

> > > > > >

CARRERA: DESARROLLO FULL STACK

Plan de estudios

CODER HOUSE

CODER HOUSE

CODERHOUSE es la nueva forma de entender el trabajo. Estudiamos y nos formamos porque estar en las últimas tendencias digitales es la manera de estar a la vanguardia en la sociedad. A través de la formación digital podemos ser independientes, conseguir el trabajo que tanto nos interesa o llevar a la cima nuestro propio emprendimiento.

Atravesamos una era donde los nuevos rockstars son quienes saben vivir la vida y disfrutarla. Por eso todos quieren ser Steve Jobs o Mark Zuckerberg. Esto quiere decir que ser un CODER es como ser un estrella.

#NERDS4EVER



DESARROLLO FULL STACK

En esta carrera te desarrollarás desde cero como Full Stack Developer. Aprenderás a crear un sitio web sumergiéndote en las mejores prácticas del diseño web responsive, trabajando con HTML y CSS. Asimismo, sabrás cómo programar con Javascript, agregando interactividad y dinamismo a tus proyectos. Además, conocerás el framework React JS, que te permitirá realizar sitios más complejos, dinámicos y escalables. Por otro lado, en el curso de Programación Back End aprenderás a desarrollar aplicaciones modernas con Node.js y MongoDB. Programarás en JavaScript del lado del servidor aplicando técnicas asincrónicas, y aprenderás a trabajar con Bases de datos SQL y NoSQL, dominando la gestión de los datos en forma eficiente, ágil y con una gran facilidad de escalabilidad.

Serás capaz de diseñar complejas aplicaciones back end robustas, rápidas y escalables, pudiendo dominar diferentes técnicas de comunicación, manejo de procesos distribuidos, control de grandes volúmenes de datos y despliegue a distintas plataformas web. Al finalizar la carrera, contarás con los conocimientos para desempeñarte como Full Stack Developer.

MODALIDAD

Clases Online en Vivo

DURACIÓN

12 meses / 48 semanas

CODER HOUSE

Plan de estudios



Experto que valida el curso:

MARIO COLQUE

Full Stack Engineer en PedidosYa



Experto que valida el curso:

EDUARDO SANCHEZ

Full Stack Engineer en PedidosYa



Experto que valida el curso:

YECID FERNANDO GOMEZ CUACES

Software Development Sr. Analyst
en Mercado Libre



CURSOS DE LA CARRERA

1. Desarrollo Web

En este curso aprenderás a crear tu sitio web partiendo del prototipo en papel. Te sumergirás en las mejores prácticas del desarrollo web, trabajando con HTML y CSS. Conocerás herramientas para optimizar al máximo tu sitio web, implementando prácticas de versionado de código con GIT, y preprocesadores como SASS. Al finalizar, sabrás cómo aplicar Bootstrap a tus proyectos, y comprenderás lo importante del SEO en tus desarrollos. Subirás tu sitio a un servidor, y aprenderás a interactuar con este servicio. También sabrás cómo presentar un presupuesto y atender a tu cliente final.

2. JavaScript

En este curso aprenderás los fundamentos del lenguaje de programación más usado en la actualidad, con el cual es posible crear aplicaciones de todo tipo. Explorarás inicialmente herramientas propias del mismo, indagando casos prácticos de aplicación. Reconocerás la utilidad de las librerías estudiando jQuery, y cómo aplicar técnicas de desarrollo para apps modernas con AJAX. Al graduarte, estarás en condiciones de crear soluciones web interactivas, y trasladar los conocimientos del curso a cualquier framework JavaScript.

CURSOS DE LA CARRERA

3. React JS

En este curso, que es el tercer nivel de la carrera de programación, aprenderás a programar por componentes, mediante Javascript, JS, ES6, y también conocerás las ventajas de la utilización del flujos de datos.

Asimismo, aplicarás el manejo de rutas utilizando Firebase, y comprenderás la utilización del virtual DOM mediante los desarrollos de React JS.

Al finalizar, podrás crear tus propias aplicaciones SPA, y estarás en condiciones de hacer cualquier desarrollo con uno de los frameworks más populares y avanzados del momento.

4. Programación Backend

En este curso aprenderás a desarrollar aplicaciones modernas con Node.js y MongoDB. Programarás en Javascript del lado del servidor aplicando técnicas asincrónicas. Aprenderás a trabajar con Bases de datos SQL y NoSQL dominando la gestión de los datos en forma eficiente, ágil y con una gran facilidad de escalabilidad. Al finalizar el curso serás capaz de diseñar complejas aplicaciones backend robustas, rápidas y escalables, dominando diferentes técnicas de comunicación, manejo de procesos distribuidos, control de grandes volúmenes de datos y despliegue a distintas plataformas web.

