Curso de Single Page Application con JavaScript Vanilla

# Introducción a SPA

**SPA** Son aplicaciones construidas con JavaScript que nos permiten cargar nuestro contenido una sola vez, es decir, se envían los archivos HTML, CSS y JS una sola vez al navegador y ahí es donde va a vivir toda nuestra aplicación, de tal forma que la aplicación no necesita ser recargada, por ello, si se necesita navegar en la aplicación, se navega entre secciones y no páginas.

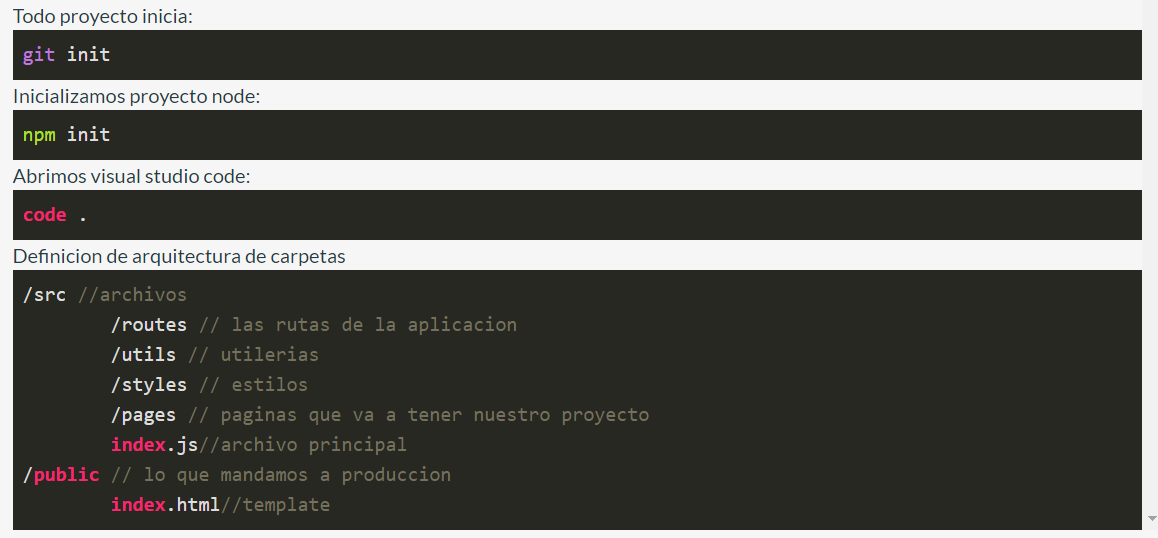
*Ventajas*

* Son fáciles de debuggear.
* Son fáciles de crear.

*Desventajas*

* No es tan compatible con el SEO.
* No es recomendable aplicarlas para grandes aplicaciones (Aplicación grande => más de 1000 usuarios y más de 50 secciones en la página).

# Configurar el entorno de trabajo



# Preparar Webpack

npm i @babel/core babel-loader html-webpack-plugin webpack webpack-cli webpack-dev-server --save-dev

