

> > > > > >

# ***PROGAMACIÓN BACKEND***

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***

# 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.



Experto que valida el curso:

**YECID FERNANDO GOMEZ CUACES**

Software Development Sr. Analyst  
en Mercado Libre

## MODALIDAD

Online en vivo

## DURACIÓN DEL CURSO

6 meses - 24 semanas

**CODER HOUSE**

Plan de estudios



Experto que valida el curso:

**PABLO MIÑONES**

Technical Lead en Mercado Libre



Experto que valida el curso:

**YECID FERNANDO GOMEZ CUACES**

Software Development Sr. Analyst en Mercado Libre



Experto que valida el curso:

**MARIO COLQUE**

Full Stack Engineer en PedidosYa



Experto que valida el curso:

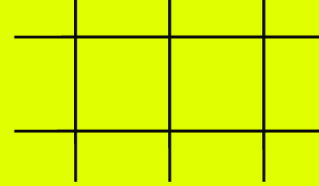
**EDUARDO SANCHEZ**

Full Stack Engineer en PedidosYa



## ***HERRAMIENTAS PRINCIPALES DEL CURSO***

(Clic en cada una para más información)



## ***PERFIL PROFESIONAL***

Al finalizar el curso de Programación Backend podrás:

- Comprender el ecosistema Node.js/JavaScript.
- Dominar los lenguajes JavaScript JS5, ES6, ES7, ES8, ES9, ES10, ES11 y Typescript
- Programar en Javascript asincrónico avanzado del lado del servidor
- Utilizar el middleware Express.js
- Usar el framework Handlebars, Pug y Ejs para manejar plantillas HTML
- Crear APIs con RESTFul y GraphQL.
- Testear y monitorear aplicaciones REST.
- Dominar la tecnología Websockets
- Instalar MySQL y Redis. Realizar un CRUD desde Node.js.
- Hacer la instalación de MongoDB.
- Comprender los principios y diferencias con SQL..
- Trabajar con la consola de MongoDB y clientes UI/Web.
- Manejar MongoDB desde Node.js a través de Mongoose.
- Realizar CRUD con MongoDB.
- Llevar a cabo importaciones, exportaciones y backups de una base de datos MongoDB
- Normalizar/denormalizar datos en formato JSON.
- Manejar patrones de diseño y code style para aplicaciones de alta concurrencia.
- Diseñar servidores web desarrollando en capas con MVC.
- Implementar proyectos en la nube a través de plataformas como AWS, Heroku, Glitch, MongoAtlas, RedisLab

# ¿POR QUÉ CODER?



## **48 CLASES**

Un total de 48 clases enfocadas en dominar el desarrollo de un proyecto backend desde cero.



## **CLASES COMPLEMENTARIAS**

Clases adicionales a la cursada, optativas y con la finalidad de asentar conocimientos o resolver dudas



## **TUTOR A CARGO**

Además del profesor del curso, contarás con un tutor que te guiará en el proceso de aprendizaje



## **CLASES ONLINE EN VIVO**

Clases online y en vivo. Además contarás con la grabación de la clase para ver en cualquier momento



## **APRENDER HACIENDO**

Aprender con la práctica, basada en 3 pasos: docente explica, luego hace, finalmente el estudiante lo hace



## **ACTIVIDADES COLABORATIVAS**

En ciertas clases en vivo, habrá momentos de interacción entre los estudiantes en grupos reducidos.

# TEMARIO

**CODER HOUSE**

## 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**

- 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**

- 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**

- 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**

- 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

# PROYECTO FINAL

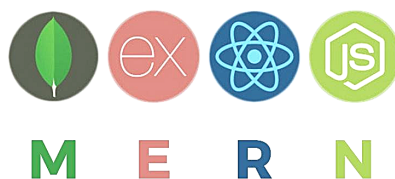
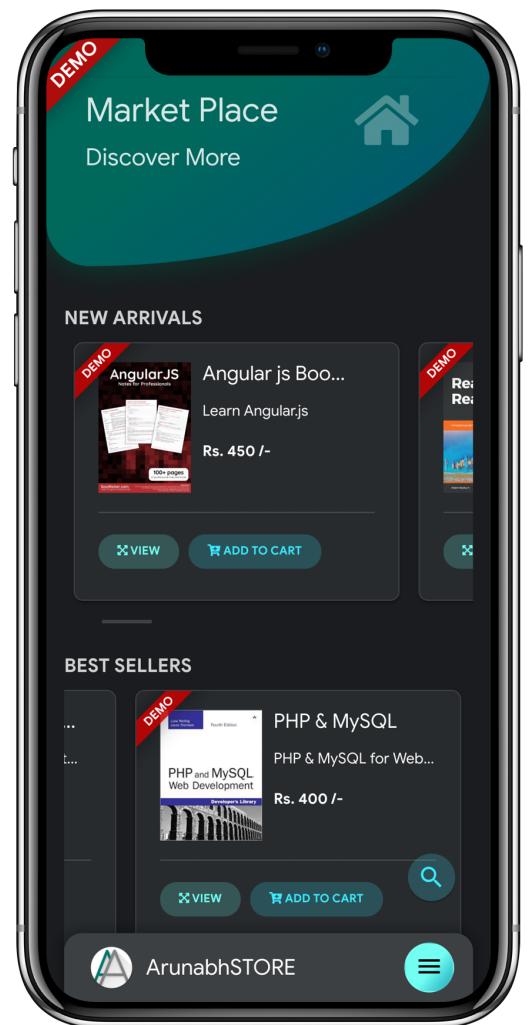
## E- Commerce Backend

Desarrollarás el backend de una aplicación de e-commerce, para poder vender productos de un rubro a elección.

La aplicación permitirá listar los productos existentes, ingresar nuevos, borrar y modificar sus detalles, así como interactuar con el carrito de compras. Los productos ingresados se almacenarán en una base de datos MongoDB. El usuario podrá registrar sus credenciales de acceso (email y password), para luego poder ingresar a su cuenta. El cliente tendrá una sesión activa de usuario, con tiempo de expiración configurable.

Asimismo, implementarás un canal de chat basado en websockets, el cual permita atender las consultas del cliente.

La arquitectura del servidor estará basada en capas (MVC). Utilizarás JWT (Jason Web Token), para brindar al frontend un mecanismo de ingreso autorizado al sistema



**CODER HOUSE**

Plan de estudios



# ***PROGRAMACIÓN BACKEND CODER HOUSE***

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