TEMARIO

CODER HOUSE

DESARROLLO WEB

0. Nivelación (optativa)

- Conceptos básicos del Desarrollo Web.
- Herramientas a utilizar.
- Instalación del software.

1. Prototipado y conceptos básicos de HTML

- Proyecto web: ¿qué es?
- Prototipado desde el papel.
- HTML y etiquetas.

2. Primeros pasos con HTML

- Listas, tablas, formularios y enlaces.

3. Incluyendo CSS a nuestro proyecto

- Multimedia con HTML: imágenes, videos, audio, iframe.
- Bases del CSS y atributo Class.

4. CSS - Medidas, colores y fuentes

- Estilos para textos y listas.
- Tipografías web y fondos.

5. Box Modeling

- Propiedades de la caja.
- Posicionamiento y visualización.

6. Flexbox

- ¿Qué es?
- Propiedades de padres e hijos.
- Propiedades para el contenedor Flex, y los Flex items.

7. Grids

- Implementación de Grids y Grids Mobile First.

CODER HOUSE

Plan de estudios

TEMARIO

CODER HOUSE

DESARROLLO WEB

8. Animaciones, transformaciones y transiciones

- Animaciones con CSS.
- Incorporación de transformaciones y transiciones a elementos.

9. Frameworks CSS + Bootstrap

- ¿Qué es un framework?
- Funcionalidades de Bootstrap.

10. Bootstrap + Themes

- Bootstrap con JS.
- Bootstrap Themes.
- Diseño responsive.

11. SASS I

- Preprocesadores CSS.
- Nesting.
- Import y Vars.
- BEM.

12. SASS II

- Operadores.
- Condicionales.
- Bucles.
- Maps.

13. Servidores, seguridad y SEO para tu sitio

- Servidores, seguridad y SEO: conceptos básicos.

14. Git

- Comandos básicos de la terminal.
- Creación de repositorios y ramas.

15. GitHub

- Implementación de Grids y Grids Mobile First.

CODER HOUSE

Plan de estudios

TEMARIO

CODER HOUSE

DESARROLLO WEB

16. Subida al servidor

- Configuración del hosting y acceso al servidor.

17. Conociendo a nuestro cliente

- Identificación del cliente y el usuario final.
- Confección del presupuesto.

18. Proyecto final

- Repaso de los temas principales
- Revisión de los objetivos alcanzados en el curso.
- Recomendaciones finales.
- Presentación de los avances del proyecto.

CODER HOUSE

Plan de estudios

TEMARIO

CODER HOUSE

JAVASCRIPT

1. Conceptos generales: sintaxis y variables

- Presentación Javascript: ¿para qué se usa?
- Variable: ¿qué es y cómo declararla? Asignación y cambio del valor.

2. Control de flujos

- Condicional: ¿qué es?
- Operadores de comparación y lógicos: ¿qué son y cuál es su uso en los condicionales?

3. Ciclos/iteraciones

- Ciclos: ¿qué son? Tipos y diferencias entre sí.
- Operadores lógicos, ciclos y funciones: ¿cómo combinarlos?

4. Programación avanzada con funciones

- Función: ¿qué es? ¿Qué son los parámetros de entrada y de salida?
- Scope global y local.
- Función anónima vs. flecha

5. Objetos

- Objetos: ¿qué son y cómo se usan? ¿Cuáles son sus propiedades y métodos?
- Función constructora.

6. Arrays

- Funciones nativas de JS para operar arrays.
- Trabajar con array de objetos.
- Algoritmos.

CODER HOUSE

Plan de estudios

TEMARIO

CODER HOUSE

JAVASCRIPT

7. Storage y JSON

- localStorage y sessionStorage.
- JSON: alcance del formato y situaciones de uso.

8. DOM

- Definición, alcance y su importancia para operar sobre elementos HTML.

9. Eventos

- ¿Qué son, para qué sirven y cuáles son los más comunes?
- Escuchar un evento sobre el DOM.

10. Workshop I

- Repaso de temas vistos.
- Recomendaciones para el proyecto final.
- Avance individual sobre el proyecto.

11. jQuery y selectores

- Librería: ¿qué es?
- jQuery: ¿para qué sirve?

CODER HOUSE

- Uso de selectores.
- Agregar elementos al DOM.

12. jQuery: selectores y eventos

- Uso de selectores.
- Manejo de eventos en jQuery.

13. Efectos y animaciones con jQuery

- Animaciones de jQuery, y combinaciones posibles.
- Modificación del CSS desde jQuery.