**Dependencias:**

**@babel/core :** incluye toda la libreria de babel.

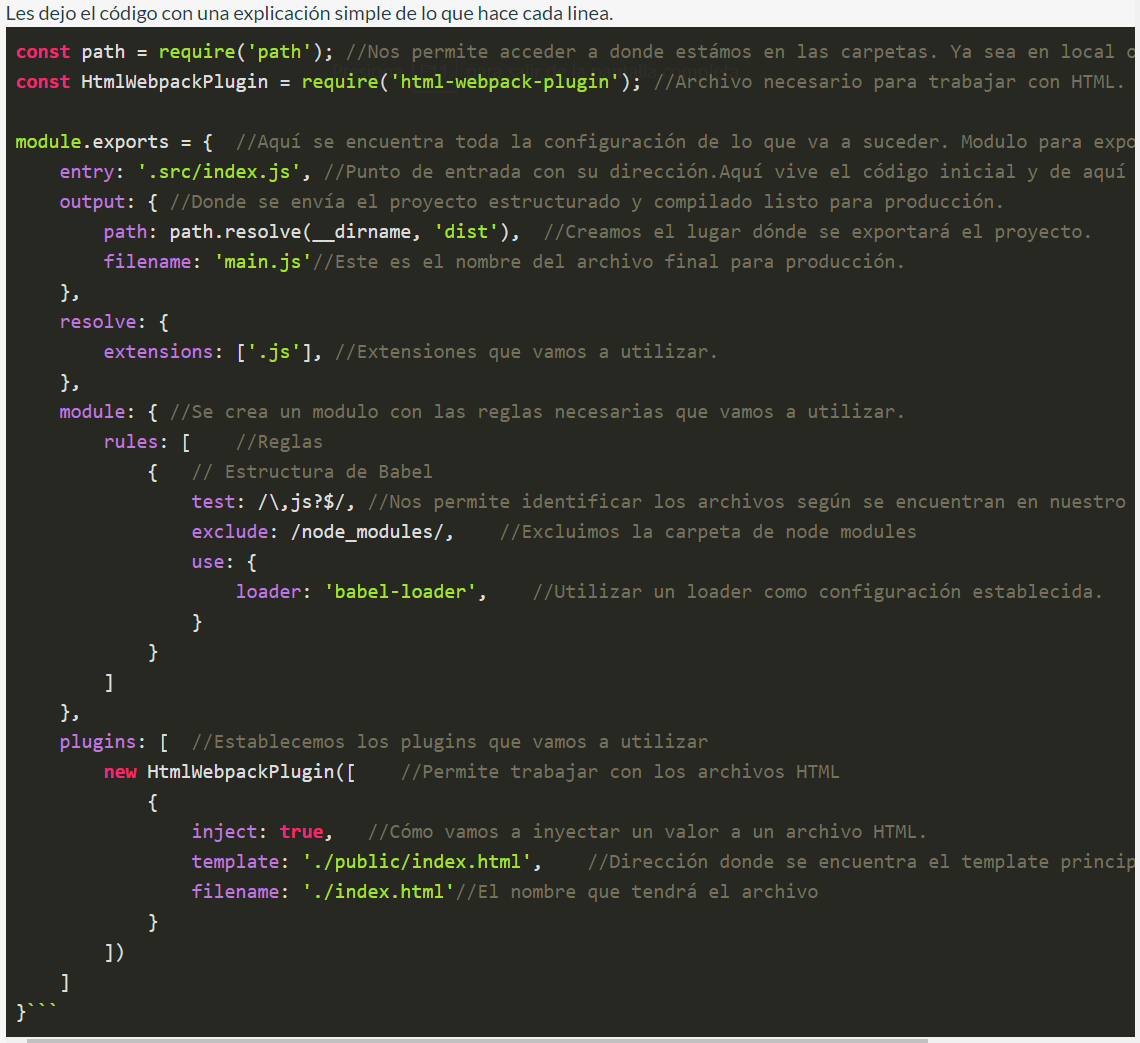
**babel-loader :** Este paquete permite transpilar archivos JavaScript utilizando Babel y webpack .

