



CURSO NODE.JS BACKEND

Plan de estudio



educación 





Nuestro propósito

Transformar positivamente la vida de las personas.

Queremos que seas protagonista en la transformación que estamos viviendo. Por eso, nos comprometemos a capacitarte para que estés al día con las necesidades digitales actuales.

Te invitamos a trabajar en conjunto para que descubras tu mejor versión y la potencies. Anímate, toma las riendas de tu futuro.

Code your future!

Contenido del curso

Ingresa al mundo de Node.JS, el lenguaje que ofrece la mayor velocidad de procesamiento a tus aplicaciones.

Proyecto Integrador

App para un supermercado

Desarrollarás el backend de un e-commerce. En la primera etapa, realizarás la versión estática del sitio. La arquitectura del sistema estará basada en capas utilizando Node.js express y base de datos MongoDB Atlas.

La aplicación permitirá las siguientes funciones:

- Agregar, actualizar y borrar productos desde un sistema de stock.
- Consultar por producto, por stock o por precio.
- Compra de productos con pedidos online o cajas locales.
- Chat entre clientes conectados y el supermercado.
- Login y sesiones de usuarios con distintos roles (cliente, caja, administrador).
- Panel de visualización del estado de productos en tiempo real.
- Envío de mail informando distintas situaciones: por ejemplo, si un producto tiene stock crítico.

¿Qué aprenderás?

- Ecosistema Node.js/JavaScript.
- ECMAScript moderno del lado del servidor.
- Servidor API RESTful JSON con npm y express.
- Implementación de motores de plantillas.
- Conexión MongoDB y librería Mongoose.
- WebSockets: Conexión cliente y servidor.
- Sesiones storage, autenticación y autorización.
- Patrones, capas y optimización.
- Webpack y Babel para compilación de código.
- Testing de servidores, metodologías TDD y BDD.
- Frameworks: Adonis, Nest, Sails y KOA.
- Introducción a Deno: el futuro de Node.js.

Plan de estudios

1

Plataforma Node.js

- ¿Qué es Node.js? Estructura, funcionamiento, comparación con otras plataformas, versiones, ciclo de vida
- Consolas, uso básico y avanzado
- Instalación de Node.js
- ¿Qué es npm? Modos de uso, instalación de dependencias
- Uso de GIT básico, GitHub
- Uso de Node.js en consola
- Inicialización de un proyecto en Node.js, análisis del package.json, scripts
- Uso de Nodemon
- Uso de Node.js en consola (REPL)
- Global Objects: __dirname, __filename, exports, require (CommonJS)
- Module import con ES Modules
- Global process
- Módulo Assert
- Módulo Event: Event Emitter
- Módulo Útil
- Módulo Net

2

Javascript moderno en servidores

- JS05: callbacks, foreach, for in, for of, try catch, uso de throw, excepciones

- ES06: let, const, template string, arrow function, class, map, filter, spread, rest, Promises
- ES07: includes, operador exponencial (**)
- ES08: Object values y entries, String Padding, trailing commas, async/await
- ES09: Extensión de funcionalidades rest/spread, Promise finally
- ES10: BigInt, String trimstart/end, Array flatMap, dynamic import, ObjectfromEntries, Optional catch Binding, globalThis, Private Class variables #
- ES11: Promise.allSettled, Nullish Coalescing Operator, Optional Chaining.
- Typescript
- Librería RxJS

3

Servidores HTTP en Javascript y express

- Desarrollo de un servidor con módulo nativo HTTP, parseo de rutas y métodos.
- Publicación local y prueba con navegador
- Postman: instalación y uso para test de servidores
- Manejo del file system (fs) en Node.js: modo sincrónico y asincrónico, uso de promesas y async await
- Integrar file system en nuestro servidor HTTP: servir documento html estático
- ¿Qué es Express?
- Creación de un proyecto de servidor basado en Express
- Definiciones de rutas, listen del servidor, manejo de headers, errores, eventos

- Servicio de archivos estáticos en express. Ejemplo con un formulario post
- Métodos use, set y get en Express. Creación de middlewares
- Uso de plataforma Glitch para despliegue de un proyecto Node.js HTTP a la nube

4

Uso avanzado de Middleware express

- Generación de HTML dinámico con template strings, desventajas
- Motores de plantillas, concepto y uso en un servidor basado en express
- Handlebars, pug y ejs: instalación de dependencias, configuración
- Realización de un proyecto para cada una de las plantillas, sirviendo contenido dinámico
- Uso de la plataforma Heroku para despliegue de un proyecto Node.js HTTP a la nube
- Desarrollo de un Api Restful básica (get, post, put, delete) y prueba con postman
- Dependencias para validar datos entrantes: joi
- Subir archivos al servidor con multer: instalación y uso
- Enviar correo desde Node.js nodemailer: instalación, configuración y uso

5

Bases de datos modernas en servidores Javascript

- Persistencia con file system y base de datos. Ventajas y desventajas

- ¿Qué es MongoDB?
- Instalación local del servidor de base de datos
- Uso de cliente CLI: mongo Shell
- CRUD en consola: lenguaje de query, find, insert, update, delete
- Instalación y uso de cliente GUI: Mongo Compass, Robo 03-
- Mongo Atlas, creación de una cuenta, conexión e interacción con clientes CLI y GUI
- Creación de un proyecto node.js cliente de MongoDB
- Instalación y uso de mongoose para operaciones CRUD: modelos y esquemas
- Operación de mongoose con callbacks y promises (then catch y async await)
- Conexión del servidor node.js con mongo Atlas