14. AJAX con jQuery

- AJAX: ¿qué es y cómo se usa?
- Llamadas simples mediante AJAX.
- API: concepto.

15. Introducción a SPA con jQuery

- SPA: ¿qué es y cuáles beneficios tiene sobre las páginas?
- MVC: concepto.

16. Workshop II

- Revisión de consultas acerca del curso.
- Presentación de proyectos.

Plan de estudios

TEMARIO

CODER HOUSE

REACT JS

1. Nivelación

- Repaso: HTML, CSS y fundamentos JavaScript.
- Sintaxis de JavaScript y ECMAScript 6.
- Variable 'this'.
- Introducción a React JS.

2. Instalación y configuración del entorno

- Funcionamiento del virtual DOM en React JS.
- Instalación y configuración de Node.js.
- Creación de una app utilizando el CLI.

3. JSX y transpiling

- JSX.
- Expansión de la sintaxis avanzada de JS.
- Rol de webpack y babel en el bundling/retrocompatibilidad.

4. Componentes I

- ¿Qué son?
- ¿Cuáles problemas resuelven?
- ¿Cómo implementarlos?

5. Componentes II

- Estados y ciclos de vida.
- Propiedades y eventos.

6. Promises, asincronía y MAP

- API Promise.
- Aplicación del método MAP para el rendering de listas.

7. Consumiendo APIs

- Paradigma de intercambio de datos.
- Consumo de recursos vía llamados a APIs.

8. Routing y navegación

- Organización de la app.
- Configuración de la navegabilidad entre componentes.

CODER HOUSE

Plan de estudios

TEMARIO

CODER HOUSE

REACT JS

9. Eventos

- Sistema de eventos de React JS.
- Diseño de componentes orientados a eventos.

10. Context

- Organización de los eventos aplicativos en cada proyecto.
- Creación de interacciones persistentes entre componentes.
- Desarrollo de implementaciones custom de context.

11. Técnicas de rendering

- Profundización acerca de renderizado condicional y sus implicancias.
- Diagnóstico y solución de problemas de rendering.

12. Firebase I

- Modelo de serverless-computing vs. tradicional.
- Conexión de la app a servicios cloud-first de alta disponibilidad.
- Configuración de un storage, sus colecciones y acceso a la información.

13. Firebase II

- Inserción y actualización de información usando Firestore.
- Flujos de control.

14. Workshop

- Parte 1: optimización de código.
- Parte 2: checkout final, limpieza de código y configuración de readme.md.

CODER HOUSE

Plan de estudios

TEMARIO

CODER HOUSE

PROGRAMACIÓN BACKEND

Módulo 1: Servidores con Javascript desde cero

- Javascript, ECMAScript y Typescript en Node.js: Conceptos básicos para desarrollar Javascript en Node.js. ES6 y nuevas sintaxis. Aportes de ES7 a ES11. Introducción y uso de Typescript.

- Programación Sincrónica vs. Asincrónica: Funciones y callbacks. Sincronismo vs asincronismo. Conceptos de Iteradores, Promesas y Observables. Ejemplos prácticos.

- Nuestro primer servidor web: Qué es Node.js, Patrón Módulo. Escribir nuestro primer programa en Node.js. Módulos nativos. Módulo HTTP. Cómo implementar nuestro primer servidor. Despliegue en glitch.com.

- Manejo de archivos: En forma sincrónica. Con callbacks. En forma asincrónica. Con promesas (then-catch y async await).

- NPM e introducción a Express: ¿Qué es npm? ¿Cómo instalar dependencias? Qué es Express y para qué nos sirve. Cómo instalar Express en nuestro proyecto. Crear nuestro primer servidor con Express. Despliegue en glitch.com

- Express avanzado: Servir archivos estáticos con Express. Crear rutas GET, POST, PUT y DELETE. Uso de POSTMAN. Creación de capas middleware. Formulario de login. Subida de archivos al servidor con Multer.

CODER HOUSE

Plan de estudios

TEMARIO

CODER HOUSE

PROGRAMACIÓN BACKEND

- Motores de plantillas: Handlebars, Pug y Ejs. Instalación e integración a un proyecto Express. Layouts, vistas y componentes parciales. Servir contenido HTML utilizando Handlebars, Pug y Ejs.

- Websockets: ¿Qué es Socket.io? Diferencias entre HTTP estándar y Websockets. Integración Socket.io a un proyecto Node.js. Creación de aplicaciones de consumo de datos constantes. Aplicación websocket Chat y Paint. Despliegue en glitch.com.