**html-webpack-plugin**: Permite copiar los archivos HTML del proyecto.

**webpack-cli:** Permite manejar ciertos comandos

**webpack-dev-server** : Permite crear un entorno de desarrollo local para probar los cambios que se esta realizando.

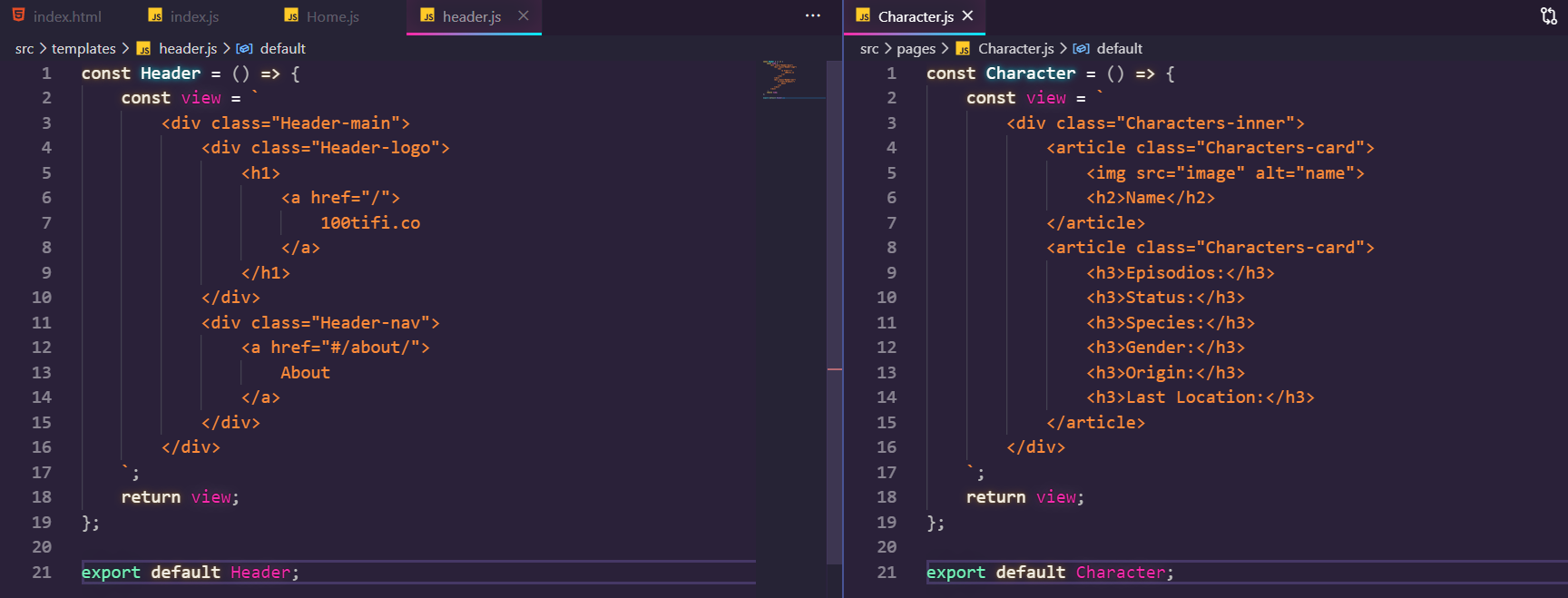
**–save-dev** : Permite instalar un conjunto de dependencia en la modalidad de desarrollo para el archivo packet jsom



