[Creando APIs con Node JS, Express y MongoDB](https://legacy.gitbook.com/book/gmoralesc/creando-apis-con-node-js-express-y-mongodb)

Portada

[Introducción](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/)

Configurando el proyecto con express

[Crear un simple Web Server](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/configurando-el-proyecto-con-express/crear-un-simple-web-server.html)

[Utilizando Express](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/configurando-el-proyecto-con-express/utilizando-express.html)

[Herramientas de desarrollo](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/configurando-el-proyecto-con-express/configurando-el-ambiente-de-desarrollo.html)

[Configuración y variables de entorno](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/configurando-el-proyecto-con-express/configuracion-y-variables-de-entorno.html)

[Middleware, manejo de errores y logs](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/configurando-el-proyecto-con-express/middleware-manejo-de-errores-y-logs.html)

Router y Routes en Express

[Utilizando el Router y Routes de Express](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/router-y-routes-en-express/utilizando-el-router-y-routes-de-express.html)

[Creando el layout del API](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/router-y-routes-en-express/creando-el-layout-del-api.html)

[Capturando y procesando parámetros de las peticiones](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/router-y-routes-en-express/capturando-y-procesando-parametros-de-las-peticiones.html)

Persistencia de datos con MongoDB

[Instalando y configurando MongoDB](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/instalando-mongo-db.html)

[Conectando con MongoDB](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/conectando-con-mongodb.html)

[Mongoose Models](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/mongoose-models.html)

[Procesando parámetros comunes con middleware](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/procesando-parametros-comunes-con-middleware.html)

[Estandarización de la respuesta](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/estandarizacion-de-la-respuesta.html)

[Paginación](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/paginacion.html)

[Mongoose Schemas](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/mongoose-schemas.html)

[Ordenamiento](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/ordenamiento.html)

[Creando recursos](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/creando-recursos.html)

[Relaciones entre recursos](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/relaciones-entre-colecciones.html)

[Consultar recursos anidados](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/consultar-recursos-anidados.html)

[Añadir recursos anidados](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/anadir-recursos-anidados.html)

Asegurando el API

[Añadir y remover campos de un documento](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/asegurando-el-api/anadir-y-remover-campos-de-un-documento.html)

[Encriptando información sensible](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/asegurando-el-api/encriptando-informacion-sensible.html)

[Publicado con GitBook](https://www.gitbook.com/)

[Herramientas de desarrollo](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/)

**Herramientas de desarrollo**

Antes de continuar vamos a mezclar el *feature* anterior y crear una rama para trabajar:

git checkout master
git merge feature/02-using-express
git checkout -b feature/03-add-development-tools

**Nodemon**

Es muy tedioso cada vez que realizamos un cambio en los archivos tener que detener e iniciar nuevamente la aplicación, para esto vamos a utilizar una librería llamada *nodemon,* esta vez como una dependencia solo para desarrollo, ya que no la necesitamos cuando la aplicación esté corriendo en producción, esta librería nos brinda la opción de que después de guardar cualquier archivo en el directorio de trabajo, nuestra aplicación se vuelva a reiniciar automáticamente.

npm install -D nodemon

Luego modificamos la sección de scripts del archivo package.json para añadir el *script* que ejecutará nuestra aplicación en modo de desarrollo y en modo de producción de la siguiente manera:

"scripts" : { "dev" : "nodemon" , "start" : "node index" , "test" : "echo \"Error: no test specified\" && exit 1" },

No es necesario indicarle a nodemon que archivo va a ejecutar pues por convención buscará en archivo index.js , el cual es el punto de entrada de nuestra aplicación. Mas adelante se configurará el *script* de pruebas: test

Si el servidor se está ejecutando lo detenemos con CTRL+C , pero esta vez lo ejecutamos con npm run dev y abrimos el navegador en la siguiente dirección http://localhost:3000 .

Antes de finalizar hacemos commit de nuestro trabajo:

git add .
git commit -m "Add Nodemon and npm scripts"

Más información:

[Nodemon](https://nodemon.io/)

**ESLint**

Es recomendado añadir una herramienta que nos compruebe la sintaxis de los archivos para evitar posibles errores y porque no normalizar la forma en que escribimos el código, para ello utilizaremos una librería llamada ESLint que junto con el editor de texto comprobará la sintaxis de los archivos en tiempo real:

npm install -D eslint

ESLint nos permite seleccionar una guia de estilo ya existente o también poder crear nuestra propia guia, todo depende del común acuerdo del equipo de desarrollo.

Inicializamos la configuración con el siguiente comando:

./node\_modules/.bin/eslint --init

Si ESLint está instalado de manera global, el comando anterior se podría reemplazar con: eslint --init

Contestamos las preguntas del asistente de configuración de la siguiente manera:

? How would you like to configure ESLint? Use a popular style guide
? Which style guide do you want to follow? Airbnb
? Do you use React? No
? What format do you want your config file to be in? JSON

Una vez finalizado el proceso creará un archivo en la raíz de nuestro proyecto llamado .eslintrc.json , el cual contiene toda la configuración. Adicionalmente vamos a añadir las siguientes excepciones:

Vamos a utilizar el objeto global console para imprimir muchos mensajes en la consola y no queremos que lo marque como error.

A modo de ejemplo mas adelante vamos añadir un parámetro que no utilizaremos inmediatamente pero nos servirá mucho para explicar un concepto el cual es next y no queremos que no los marque como error.

{ "extends" : "airbnb-base" , "rules" : { "no-console" : "off" , "no-unused-vars" : [ "error" , { "argsIgnorePattern" : "next" }]
}
}

Si estas utilizando VSCode puedes añadir estos parámetros en la configuración para que siempre elimine los espacios en blancos restantes a la hora de guardar el archivo pero a su vez también deja una línea en blanco al final de cada archivo, estas son buenas prácticas y son requeridas por defecto en el ESLint:

{ "editor.formatOnSave" : true , "files.insertFinalNewline" : true ,
}

Antes de finalizar hacemos commit de nuestro trabajo

git add --all
git commit -m "Add ESLint and config"

Más información:

[ESLint](https://eslint.org/)