

## Fundamentos del desarrollo web

### Objetivo:

El estudiante obtendrá las habilidades necesarias para desarrollar de manera correcta la parte visual de un sitio web responsivo sin utilizar librerías/frameworks, así como agregar contenido dinámico con javascript, trabajar de manera colaborativa con git y finalmente entenderá y aplicará conceptos de accesibilidad.

El estudiante también obtendrá conocimientos de programación funcional y programación orientada a objetos.

### Requisitos

Uso de computadora:

- Instalar programas
- Crear/Eliminar/Acceder archivos
- Navegar a través de internet

El alumno no necesita conocimientos previos de programación, sin embargo, tener conocimientos básicos de html, css y javascript le facilitarán mucho el aprendizaje.

### Semana 1 - Introducción a Javascript

Objetivo: Tener el primer acercamiento a la programación con javascript y mejorar nuestro razonamiento lógico.

Temas

- Razonamiento lógico
- Rutas de archivos
- Variables
- Tipos de datos
- Operadores de comparación
- Operadores lógicos
- Métodos para strings
- Funciones
- Convenciones JS
- Webstorm y Stepik

### Semana 2 - Javascript básico

Objetivo: Tener las herramientas y habilidades necesarias para resolver problemas de dificultad media

Temas

- Tipos de datos
- Condicionales
- Operadores aritméticos
- Ciclos
- Métodos para arreglos
- Debug

### **Semana 3 - Bases HTML y CSS**

Objetivo: Comprender cómo funciona un navegador, elementos HTML y CSS básico.

#### Temas

- Cómo funciona el internet
- Navegadores
- Elementos y semántica HTML
- Atributos de elementos HTML
- Convenciones para nuestro HTML
- Introducción a CSS
- Selectores
- Pseudo-clases y pseudo-elementos
- Combinación de selectores
- Especificidad
- Propiedades css más utilizadas

Entregable:

Desarrollo de componentes básicos con HTML y CSS.

### **Semana 4 - CSS intermedio**

Objetivo: Desarrollar componentes con dificultad media y organizarlos de manera adecuada para construir un sitio web responsivo. También se aprenderán a desarrollar animaciones básicas con css.

#### Temas:

- Modelo de caja
- Posicionamiento
- Flexbox
- Diseño responsivo
- Animaciones básicas
- Primer sitio web: nuestro portafolio

Entregable: Desarrollo de nuestro propio grid con Flexbox

### **Semana 5 - Git y Accesibilidad web**

Objetivo: Terminar el desarrollo de nuestro portafolio con contenido accesible para personas con discapacidades, así como comprender el propósito y uso de git.

#### Temas:

- Accesibilidad web
- Git
- Github
- Repositorios remotos
- Despliegue de mi portafolio en Netlify
- Trabajo colaborativo con git
- Proyecto panel de administración en equipo

Entregable: Despliegue en Netlify de nuestro portafolio web terminado

### **Semana 6 - Javascript intermedio**

Objetivo: Comprender conceptos intermedios-avanzados de Javascript y aplicarlos al desarrollo web.

Temas:

- Tipos de funciones
- Call Stack
- Callbacks
- Closures
- Scope
- Métodos nativos en javascript
- Eventos
- Trabajando el DOM con javascript
- Módulos
- JSON
- Almacenamiento en el navegador

Entregable: Sitio web con CRUD de elementos con Javascript, el sitio web debe contar con login y mantener la sesión iniciada con ayuda del almacenamiento del navegador.

### **Semana 7 - Programación orientada a objetos y consumo de API's**

Objetivo: Comprender lo necesario para consumir una API aplicando programación orientada a objetos.

