constructora.

		3. Arrays	<ul style="list-style-type: none"> • Métodos de Array • ForEach • Filter • Manipulando strings • Math y date
	Hackathon 9: El estudiante podrá generar instrucciones para realizar cálculos o tareas específicas con respecto al funcionamiento de su E-commerce.		
	Semana 10: Programación Orientada a Objetos y Asincronismo Emplea POO para la manipulación de datos y reutilización del código.	1. POO	<ul style="list-style-type: none"> • Paradigmas de programación • POO y la abstracción • Objetos literales • Asignación de atributos • Notación ES6 • Atributos y métodos
		2. Clases y objetos	<ul style="list-style-type: none"> • Redundancia • Función constructora • Clases • Métodos • Herencia
		3. Asincronismo y peticiones	<ul style="list-style-type: none"> • Temporizadores: setInterval y setTimeout • Control de errores: try-catch-finally • Asincronismo: async – await • Peticiones • Fetch
	Hackathon 10: Hacer uso de la Programación Orientada a Objetos para la manipulación de datos y aplicar una reutilización de código. Así como también emplear el asincronismo dentro de la aplicación.		
	Semana 11: Manipulación del localStorage, DOM y Eventos Desarrolla el almacén de información con el local Storage y la manipulación del DOM mediante eventos.	1. Storage y JSON	<ul style="list-style-type: none"> • localStorage y sessionStorage. • JSON: alcance del formato y situaciones de uso.
		2. DOM	<ul style="list-style-type: none"> • DOM: Definición, alcance y su importancia para operar sobre elementos HTML.
		3. Eventos	<ul style="list-style-type: none"> • ¿Qué son los eventos, para qué sirven y cuáles son los más comunes? • Escuchar un evento sobre el DOM. • Uso de selectores. • Agregar elementos al DOM.
	Hackaton 11: El estudiante podrá manipular del DOM y crear eventos que de la mano con las animaciones mejorará la visualización del proyecto.		

MÓDULO 5: REACT JS Elabora una aplicación con React JS a través del JavaScript Avanzado y consumiendo APIs.	Semana 12: NodeJS y NPM Aplica NodeJS y NPM para la ejecución de varios comandos como instalar y desinstalar paquetes.	1. NodeJS y NPM	<ul style="list-style-type: none"> ¿Qué es Node.js? ¿Qué es npm? Comandos npm package.json ¿Qué es npx? Paquetes dev Paquetes prod Paquetes globales Proyecto usando NPM.
		2. Package Manger	<ul style="list-style-type: none"> Gestión de Paquetes
		3. API	<ul style="list-style-type: none"> ¿Qué es una API? Como funciona una API
	Hackaton 12: El estudiante podrá aprenderá como es el funcionamiento de la API así como usar NodeJS dentro de proyecto y convertirlo en una Aplicación Web.		
	Semana 13: HTTP Verbs, Callbacks Aplica promesas y CORS para representar la terminación o el fracaso de una operación asíncrona.	1. HTTP Verbs	<ul style="list-style-type: none"> GET HEAD POST PUT PATCH DELETE
		2. Callbacks y Promises	<ul style="list-style-type: none"> ¿Qué es un callback? Callback Hell ¿Qué son las promesas? Manejando las promesas Encadenamiento de promesas Otros métodos de promesas
		3. CORS	<ul style="list-style-type: none"> ¿Cómo funciona el CORS? Estructura de la CORS header
	Hackaton 13: El estudiante aprenderá usar los HTTP Verbs para determinar que acción se va a realizar y complementarla con los Callbacks y Promesas para indicar la finalización o el fracaso de un proceso dentro de la aplicación.		
	Semana 14: React JS Utiliza React JS para instalarlo y configurarlo en el proyecto web.	1. Frameworks	<ul style="list-style-type: none"> ¿Qué es un Framework? Diferentes Framework Como utilizar un Framework
		2. React JS	<ul style="list-style-type: none"> ¿Qué es React JS? ¿Cómo se usa?

