

Despliegue en GitHub Pages con Parcel

Modo de desarrollo

Cuando vimos que Parcel se debía arrancar escribiendo en una terminal el comando `$ npx parcel src/index.html`, realmente lo que estamos haciendo es ejecutando una versión abreviada del modo de desarrollo de Parcel, que se inicia con el comando `$ npx parcel serve src/index.html` (modalidad por defecto).

Por tanto, lo que estamos haciendo con el modo de desarrollo es iniciar un **servidor local** para ir viendo los resultados en el navegador mientras editamos cómodamente. Sin embargo, en algún momento llegará la hora de subir o desplegar nuestro proyecto a un servidor o alojamiento, lo que comúnmente llamamos un **servidor de producción**.

Para ello, debemos saber que no se debe subir la carpeta generada con el modo de desarrollo, sino que tenemos que utilizar el modo de producción o modo build de Parcel, que generará esos archivos finales de una forma más apropiada.

Modo de producción

El modo de producción de Parcel se consigue ejecutando el **modo build** en lugar del modo `serve`

```
$ npx parcel build src/index.html
```

Esto generará los archivos finales en la carpeta **dist** (por defecto).

Modo de producción con GitHub Pages

Para hacer el despliegue de nuestros proyectos con GitHub Pages debemos configurar varios scripts de NPM en el package.json del proyecto que utilizarán Parcel para:

`npm run start`: Crea un servidor local para tareas de desarrollo

`npm run build`: Crea una nueva build en la carpeta build

`npm run deploy`: Sube el contenido de build y lo despliega en la rama gh-pages

Paso a paso

1. Añadimos a nuestro archivo de .gitignore las siguientes carpetas:

```
git .gitignore
1  node_modules/
2  .cache
3  dist/
4  build/
```

2. En el archivo del package.json, incluimos los siguientes scripts:

```
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1",
  "start": "parcel serve -d dist src/index.html",
  "build": "parcel build -d build --public-url /javaspotify/ src/index.html",
  "deploy": "gh-pages -d build"
},
```

Observa que habría que modificar **/javaspotify/** con el nombre del repo de GitHub para que funcione correctamente.

3. Instalamos el paquete gh-pages con el comando `$ npm install -D gh-pages`
4. Nos aseguramos de tener nuestro repo de GitHub completamente actualizado con los últimos cambios que hemos realizado.
5. Ejecutamos el comando `$ npm run build`
6. Ejecutamos el comando `$ npm run deploy` (esto llevará unos segundos)
7. Confirmamos que se ha realizado el despliegue correctamente accediendo a la URL que nos ha asignado GitHub

