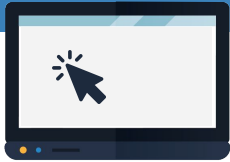


LOS CONTROLADORES

USTED ESTÁ **AQUÍ**



VISTA



CONTROLADOR



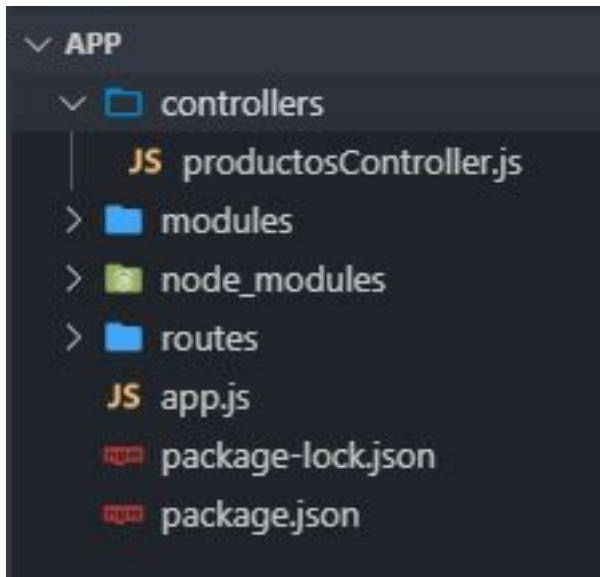
MODELO



ARQUITECTURA DE ARCHIVOS

En la carpeta raíz del proyecto, crearemos la carpeta **controllers**. Adentro almacenaremos un controlador por cada recurso que tenga la aplicación.

Cada controlador será un **módulo** que exportaremos, para luego requerirlo en donde lo necesitemos.



CREANDO UN CONTROLADOR

Crearemos un archivo para manejar nuestro recurso. Para nombrar los archivos se estila empezar con el **nombre** del **recurso** seguido de la palabra **Controller** usando el estilo *camelCase*.

```
JS productosController.js ×  
controllers > JS productosController.js  
1 // código
```

CREANDO UN CONTROLADOR

Lo primero que hay que hacer es definir una variable en el archivo del controlador del recurso y asignarle un objeto literal.

```
{  
const controlador = {};
```

Dentro del objeto, iremos definiendo los métodos que se encargarán de manejar, cada uno, un request en particular.

```
{  
const controlador = {  
  index: //mostrar listado de productos,  
  show: //mostrar detalle de un producto,  
  create: //enviar datos para agregar un producto,  
};
```

CREANDO UN CONTROLADOR

Hasta el momento, la **lógica** de cómo manejar cada request que llegaba, la veníamos escribiendo en el archivo de rutas de cada recurso, en donde definíamos una **url** y un **callback** que se encargaba de manejar esa petición.

```
// código del archivo productos.js en la carpeta routes
router.get('/', (req, res) => {
  res.send('Index de productos');
});
```



CREANDO UN CONTROLADOR

Al trabajar con el patrón MVC, podemos apoyarnos en sus reglas y dividir las responsabilidades de nuestros archivos. Siguiendo con esa línea, serán los **métodos** de cada controlador los que **recibirán** esos datos y **enviarán** la información correspondiente.

De modo que, quitaremos el callback que habíamos definido en las rutas y lo escribiremos en el método index del controlador de productos.

```
const controlador = {  
  index: (req, res) => {  
    res.send('Index de productos');  
  },  
};
```


IMPLEMENTAR UN CONTROLADOR

Primero hay que hacer visible todo el código que definimos en el controlador. Para eso, exportaremos la variable en la última línea del archivo.

```
{ } module.exports = controlador;
```

Para empezar a usar los métodos que definimos, debemos requerir el módulo dentro del archivo de ruteo del recurso, en este caso `productos.js` tro de la carpeta `routes`

```
{ } const productosController = require('../controllers/productosController');
```


IMPLEMENTAR UN CONTROLADOR

Con nuestro módulo ya visible en el archivo, es momento de terminar de configurar aquella ruta a la que le quitamos el callback.

La misma va a seguir recibiendo ese callback pero ahora no va a ser su responsabilidad definirlo. Para eso llamaremos al método **index** del controlador de productos y le pasaremos ese método como segundo parámetro. Al ser un callback no le escribimos los paréntesis.

```
{ }
```

```
router.get('/', productosController.index);
```