

SISTEMA DE RUTEO

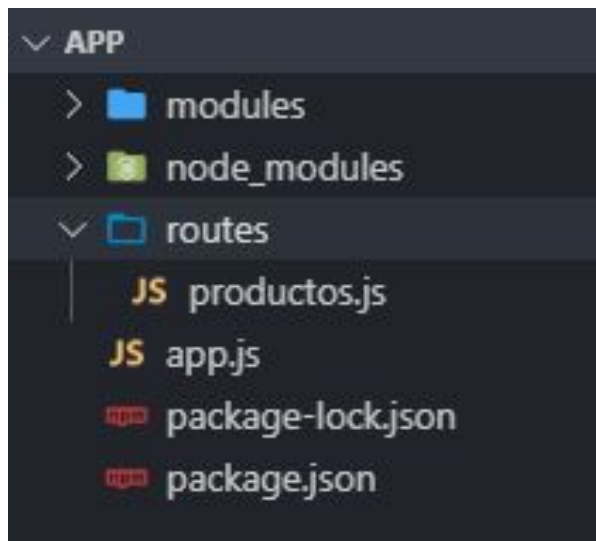
Es importante plantear y
mantener un **orden estructural**
cuando desarrollamos.



ARQUITECTURA DE ARCHIVOS

En la carpeta **routes** guardaremos, por cada recurso, un archivo `js` que administre los **request** a las rutas que tengan que ver con ese recurso.

Cada archivo `js` será un **módulo** que exportaremos, para luego requerir dentro del entry point de la aplicación: `app.js`



SISTEMA DE RUTEO

Para empezar a definir el sistema de ruteo de cada recurso, hace falta situarse dentro del **archivo** js del recurso, requerir el **módulo express** y guardar la **ejecución** del método `Router()` que nos provee Express en una variable nueva.

```
{}
```

```
const express = require('express');  
const router = express.Router();
```

Este método nos va a permitir **modularizar** por completo todo el **sistema** de **ruteo** de una manera sencilla.

SISTEMA DE RUTEO

Definimos las rutas que consideremos necesarias para manejar distintos tipos de **request**.

```
// Ruta raiz de los productos / Inicio
router.get('/', (req, res) => {
  // código
});
// Ruta que muestra el detalle de un producto
router.get('/detalle/:id', (req, res) => {
  // código
});
```

{ }

SISTEMA DE RUTEO

En la última línea del archivo **exportamos** todo el contenido de **router** para hacerlo visible.

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

Para implementarlo dentro de `app.js`, creamos una constante y requerimos el módulo.

```
{ } const rutasProductos = require('./routes/productos');
```

SISTEMA DE RUTEO

Por último, hacemos uso del método `use()` que recibe dos parámetros. El primero un string que será el **nombre** del **recurso**, en este caso *productos*. Al ser una ruta debe empezar con la `/`.

El segundo el nombre de la constante en la que almacenamos el módulo del recurso.

```
{ }
```

```
app.use('/productos', rutasProductos);
```

De esta forma estamos definiendo que cada solicitud del recurso **productos**, sea atendida por el módulo **rutasProductos** y toda su lógica.