- Node.js como herramienta de desarrollo: ¿Qué es un transpiler? Instalación y uso de Babel. Instalación y uso de TSC (TypeScript Compiler). Qué es un bundler y para qué nos sirve. Instalación y uso de

Webpack. Configuración de Webpack, Babel y TSC para compilación de código. Compilación de código en Node.js para producción.

Módulo 2: Bases de datos - Tecnologías e integración

- SQL y Node.js: Cómo servir una base de datos SQL MySQL. Clientes MySQL (CLI, UI). Conectar base de datos MySQL a través de Knex. Integrar base de datos MySQL a un proyecto Node.js. Knex con SQLite3. CRUD.

- Introducción a MongoDB: Qué es MongoDB. Tipos de bases de datos NoSQL vs SQL. Arquitectura, instalación y configuración de MongoDB. Acceso a una base de datos MongoDB. Clientes de MongoDB (CLI, UI)

CODER HOUSE

Plan de estudios

TEMARIO

CODER HOUSE

PROGRAMACIÓN BACKEND

- CRUD en MongoDB Client: Qué es CRUD. Operaciones básicas en MongoDB. Creación de bases de datos, colecciones y documentos en colecciones. Edición, reemplazo y borrado de documentos. Creación y configuración de usuarios.
- MongoDB y Node.js: Integrar y conectar una base de datos MongoDB con Node.js. Estructura de una base de datos no relacional con Mongoose. CRUD. MongoAtlas
- Integración con Firebase: Conectar una base de datos Firebase con Node.js. CRUD.

Módulo 3: Características avanzadas de desarrollo Backend

- Diseño de mocks de datos: Diseño para Back/Front. Desarrollo mockapi
- JSON Normalización: Normalizando orígenes de datos. Concepto y ventajas. Denormalización.
- Cookies, sesiones, storages: Qué son, cómo usarlas. Asegurando aplicaciones. Storage de sessions: memory, file, Redis, RedisLab.
- Autorización y Autenticación: Mecanismos de Auth. Integración de passport a un servidor basado en Express. Estrategias de login con redes sociales. JSON Web Token (JWT). Desarrollo registro y login de usuario utilizando MongoDB

CODER HOUSE

Plan de estudios

TEMARIO

CODER HOUSE

PROGRAMACIÓN BACKEND

- Global Process y Child Process: Propiedades, métodos y eventos. Spawn, fork, exec y execFile.
- Módulo Cluster: Definición y uso. Forever y PM2.
- Nginx y escalabilidad: Nginx como reverse proxy - configuración. Balanceo de carga. Esquemas de red. Contenido estático. Integración con Node.js
- Logs, profiling & debug: Logs - principios, módulos, rendimiento. Log4s, Winston y Pino. Profiling - qué es, herramientas y análisis. Rendimiento - artillery, autocannon y Ox.

- Productos Cloud y soluciones de terceros: Uso de Git para deploy. Deploy en Heroku. Deploy en AWS: EC2, S3, RouteS3, Balance.
- E-mailing y Messaging: Nodemailer, enviando mails desde Node.js. Twilio, envío de SMS y Whatsapp.
- Seguridad: OWASP paso a paso. Top 10.

Módulo 4: Arquitectura y configuración del servidor

- Creación de proyectos: Administradores de paquetes. Generalidades y particularidades. Configuración de proyectos en capas.

CODER HOUSE

Plan de estudios

TEMARIO

CODER HOUSE

PROGRAMACIÓN BACKEND

- Servicios Web: ¿Qué es una API? Aplicaciones RESTful. Aplicaciones GraphQL. Manejo y documentación de APIs.
- Cliente HTTP de pruebas: Uso de dependencia request. Uso para testeo de endpoints de servidor.
- Arquitectura del Servidor - Diseño: Arquitectura: separación en capas MVC (ruteo, lógica de negocio, persistencia). Patrones de diseño: Singleton, Factory.
- Arquitectura del Servidor - Persistencia: Patrones de diseño: DAO/DTO, Repository. Bases de datos: archivos de configuración, drivers, SQL vs NoSQL.
- Configuración del Servidor Web: Uso de dotenv.
- Desarrollo de un Servidor Web basado en capas: Diseño API REST y GraphQL MVC stack MERN.
- Introducción a frameworks de desarrollo Backend: Adonis, Nest, Sails, Koa. Ejemplos de uso.
- El futuro de Node.js - Deno: Generalidades. Ejemplo básico de uso.

CODER HOUSE

Plan de estudios



CARRERA: DESARROLLO FULL STACK CODER HOUSE

hola@coderhouse.com
www.coderhouse.com
[instagram.com/coderhouse](https://www.instagram.com/coderhouse)

