
Desarrollo del lado servidor con NodeJS, Express y MongoDB

Comenzando un proyecto con Nodejs y Express

Hemos visto los conceptos básicos de Nodejs y NPM. Ahora vamos a crear nuestro primer proyecto Nodejs. Vamos a configurarle el repositorio GIT para luego vincularlo en Bitbucket. Mientras creamos el proyecto, vamos a repasar los principales módulos que se cargarán. Por último, vamos a levantar un servidor web en nuestro proyecto con *express*.

Comenzaremos por abrir nuestra terminal y crear una carpeta para el proyecto en nuestro espacio de trabajo. Nos dirigimos a esa carpeta y vamos a ejecutar el comando 'git init'.

En la terminal veremos que aparece un mensaje indicando que el repositorio fue creado exitosamente. Con esto ya tenemos el repositorio de git. Nos queda pendiente vincularlo a Bitbucket para alojarlo en la nube.

Vamos a Bitbucket, ingresamos con nuestra cuenta y creamos el repositorio.

Luego ejecutamos los comandos para vincular el proyecto con el creado en la nube y continuamos.

Veremos que aparece un error como este:

```
"error: src refspec master does not match any.  
error: failed to push some refs to 'git@bitbucket.org:elamonica/fullstack.git'"
```

Eso es porque todavía no hemos hecho ningún commit y, por lo tanto, no hay nada para pushear.

Vamos a dejar esta acción para más adelante. Ahora configuremos nuestro proyecto Node con express. Les presento una herramienta muy útil para comenzar un proyecto con express: el *express generation tool* (aquí pueden ver su web <https://expressjs.com/en/starter/generator.html>).

Esta herramienta nos permite crear toda la estructura de un proyecto desde línea de comandos, simplemente ingresando el nombre del proyecto. Definirá algunas dependencias útiles en todo proyecto express, que luego podremos modificar, si lo necesitamos. La estructura que se generará excede a los ejemplos que veremos en breve, pero verás que es útil tenerla disponible, ya que a medida que avancemos, con los módulos las iremos utilizando y explicando a su debido momento.

Lo primero que haremos es instalar la herramienta:

```
npm install express-generator -g
```

Si ejecutas: `express -help`, verás las opciones disponibles para la creación de tu proyecto. Podrás configurar el *view engines* que desees o dejar el que viene por defecto que se llama *jade*; lo mismo puedes hacer con los preprocesadores de css como SASS o LESS.

Los *view engine* permiten crear documentos HTML para las vistas de tu proyecto. Las vistas mezclan código estático HTML y código de programación dinámico, tales como atributos de determinados objetos de tu modelo. De esta forma, podrías tener una vista de un modelo tales como un objeto que representa una bicicleta y, luego, con el view engine, definir la vista, de acuerdo al objeto bicicleta que se utilice como parámetro de la generación de la vista. Si te resultó compleja esta definición, no te preocupes, ya que haremos muchos ejercicios y terminarás comprendiendo de punta a punta la importancia de un buen *view engine*.

Pasemos a crear nuestro primer proyecto:

```
express red-bicicletas --view=pug
```

Fíjate que definí el *view engine Pug* (<https://github.com/pugjs/pug>).

Al finalizar la creación, fijate que se encuentra el archivo *package.json* con varias dependencias. Para instalarlas debemos ejecutar el famoso *npm install*, que toma las dependencias del archivo *package.json* y las descarga en el ambiente del proyecto.

Ahora crearemos el archivo *'.gitignore'*. Es importante que lo escribas todo en minúsculas y con el punto por delante. Este nombre es una convención en el ambiente git. El archivo va a contener todas las carpetas o archivos que no nos interese que git observe para detectar cambios en los mismos y pasarlos. Para que quede más claro, cuando realizamos un cambio en nuestro espacio de trabajo y ejecutamos *'git status'*, vemos en rojo los archivos modificados. Si esos archivos están indicados en el *.gitignore* o dentro de una carpeta indicada, no nos aparecerán en rojo, es decir, git los ignorará.

Dentro del archivo escribimos *node_modules*. Esa carpeta incluirá muchos archivos de las librerías que utilizaremos en nuestro proyecto. La excluimos porque npm trae el comando *install*, que permite instalar todas las dependencias de nuestro proyecto y eso implica que se guardan los archivos de las mismas en la carpeta *node_modules*. Por lo tanto, no es necesario versionar *node_modules* ya que no aporta ningún valor y podría generar conflictos de versiones innecesarios.

Voy a usar Visual Studio Code (VS) como editor de texto y entorno de trabajo, pero puedes usar el que gustes, obviamente. Lo puedes descargar desde <https://code.visualstudio.com/>.

Abramos nuestro proyecto desde el VS y, desde la terminal, ejecutaremos *npm start*.
Puedes ver por la consola que el servidor está disponible y escuchando nuevas conexiones.

Desde un browser ingresas a <http://localhost:3000> y ver el mensaje de bienvenida de Express.

Hemos creado nuestro primer proyecto de Node y Express.

GIT está configurado y con la cuenta de Bitbucket.

A partir de ahora iremos creando nuestro modelo, siguiendo interfaces web (páginas HTML) y una API REST.

Vamos a trabajar con una arquitectura MVC, *Model View Controller*.

Paso a explicarte de que se trata esto en la siguiente lección.