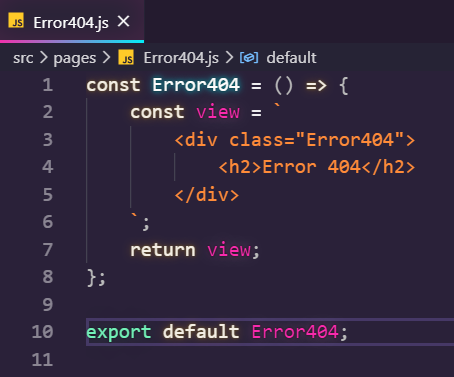
# Crear el HomE



# Crear template de personajes

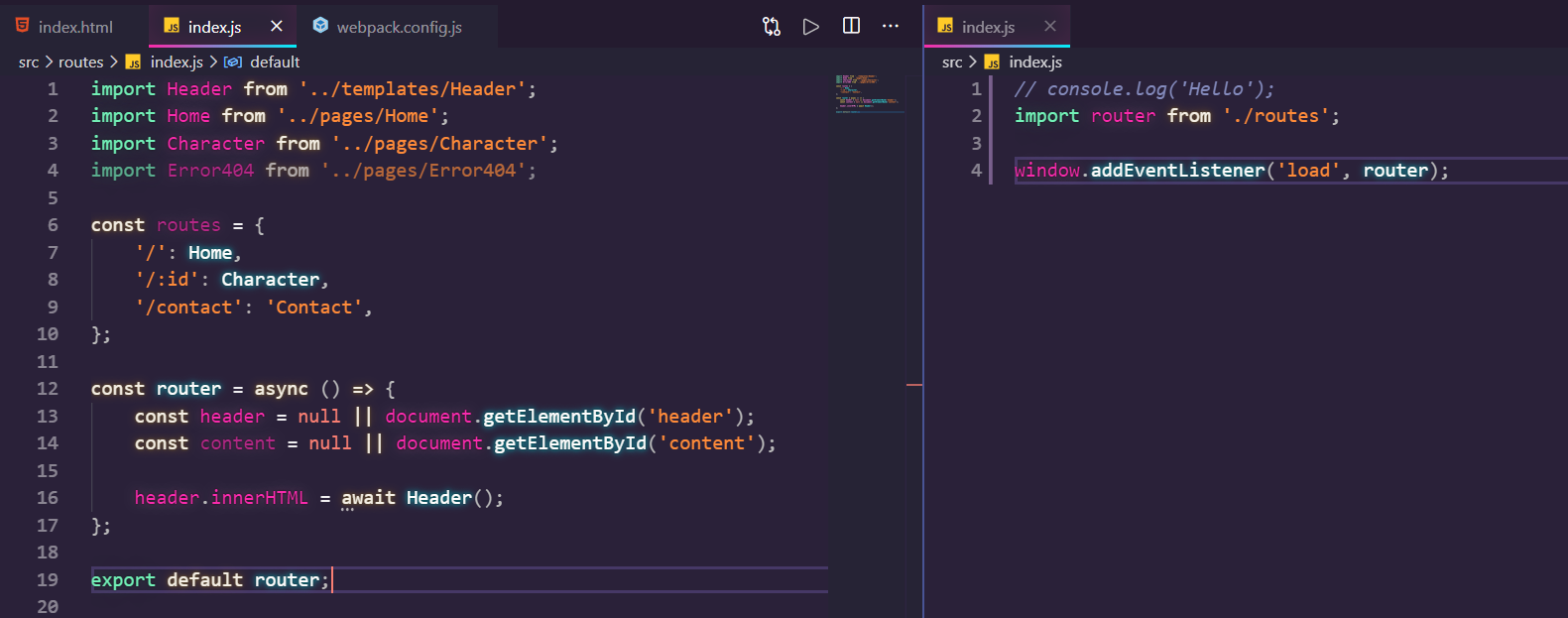


# Crear página de error 404

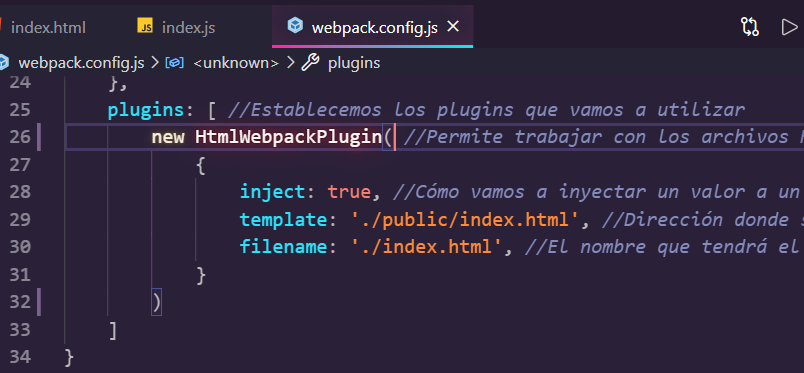