		3. Instalación y configuración del entorno	<ul style="list-style-type: none"> • Funcionamiento del virtual DOM en React JS. • Instalación y configuración de Node.js. • Creación de una app utilizando el CLI.
	Hackaton 14: El estudiante va a crear su proyecto en React JS en el cual tendrá que configurar para iniciar con la creación de su Aplicación Web.		
	Semana 15: JSX y Componentes Aplica componentes para reutilizarlos dentro de la aplicación.	1. JSX y transpiling	<ul style="list-style-type: none"> • JSX. • Expansión de la sintaxis avanzada de JS. • Rol de webpack y babel en el bundling/retrocompatibilidad.
		2. Componentes I	<ul style="list-style-type: none"> • ¿Qué son? • ¿Cuáles problemas resuelven? • ¿Cómo implementarlos?
		3. Componentes II	<ul style="list-style-type: none"> • Estados y ciclos de vida. • Propiedades y eventos.
	Hackaton 13: El estudiante realizará la migración del HTML y CSS en React para ir creando cada componente de la Aplicación y poder reutilizar cada componente.		
	Semana 16: Consumo de API, promesas y asincronía. Aplica APIs para mostrar la información y manejo de la navegación de la aplicación.	1. Promises, asincronía y MAP	<ul style="list-style-type: none"> • API Promise. • Aplicación del método MAP para el rendering de listas.
		2. Consumiendo APIs	<ul style="list-style-type: none"> • Paradigma de intercambio de datos. • Consumo de recursos vía llamadas a APIs.
		3. Routing y navegación	<ul style="list-style-type: none"> • Organización de la app. • Configuración de la navegabilidad entre componentes.
	Hackaton 14: El estudiante creará las promesas dentro de su aplicación para que pueda indicar la finalización o el fracaso de un proceso mediante el uso de API. Además de mejorar la navegación de la Aplicación.		

	Semana 17: Interacción con Eventos Desarrolla acciones de cómo debe actuar nuestra aplicación en caso de que ocurra un acontecimiento dentro de la aplicación.	1. Eventos	<ul style="list-style-type: none"> Sistema de eventos de React JS. Diseño de componentes orientados a eventos.
		2. Context	<ul style="list-style-type: none"> Organización de eventos cada proyecto. Creación de interacciones persistentes entre componentes. Desarrollo de implementaciones custom de context.
		3. Técnicas de rendering	<ul style="list-style-type: none"> Renderizado condicional y sus implicancias. Diagnóstico y solución de problemas de rendering.
		Hackaton 15: Por último el estudiante finalizará con el desarrollo de acciones o soluciones de problemas con respecto al rendering de la aplicación.	
MÓDULO 6: FEEDBACK, ASESORÍA Y PRESENTACIÓN DEL PROYECTO FINAL	Semana 18: Proyecto Final	1. Asesorías virtuales	<ul style="list-style-type: none"> Herramientas para la gestión de proyectos Recomendaciones Feedback sobre proyecto final
		2. Asesorías virtuales	<ul style="list-style-type: none"> Herramientas para la gestión de proyectos Recomendaciones Feedback sobre proyecto final
		3. Presentación del Proyecto Final	<ul style="list-style-type: none"> Temas Presentaciones Evaluación y Feedback Entrega de certificado Idat.

Sistema de Evaluación

EC	EF
Evaluación continua	Evaluación final
Peso	Peso
40%	60%

Referencias bibliográficas

Bibliografía
virtual
básica

Base de datos E-Libro:

- Mozilla (2023) Resources for Developers, by Developers, Mozilla Firefox. Recuperado el 26 de enero de 2023. <https://developer.mozilla.org/es/>
- Mozilla (2023) JavaScript, Mozilla Firefox. Recuperado el 26 de enero de 2023. <https://developer.mozilla.org/es/docs/Web/JavaScript>
- React (2023) React. Recuperado el 26 de enero de 2023. <https://es.reactjs.org/>