Temas

- Introducción a POO
- POO
- Promesas
- Async-Await
- Fetch
- HTTP
- Consumo de API
- Presentación del proyecto final

Entregable: Sitio web consumiendo API con POO

### **Proyectos que se agregaran a su portafolio:**

1. Desarrollo de portafolio web:  
El alumno desarrolla y publica su portafolio en internet con lo aprendido en el curso.
2. Juego Trivia:  
El alumno consume información de una API ([Trivia API](#)) para mostrar un juego de preguntas y respuestas, donde el usuario puede aplicar filtros como cantidad de

preguntas, categoría y tipo de pregunta y finalmente muestra los aciertos y errores al usuario.

## **Desarrollo de aplicaciones web con React**

**Objetivo:** El estudiante aprenderá a crear aplicaciones web con React JS, logrando una gran interacción con el usuario usando los conocimientos previos de HTML, CSS y Javascript.

### **Requisitos:**

- Funciones de callback
- Programación orientada a objetos
- Funciones anónimas

## **Semana 1(8) - Primeros pasos con React**

**Objetivo:** El estudiante conocerá el propósito de React para proyectos de gran escala y comprenderá el término virtual DOM, JSX y componente funcional.

### **Temas:**

- ¿Qué es React?
- Single Page Application
- Configurando el entorno de trabajo
- JSX
- DOM y Virtual DOM
- Componentes funcionales

**Entregables:** Una aplicación que muestre una vista similar a la página de inicio de youtube con tarjetas que contengan una imagen, título, canal, vistas y miniatura, con datos estáticos.

## **Semana 2(9) - Formularios, componentes funcionales y componentes de clase**

**Objetivo:** Comprender la diferencia entre un componente de clase y un componente funcional.

### **Temas:**

- Estado
- Propiedades (Props)
- Componentes de clase
- Eventos
- Tipos de inputs
- Atributos
- Validación de formularios con Formik

**Entregables:** Una aplicación que permita agregar y validar un usuario a través de los datos ingresados en un formulario.

### **Semana 3(10) - Renderizado condicional y Listas**

Objetivo: Comprender cómo mostrar y ocultar componentes usando estructuras if, switch o operadores ternarios.

Temas:

- Uso de if, else, switch en un componente de React
- Operador lógico &&
- Operadores ternarios
- Filtros
- Métodos map, filter, find, findIndex, every, some, splice, slice, push, pop para el manejo de arreglos

Entregables: Crear una aplicación que muestre una lista de empleados con su información y que filtre esos usuarios por edad, nombre y cargo.

### **Semana 4(11) - Ciclo de vida de React y consumo de API**

Objetivo: Comprender las diferentes fases del ciclo de vida de un componente de React y podrá consumir una API con el conocimiento del ciclo de vida de react.

Temas:

- ComponentDidMount
- ComponentDidUpdate
- ComponentWillUnmount
- Fetch y Axios
- Consumir una API con React

Entregables: Crear una aplicación que consuma la API pokeapi y muestre tarjetas con información de cada uno de los pokemones, agregar un filtro de búsqueda y un filtro de características del pokémon (ataque, defensa, velocidad).

### **Semana 5(12) - Hooks**

Objetivo: Comprender las diferencias entre componentes de clase contra un componente funcional que usa la característica de hooks.

Temas:

- Hooks de estado, efecto y contexto
- Hooks vs componentes de clase

Entregables: Crear una aplicación tipo memorama que muestre varios pares de tarjetas donde el usuario deberá sumar puntos cada vez que acierte en la pareja de tarjetas del mismo tipo, esta aplicación contará con un temporizador para terminar el juego pasados 2 minutos.

### **Semana 6(13) - React Router**

Objetivo: Construir una aplicación con una navegación entre vistas usando react router.

Temas:

- Rutas estáticas
- 404
- Redireccionamiento
- Rutas protegidas
- Parámetros

Entregables: Extender la aplicación hecha con pokeapi para implementar un inicio de sesión que te lleve a vista con todas las tarjetas de los pokemones y que cada vez que el usuario de click en la tarjeta del pokémon, te lleve al detalle del pokemon.

### **Semana 7(14) - Redux**

Objetivo: Comprender el propósito del uso de Redux y React en proyectos grandes, los conceptos de actions, reducers y store.