Router

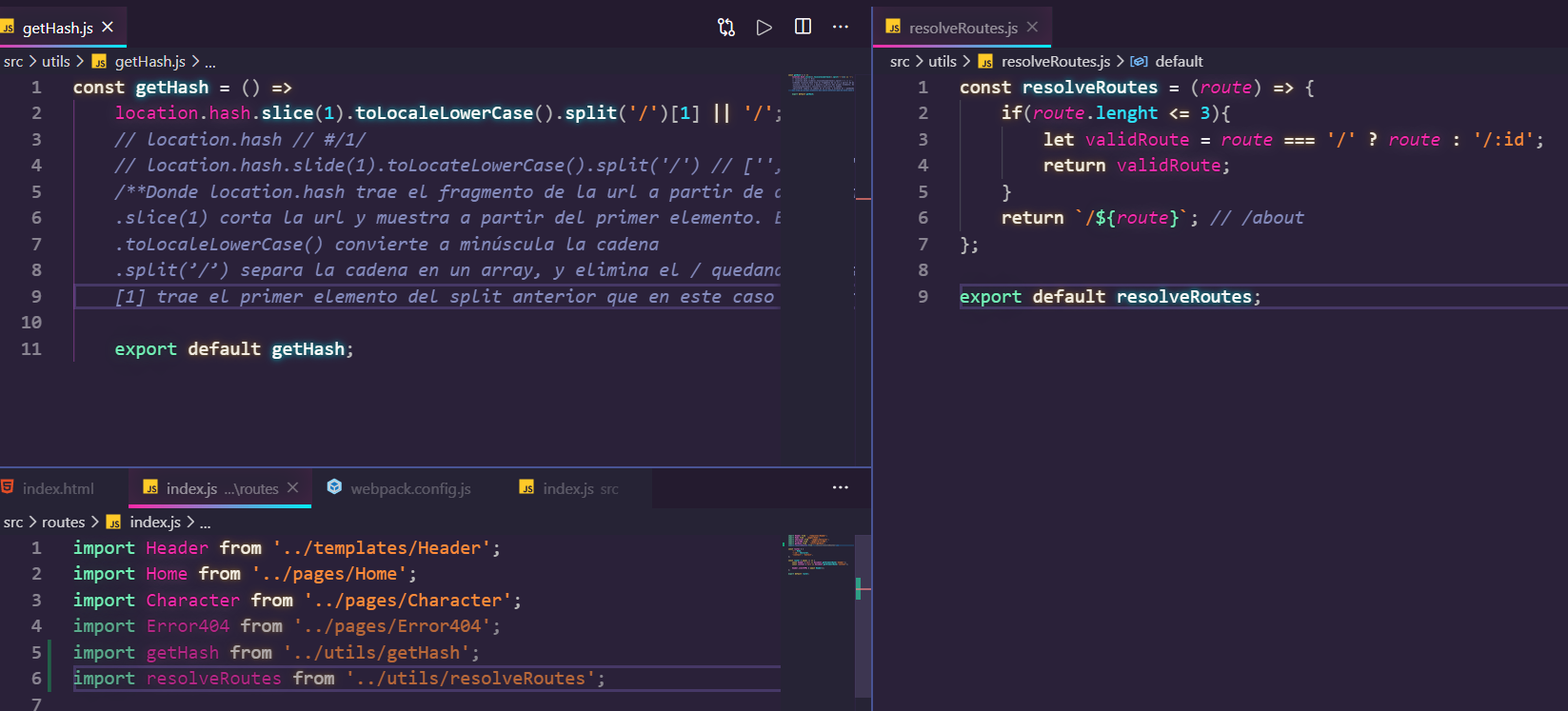
# Crear rutas del sitio

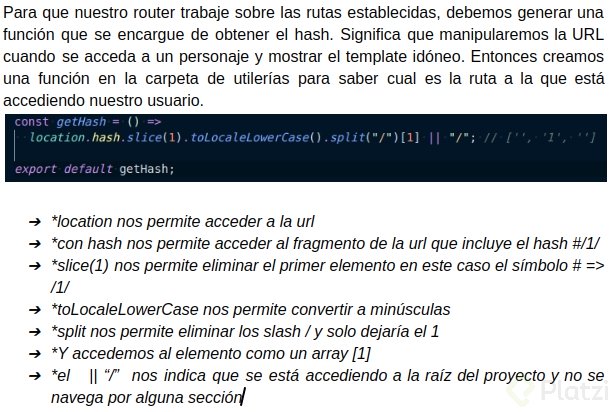


Correccion, se eliminan [] innecesarios en las llaves interiores {[inject, template, filename]}

