

ReactJS Developer

Módulo 9

Deploy

Construir la aplicación React

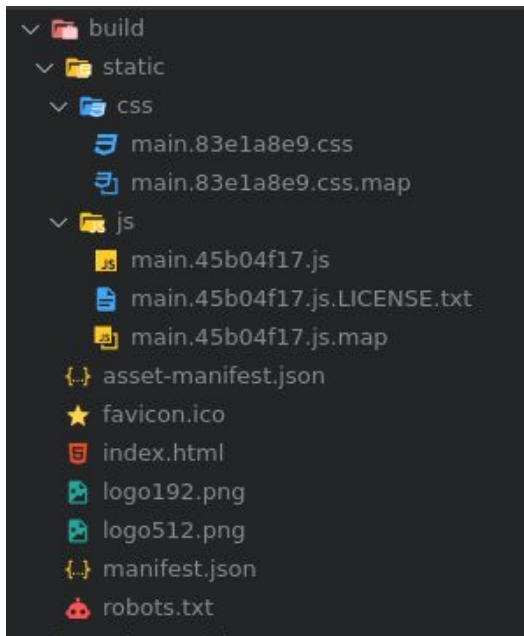
Para construir el resultado final de la aplicación, vamos a seguir los siguientes pasos:

1. Abrimos **la carpeta de nuestro proyecto**.
2. Ejecutamos en esa carpeta el comando **npm run build**.
3. Este comando generará **el resultado final de nuestro proyecto**. En concreto, hará lo siguiente:
 - a. Creará un nuevo directorio llamado **/build** (si ya existe, se borra y se vuelve a crear).
 - b. Tomará los archivos de la carpeta **/public** y los copiará directamente al directorio **/build**.
 - c. Tomará el archivo **/src/index**, analizará las dependencias, ejecutará la compilación y generación por **Webpack/Babel/TypeScript**, y copiará a **/build** el resultado final de dicha generación (y no el código en sí).

Estructura del resultado final

- **/build/static:** dentro de este directorio se encuentra el **resultado final generado a partir del /src/index**. No se recomienda cambiar su contenido directamente en el *build*.
- **/build/index.html:** es el **archivo HTML principal en el que se carga la página**. Puedes editar su contenido en **/public/index.html**, pero ten en cuenta que Webpack hace cambios a este archivo durante la compilación.

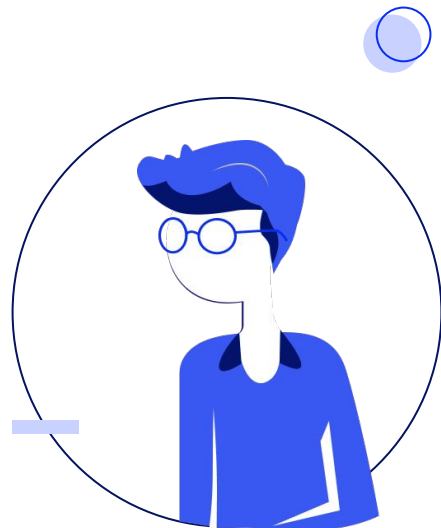
Cuando ejecutamos el **Build de React** este copia directamente en el directorio **/build** todo el contenido de **/public**, respetando la estructura de archivos y carpetas que existía en **/public**.



Requisitos para subir una SPA sin History

¿Qué requisitos de infraestructura necesito para una aplicación React sin *History*?

- Un servidor para **páginas estáticas**.
- Que soporte **HTML / CSS / JS (todos)**.
- Si no usas el **BrowserRouter** (que utiliza la *API History*) tienes una gran oportunidad de abaratar costos, ya que no necesitas nada más que un lugar donde publicar HTML/CSS/JS. Hay hasta opciones gratis como *GitHub Pages*, por ejemplo.



Requisitos para subir una SPA con History

¿Qué requisitos de infraestructura necesito para una aplicación React con History?

- Un servidor para **páginas estáticas**.
- Que soporte **HTML / CSS / JS (todos)**.
- Si usas el **BrowserRouter** (que utiliza la API History) tienes que contemplar también el costo del servidor (necesario para realizar el *fallback*). Este es, en muchos casos, un costo extra que debes afrontar. Asegúrate, por lo tanto, de que sea la mejor opción posible.
- En general las opciones son pagas, y depende de la tecnología utilizada para el back-end. En esto puedes usar la que más te convenga (si tienes créditos Azure, por ejemplo, puedes usar eso). Hay alternativas gratuitas como Heroku, que soporta Node.JS.

**¡Sigamos
trabajando!**