Temas:

- Propósito de Redux
- Actions
- Reducers
- Store
- Flujo de datos
- Middlewares (logger, thunk)
- Acciones asincronas

Entregables: Crear una aplicación con inicio de sesión y registro en el cual el usuario podrá ingresar una noticia, editarla, eliminarla y mostrar un panel con las noticias más destacadas o más vistas, además tendrá un filtro para mostrar las noticias por nombre por más recientes y por rango de fecha.

### **Desarrollo backend con Node y Express**

**Objetivo:**

El estudiante aprenderá a construir una aplicación desde el servidor con Node y Express desde 0 hasta el despliegue en un servidor. Tendrá la habilidad de realizar una API Rest usando una base de datos relacional, autenticación de usuarios, seguridad ante ataques

XSS y CSRF, podrá validar los datos que recibe el servidor y construirá middlewares personalizados.

**Requisitos:**

- Conocimiento intermedio del lenguaje de programación Javascript (programación funcional).
- Conocimientos básicos de Programación Orientada a Objetos.
- Conocimientos básicos de HTTP.

**Semana 1(15) - Introducción a Node con express:**

Objetivo: Comprender qué es node, desarrollar una aplicación para el manejo de archivos y peticiones básicas de tipo get para los usuarios.

**Temas:**

- ¿Qué es node?
- Configurando el ambiente de desarrollo
- Manejo de archivos con filesystem de node
- Servir archivos estáticos
- Diferencias entre el desarrollo con Node y Express
- Creando un servidor con Express
- Middlewares y rutas
- Métodos HTTP
- Configurando Babel

Entregable: Desarrollar una aplicación con node y express que maneje diferentes rutas típicas de una página web y que en una de las rutas lea un archivo y lo muestre en una respuesta de tipo HTML.

**Semana 2(16) - Base de datos relacionales**

Objetivo: Saber crear diagramas ER con sus relaciones.

**Temas:**

- ¿Qué son las bases de datos relacionales y que es Postgresql?
- Usar postgresql desde la terminal
- Diseñar diagramas de base de datos
- SQL (Crear base de datos, crear tablas, llaves primarias y llaves foráneas)
- Importar y exportar desde la terminal

**Semana 3 (17) - Postgresql**

Objetivo: Aprender a hacer normalización sobre una tabla y realizar consultas sobre la bases de datos y saber cuales son los diferentes joins y como usarlos.

**Temas:**

- Normalización
- Obtener, agregar, actualizar y eliminar registros
- Cláusulas, operadores, palabras clave SQL

- Joins

## **Semana 4 (18) - Node con ORM Sequelize**

Objetivo: Construir nuestra primera base de datos usando los conceptos de tabla, columnas, tipos de datos, relaciones, llaves primarias, llaves foráneas.

Temas:

- Conceptos básicos de base de datos relacionales
- Instalación de postgresql
- Sequelize
- Modelos
- Migraciones

Entregable: Crear una base de datos usando sequelize basándonos en un modelo entidad relación.

## **Semana 5(19) - API Rest con Node y Express**

Objetivo: Comprender el flujo para autenticar a un usuario usando Json Web Token y crear nuestra primera API Rest con autenticación.

Temas:

- Códigos de estado
- Patrones de diseño
- Autenticación
- OAuth 2.0
- Middlewares para autorización
- OWASP
- Seguridad ante XSS y CSRF
- Controladores
- Documentación con Swagger (OpenAPI 3.x.x)

Entregable: Crear una API Rest con un módulo de autenticación y autorización.



## **Semana 6(20) - API Rest con Node y Express**

Objetivo: Comprender cómo validar los datos que llegan a nuestro servidor y agregar características extra para cargar archivos a nuestro servidor, procesar pagos y enviar correos electrónicos.