6

Sesiones, storages y mecanismos de Auth

- Cookies, sesiones en express
- Storages de sesiones (memory, file, redis, Mongo)
- Crear formulario de login autenticado, uso de Passport
- Login con redes sociales
- Uso de passport para Autenticación con redes sociales
- Mecanismos de Auth empleando JSON Web Token (JWT)
- Proyecto ejemplo fullstack con Auth JWT
- CORS en servidores

7

Websockets

- ¿Qué es un WebSocket? Comparación con HTTP

- Configuración de la librería websocket Socket.io en cliente y servidor
- Creación de un proyecto node.js con express basado en la librería websocket Socket.io
- Librerías WS y SocketJS en Node.JS
- Creación de una aplicación de chat con websockets, publicación local y en la nube
- Creación de una aplicación de paint colaborativo con websockets, publicación local y en la nube

8

Estructura del servidor y herramientas de desarrollo

- Arquitectura: Patrón MVC, separación en capas (ruteo, lógica de negocio, casos de uso persistencia)
- Arquitectura del Servidor: Persistencia, patrones: DAO/DTO, Repository
- Configuración del servidor: uso de dotenv
- Qué es Babel, instalación y uso bajo node.js
- Qué es Webpack, instalación y uso bajo node.js
- Cómo utilizar Typescript en nuestros proyectos frontend y backend bajo node.js
- Creación de un proyecto de servidor con Webpack, Babel y Typescript integrado

9

Testing

- Metodología TDD y BDD
- Concepto de pruebas unitarias y de integración
- Utilización de jest y mocha como dependencias de testing
- Creación de suites de test

- Testear una Api Rest Http con mocha
- Creación de un Http Client para test
- Concepto de Mock y su uso en test
- Uso de Faker.js como dependencia generadora de mocks
- Ejercicios y prácticas sobre plataforma Node.js

10

Frameworks y entornos de desarrollo Backend

- Adonis
- Nest
- Sails
- Koa
- Ejemplos de uso
- El futuro: Deno
- Generalidades
- Seguridad, dependencias
- Ejemplo de uso creando un servidor en Deno

Modalidad del Curso

Duración

10 semanas / 40 h

Frecuencia semanal

2 encuentros de 2 h

Modalidad

Online en vivo

Grupos reducidos

Promedio 25 personas

Nivel: Intermedio



☐ Principiante

☒ Intermedio

☐ Avanzado

☐ Experto

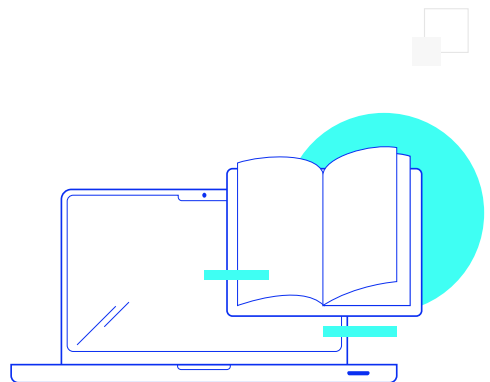
Requisitos

Es recomendable tener una base sobre:

- [Javascript Desarrollador Avanzado](#)
- [Mongo DB Fundamentos](#)

Dedicación fuera de clase

Además de las horas de clase, recomendamos que inviertas 4 h semanales extras para realizar los desafíos complementarios, estudiar el material de lectura y completar los exámenes del Alumni para potenciar tu estudio.



¿Cómo será tu experiencia?



Aprender haciendo

Ejercita y pon en práctica lo estudiado.



Trabajo en equipo

Une esfuerzos y potencia los resultados.



Clases grabadas

Consúltalas las veces que quieras.



Profesores expertos

Aprende de gigantes de la industria.



Asistente académico

Recibe soporte dentro y fuera de clase.



Plataforma Alumni

Encuentra recursos, materiales y clases.

¿Por qué Educación IT?



IT Créditos

Gana puntos al aprobar los exámenes de los cursos. Luego, podrás canjearlos por nuevos cursos sin costo alguno. Los IT Créditos que acumules no vencen ni se devalúan.



Garantía de aprendizaje

Si necesitas reforzar conceptos, recuperar clases o no estás satisfecho, ¡vuelve a tomar el curso sin ningún costo! Puede ser de forma total o parcial.



Comunidad en Discord

Mantente en contacto con la comunidad de EducaciónIT a través de nuestro servidor de Discord. Podrás hablar con tus compañeros, profesores, asistentes académicos y soporte.

Preguntas frecuentes

Si me pierdo una o más clases, ¿puedo recuperarlas?

Todas las clases quedan grabadas de por vida en tu plataforma Alumni. ¡Siempre podrás volver a verlas cada vez que lo necesites!

¿Cómo voy a aprender?

Te enfrentarás a situaciones de trabajo reales, en donde tendrás que aplicar lo aprendido de forma individual y en equipo. Por medio de la prueba y el error, irás superando desafíos y obteniendo nuevas habilidades que luego podrás aplicar en el ámbito laboral.

¿Cómo son las clases online en vivo?

Las clases duran entre 2 y 3 horas de lunes a viernes (sábados 3 o 4 hs) y se desarrollan de forma online en vivo en aulas virtuales, donde vas a poder interactuar con el instructor y tus compañeros.

Manejamos cupos reducidos para que puedas tener un seguimiento más personalizado durante tu aprendizaje.



www.educacionit.com



@educacionit
