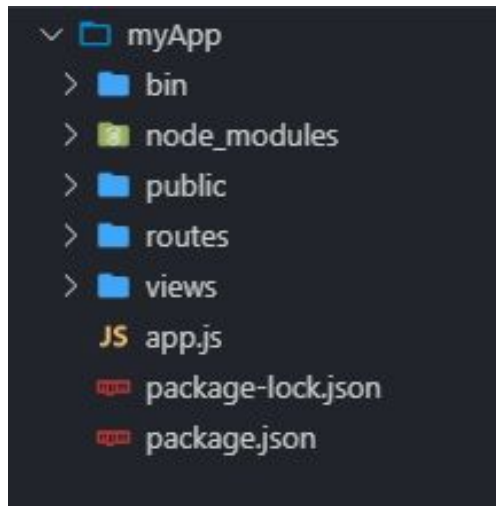


# express-generator

# ¿Qué trae **express-generator**?

Esta herramienta presenta una gran ventaja al momento de **empezar** un **proyecto** nuevo ya que, al instalarla, trae consigo un **esqueleto** de carpetas, archivos y dependencias que nos puede servir para **inicializar** cualquier aplicación.



# Estructura de **archivos**

- En la **carpeta raíz** encontramos el entry point `app.js`, y el archivo `package.json`.
- Dentro de la **carpeta bin** encontramos el archivo `www` sin extensión. El mismo trae definida una lógica interna y se encargará de hacer que la aplicación corra.
- Dentro de la **carpeta public** podremos guardar todos los recursos estáticos de nuestra aplicación: CSS, imágenes, JavaScript de front-end.
- Dentro de la **carpeta routes** estaremos administrando el sistema de ruteo de la aplicación. Encontramos los archivos `index.js` y `users.js`.
- Dentro de la **carpeta views** encontramos dos vistas iniciales que trae el generador: `index.ejs` y `error.ejs`.

“

Este generador no trae consigo la carpeta **controllers** y sus **archivos**, por lo tanto tendremos que crearla nosotros si queremos respetar la arquitectura con la que venimos trabajando.



”

# Instalando **express-generator**

Para poder trabajar con el generador de proyectos debemos instalarlo globalmente para poder usarlo en el momento que necesitemos.

```
>_ npm install express-generator -g
```

Luego, usando el comando `express` seguido de un nombre, podremos crear un proyecto de Node.js y Express. También, podemos configurar el **motor de vistas\*** que queremos usar.

```
>_ express myApp --ejs
```

\* Si no indicamos un motor de vistas (`--ejs`), se instalará por defecto Jade.

# Instalando **express-generator**

Por último, **dentro de la carpeta del proyecto**, tenemos que correr el comando para instalar todas las dependencias que vinieron configuradas en el `package.json` para que el proyecto funcione.

```
>_ npm install
```

# Motor de vistas

Los motores de vistas, también conocidos como motores de plantillas o template engines, nos permiten crear una estructura dinámica para las vistas de nuestro proyecto. Es decir, definir **bloques** de **contenido** que se pueden rellenar con **datos variables**.

Entre los más populares se encuentran EJS, Jade, Pug y Handlebars.

**EJS**

<% Embedded JavaScript %>



**pug**

**handlebars**



# ¿Qué es **nodemon**?

Esta herramienta nos permite **monitorizar** constantemente el **servidor** de Node.js.

Cada vez que hagamos modificaciones en el código de nuestro proyecto, **Nodemon** se encargará de hacer los «stops» y «reloads» de nuestro servidor en Node.js, con lo que no tendremos que estar haciéndolo manualmente.





# Iniciar el **servidor**

Dentro de la **carpeta del proyecto** debemos correr el siguiente comando, en el cual indicaremos el archivo que queremos ejecutar.

```
>_ nodemon bin/www
```

Por último, debemos ingresar a **http://localhost:3000** para comprobar que el servidor se levantó correctamente.

DigitalHouse>  
Coding School