Temas:

- Validación de datos
- Nodemailer (Envío de correos)
- Stripe (Procesar pagos)
- Socket io (Websockets - datos en tiempo real)
- Multer (Subir archivos)

Entregable: Continuar con el desarrollo de la API REST agregando nuevas características con Stripe para procesar pagos en línea.

## **Semana 7(21) - Pruebas unitarias con Jest y despliegue**

Objetivo: Comprender el proceso y la importancia de las pruebas unitarias en nuestra API REST.

Temas:

- Configurar jest
- Creando pruebas en nuestras rutas/endpoints
- Lanzando las pruebas
- Coverage
- Despliegue en Heroku (integración con github y a través de la consola)
- Despliegue en AWS Elastic Beanstalk

Entregable: Probar todas las rutas que construimos en nuestra API Rest y lanzar las pruebas para corregir bugs y crear un reporte de las pruebas realizadas.

## Desarrollo backend con Django

### Objetivo:

Aprender los conceptos básicos para crear un API Rest con Django. Al final del curso el estudiante debe ser capaz de implementar los principales componentes del framework para crear endpoints y construir su propio API Rest, valiéndose de Django y DRF. Entenderá los conceptos de modelos, serializadores, vistas, acciones, y el ORM de Django.

### Requisitos

- Conocimientos básicos de bases de datos y SQL
- Conocimientos básicos de tecnologías web
- Conocimientos básicos del protocolo HTTP y controladores para acciones base.

### Semana 1(22) - Introducción

Objetivo: Comprender los usos de python y desarrollar nociones del lenguaje.

#### Temas:

- ¿Por qué usar Python?
- Setup del ambiente
- Variables
- Tipos de datos
- Estructuras condicionales
- Funciones
- Ciclos
- Trabajar con archivos
- Manejo de errores

Entregable: Contador de palabras, crear un script que lea un archivo de texto y retorne las 10 palabras más repetidas. Debe usar listas, diccionarios, ciclos, funciones, try/except.

### Semana 2(23) - POO y tecnologías web

Objetivo: Entender los conceptos básicos de la programación orientada a objetos y lo principal de las tecnologías web: conceptos, patrones y arquitecturas.

#### Temas:

- Clases
- Instancias
- Métodos
- Herencia
- Interfaces
- Modelo OSI
- Arquitectura cliente servidor
- Patrones de diseño
- API Rest
- Ambientes virtuales

Entregable: Contador de palabras, actualizar el script anterior que lea un archivo de texto y retorne las 10 palabras más repetidas. Usando clases, métodos, instancias y lo previamente visto en clase.

### **Semana 3(24) - Django Vanilla**

Objetivo: Conocer los elementos principales de Django que nos permitan familiarizarnos con el framework y crear una aplicación base.

Temas:

- Instalación Django
- Creando nuestra primer aplicación con Django
- Modelos
- Bases de datos - ORM
- Views
- Urls
- Templates
- Rutas en templates

Entregable: Crear una aplicación para estudiantes que contenga un modelo Estudiante, una vista para listar los estudiantes en la base de datos y poder acceder al detalle de cada estudiante. Además crear un modelo para Clase, y una vista para poder ver la lista de clases disponibles y al dar click, poder listar los alumnos inscritos a cada clase.

### **Semana 4(25) - Introducción a Django Rest Framework**

Objetivo: Crear componentes genéricos para un API Rest base que nos permita entender la funcionalidad implícita del framework.

Temas:

- Instalación Django Rest Framework
- Setup
- Serializadores
- Vistas basadas en funciones
- APIView
- Vistas genéricas
- Autenticación

Entregable: Crear un crud para estudiantes y un crud para materias, es necesario que cada uno tenga implementados su modelo, serializador y vistas para los verbos HTTP.

### **Semana 5(26) - Django Rest Framework, buenas prácticas**

Objetivo: Crear componentes genéricos para un API Rest base que nos permita entender la funcionalidad implícita del framework.

