

CURSO DE NODE.JS

Septiembre 2018

25 horas

ACCESO AL CURSO

- Fuentes del curso en GitHub
- Visualización vía diapositivas:

```
git clone https://github.com/juanda99/curso-node-js.git .  
npm install  
npm start
```

PONENTE

- Juan Daniel Burró Aláez (juandacorreo@gmail.com)
 - Profesor informática en CPIFP Los Enlaces
 - Actualmente en atribución funciones:
 - Desarrollo [nuevo portal Arasaac](#)
 - SPA con React.js
 - API REST con node.js y MongoDB

TEMARIO DEL CURSO

COMO EMPEZAR CON NODE.JS

- Instalación de Node
- Gestión de versiones de node
- Uso de linters
- Debug en Node
- Vistazo general a las características de ES6 y ES7.
 - Qué puedo usar con Node

REALIZAR PROYECTOS/PAQUETES CON NODE

- Uso del gestor de paquetes
- Instalación y configuración de paquetes del repositorio de npm
- Gestión de dependencias
- Creación de una librería
- Publicación en npm

CREAR SCRIPTS DE SERVIDOR MEDIANTE NODE

- Aprender a manejar la naturaleza asíncrona de node: callbacks, promesas, async-await
- Uso de paquetes relacionados con el sistemas de ficheros
- Configuración de microservicios

NODE.JS PARA WEB:

- Crear un servidor API REST mediante express
- Operaciones CRUD contra base de datos
- Operaciones CRUD autenticadas mediante tokens según OAUTH2
- Uso de Postman para comprobar API
- Uso de WebSockets

ENFOQUE DEL CURSO

CONFIGURACIÓN INICIAL

- Trabajamos con una máquina virtual prácticamente preparada:
 - Visual Code (sin configurar)
 - Docker
 - zsh
 - ...

INTRODUCCIÓN A NODE.JS

- Características principales de Node.js

PROYECTOS

- Proyectos pequeños donde veremos los contenidos:
 - npm
 - Promises y async-await
 - Servidor web con node, arquitectura API REST y operaciones CRUD
 - Autenticación con json web tokens
 - WebSockets

- Utilizaremos en varios proyectos las siguientes tecnologías:
 - Gestión versiones mediante git
 - Test con enfoque BDD (Behaviour Driven Development) y Code Coverage
 - Integración continua
 - Dockerización de servidores adicionales (MongoDB)
 - Despliegue serverless

CONTENIDO DE LOS PROYECTOS

PROYECTO 1: APLICACIÓN ¡HOLA USUARIO!

- *Escribe un mensaje en un fichero:*

```
Hola usuario, tienes 17 años
```

- *Usuario es el username del equipo*
 - *Edad es una variable*
- **Uso de módulos propios y built-in modules**

PROYECTO 2: DOCUMENTACIÓN

APUNTES

- *Generación de slides y pdf a partir de markdown, útil para hacer apuntes*
- **Uso de módulos de terceros**
- Configuración de proyecto y gestión de paquetes mediante **npm**

PROYECTO 3: CREAR UNA LIBRERÍA

- *Crear una librería a partir de un json de datos*
- Configuración de **linters**
- Uso de **tests**
- **Publicación en el repositorio de npm**

PROYECTO 4: LISTA DE TAREAS

- *Clásico ejercicio de todo list con persistencia en fichero*
- Uso de json

PROYECTO 5: SERVIDOR WEB

- *Servidor web mediante paquetes de node*
- Uso de Express como servidor web
 - Arquitectura y uso de middlewares
 - Uso de templates
- Despliegue serverless

PROYECTO 6: API REST

- *Servidor web con implementación de API REST*
- Uso básico librería express, middlewares y arquitectura MVC
 - Uso de MongoDB como base de datos
 - Uso de docker para virtualizar servicio de bbdd
- Uso de POSTMAN para tests de la api autenticada

PROYECTO 7: AUTENTICACIÓN

- *Autenticación en servidor web*
- Uso de json web tokens
- Uso de POSTMAN para tests de la api autenticada

PROYECTO 8: WEBSOCKETS

- Uso de librería socket.io

PROYECTO 9: PROMESAS

- Uso de promesas para consumir APIs
- Uso a `async-await`

¿EMPEZAMOS?

- Configuración inicial