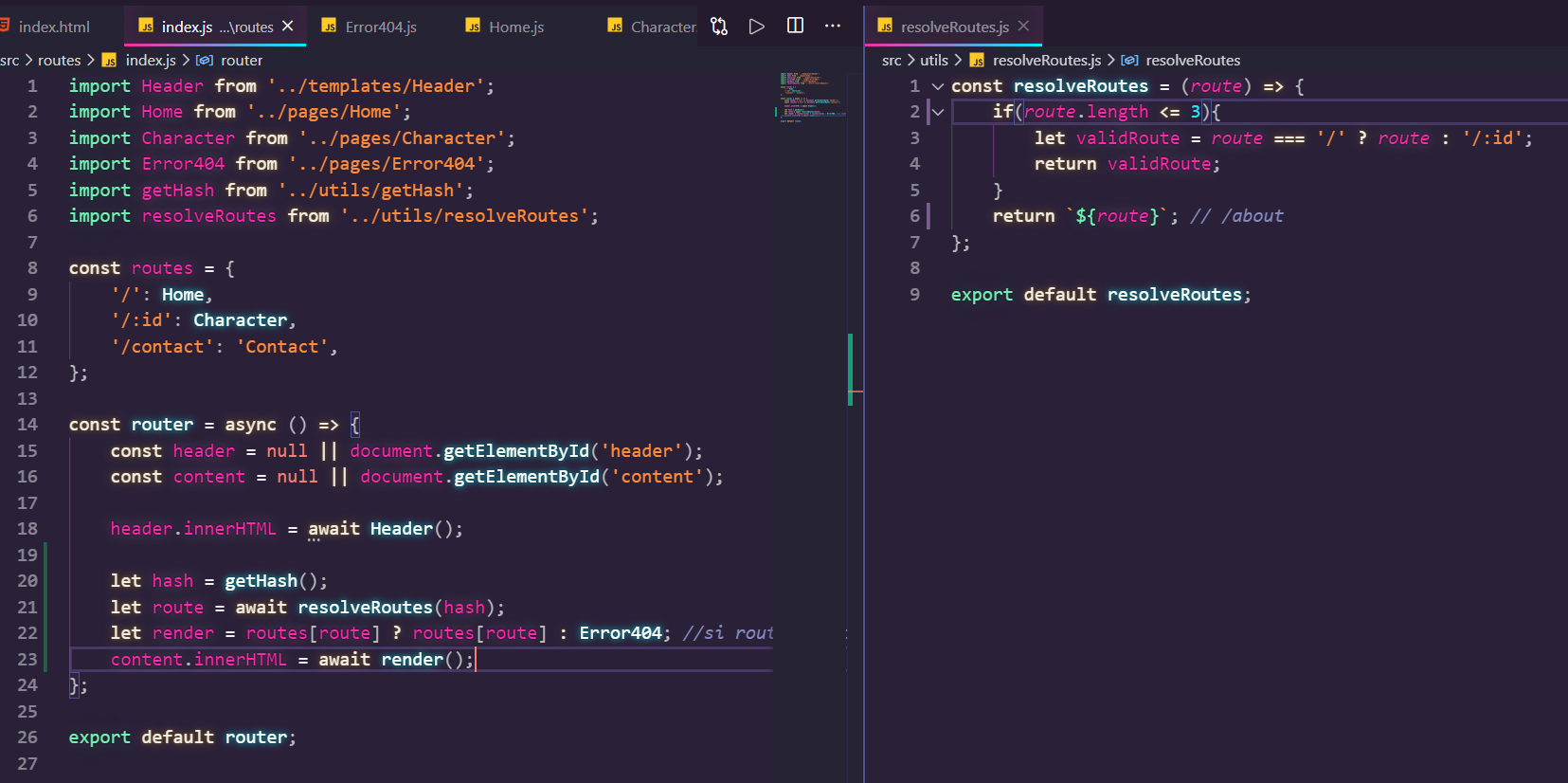


# Conectar las rutas con los templates



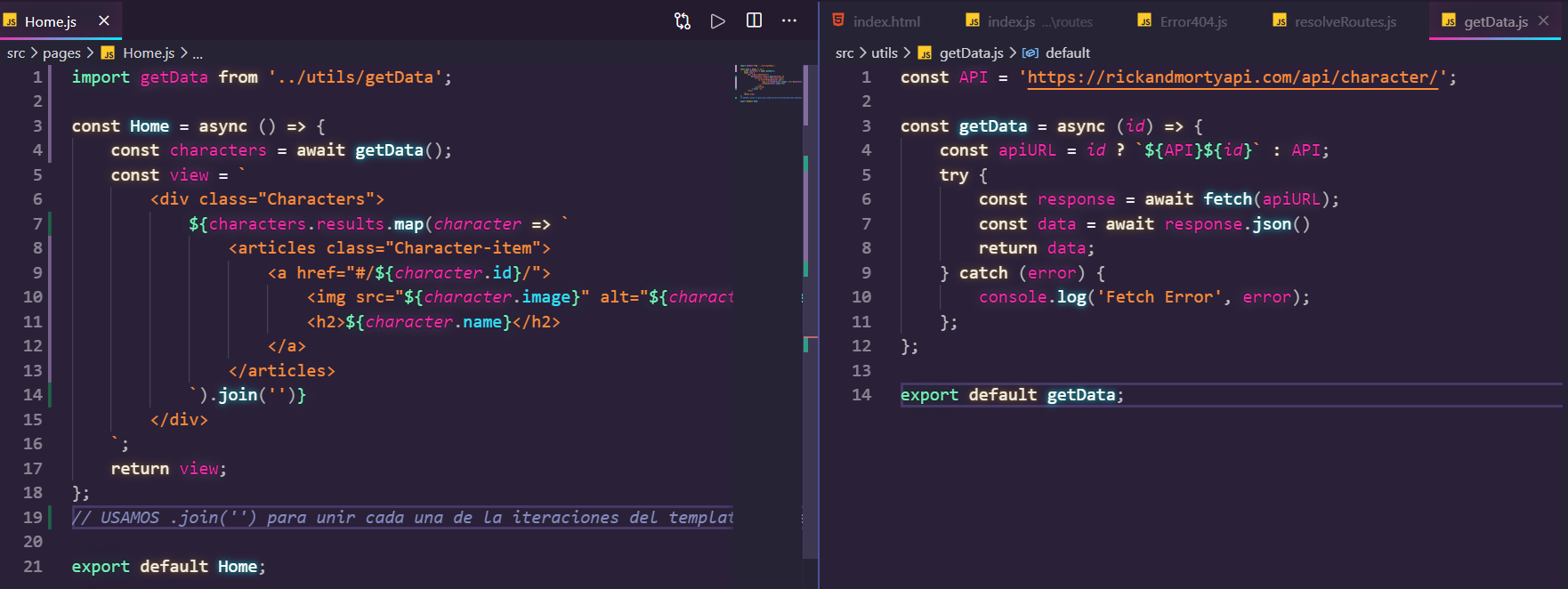


# Implementar y probar las conexiones



Fetch Data

# Obtener personajes con la función de llamado a la API



# Conectar la función con la descripción de personajes