Temas:

- ModelViewsets
- Acciones

- Filtros
- Paginación
- Permisos
- Correos

Entregable: Crear un proyecto de una librería que contenga las siguientes aplicaciones libros, autores y editoriales, cada aplicación es un endpoint y debe tener sus modelos, serializadores y vistas (ModelView set, acciones, filtros y paginación). Al crear un usuario debe mandar un correo de bienvenida.

### **Semana 6(27) - Django Rest Framework - Pruebas unitarias, Celery**

Objetivo: Aprender bases para desarrollar proyectos de calidad y comprender conceptos avanzados del desarrollo de software.

Temas:

- Test unitarios
- Celery
- Rabbit
- Swagger
- Proyectos finales

Entregable: Modificar el último proyecto creado, agregando tareas asíncronas, documentando el API y pruebas unitarias para cada vista.

### **Semana 7(28) - Entrega proyecto final**

Objetivo: Resolver dudas y concluir exitosamente el desarrollo del proyecto final del módulo.

Temas:

- Solución de dudas/problemas
- Entrega del proyecto final

## Algoritmos y estructuras de datos (7 semanas)

Objetivo: El estudiante comprenderá las estructuras de datos utilizadas con más frecuencia así como el propósito y ventajas/desventajas de cada una de estas estructuras.

También veremos cómo medir el rendimiento de nuestros programas, así como la manera de optimizarlos.

### Requisitos

- Conocimientos básicos de programación
- Conocimientos básicos de matemáticas
- Conocimientos básicos de Python

### Semana 1 (29) - Introducción y análisis de algoritmos

Objetivo: reforzar los conocimientos del lenguaje, introducir a la importancia y la forma de analizar la complejidad de un algoritmo.

#### Temas

- Tipos de datos
- Estructuras de control
- Funciones
- Manejo de excepciones
- Programación orientada a objetos
- Introducción a algoritmos
- Notación Big O
- Análisis de listas
- Análisis de diccionarios

### Semana 2 (30) - Estructuras de datos

Objetivo: Analizar y entender las estructuras de datos básicas que nos permitan resolver problemas algorítmicos sencillos.

#### Temas:

- Pilas
- Colas
- Lista ordenada
- Lista no ordenada

### Semana 3 (31) - Recursión

Objetivo: Entender la recursión y los problemas que resuelve.

#### Temas:

- Recursividad
- Leyes de la recursividad
- Marcos de pila: Implementación de la recursividad
- Visualización de la recursividad
- Problemas recursivos complejos
- Programación dinámica

### Semana 4 (32) - Ordenamiento y búsqueda

Objetivo: Analizar y entender los distintos tipos de algoritmos de ordenamiento y búsqueda así como la complejidad y usos adecuados de los mismos.

Temas:

- Búsqueda secuencial
- Búsqueda binaria
- Hashing
- Ordenamiento
- Ordenamiento burbuja
- Ordenamiento por selección
- Ordenamiento por inserción
- Ordenamiento de Shell
- Ordenamiento por mezcla

### **Semana 5 (33) - Ordenamiento y búsqueda**

Temas:

- Ejemplos de árboles
- Implementación
- Nodos y referencias
- Árbol de análisis
- Recorrido de árboles
- Árboles binarios
- Análisis de árboles de búsqueda
- Árboles binarios de búsqueda equilibrados

### **Semana 6 (34) - Grafos**

- Grafos
- Matriz de adyacencia
- Lista de adyacencia
- Búsqueda en anchura

### **Semana 7 (35) - Algoritmos de grafos**

- El problema de la gira del caballo
- Búsqueda en profundidad general
- Ordenamiento topológico
- Componentes fuertemente conectados
- Problema de la ruta más corta

### **Academlo Success: 4 Semanas para completar y pulir proyectos, bugs, etc. y prepárate para tu búsqueda de trabajo**

- asegurar una buena página web personal, con 2 - 3 proyectos que funcionan bien.
- coordinación con el student success manager
- apoyo de egresados
- talleres con el student success manager
- rastreo de aplicaciones de trabajo

- vinculación con empresas aliadas