

# Preparar playground

---

Vamos a necesitar escribir código en javascript ES6 (ECMAScript 6), y utilizar librerías como CommonJS para trabajar con módulos, y orquestar todo para poder ejecutar la aplicación en un navegador.

¿Parece algo complejo verdad? Pero necesario. Como vemos, una aplicación de frontend con javascript es algo más que un fichero html, un script y una hoja de estilos css. Vamos a crear aplicaciones complejas, y vamos a necesitar una arquitectura más compleja que la anterior. Vamos a organizar nuestra aplicación con una estructura de ficheros, utilizar varios módulos de javascript que necesitarán comunicarse entre ellos, y modificar nuestro documento HTML. Además, queremos que nuestra aplicación funcione en distintos navegadores, y distintas versiones de los mismos.

## Comencemos por el principio: instalar NodeJS

NodeJS es un entorno de ejecución para Javascript. Nos ayudará a orquestar la ejecución de nuestro código. Podemos instalarlo desde su web: [Node.js](https://nodejs.org/)

## Gestionar paquetes con NPM

NPM (Node Package Manager) es un administrador de paquetes para javascript. Nos ayudará a instalar y gestionar librerías que utilizaremos durante el desarrollo.

Podemos tener más información y buscar paquetes directamente en su web: [npm | build amazing things](https://npmjs.com/)

Podemos utilizar NPM directamente desde el terminal de Visual Studio Code (Ctrl + ñ para abrirlo).

### Inicializar nuestra aplicación con NPM

Primero, crearemos una carpeta nueva para alojar nuestra aplicación, y la abriremos desde Visual Studio Code. Abriremos el terminal con <Ctrl + ñ> y ejecutaremos el siguiente comando:

```
npm init -y
```

Después de esto, veremos que nos ha creado un fichero *package.json*. Este fichero contendrá información sobre la aplicación y la dependencia de paquetes que necesita, entre otras cosas.

### Instalar paquetes

Para instalar un paquete utilizaremos el comando *npm install [paquete]* (abreviado *npm i [paquete]*). Podemos buscar paquetes desde su propia web (y ver el comando de instalación del mismo).

Cada paquete que se instala, se añadirá como dependencia en el fichero *package.json*, de forma que si ejecutamos *npm install* se instalarán tooodas las dependencias que se encuentren en ese fichero (todos los paquetes que necesita nuestra aplicación).

## Instalar un bundler: Bienvenido Parcel

**Parcel** es un empaquetador (bundler) de aplicaciones web. Se encargará de empaquetar en un fichero javascript (bundle) todo lo necesario para que nuestra aplicación se ejecute sin problemas.

Para instalar Parcel ([🔗 Empezar](#)), utilizaremos el siguiente comando de *npm* desde nuestro terminal de Visual Studio Code:

```
npm install -g parcel-bundler
```

## Crear estructura de proyecto

Dentro de la carpeta donde estamos trabajando (y que ya tenemos el fichero `package.json`), vamos a crear un nuevo fichero `index.html` con el siguiente contenido:

### index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
  />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Playground</title>
  </head>
  <body></body>
</html>
```

Ahora vamos a crear una carpeta `src`, y dentro un fichero `index.js` con el siguiente contenido:

### index.js

```
document.write("Hello Playground!");
```

Y añadiremos la referencia a este fichero en nuestro `index.html` anterior, que quedaría así:

### index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
  />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Playground</title>
    <script src="./src/index.js"></script>
```

```
</head>
<body></body>
</html>
```

Vamos a probar que todo ha ido bien y tenemos nuestro Playground listo para funcionar con NodeJS y Parcel. Vamos a ejecutar el siguiente comando en el terminal de VS Code:

```
parcel index.html
```

Cuando termine, si todo ha ido bien, deberíamos ver un mensaje como este:

```
Server running at http://localhost:1234
√ Built in 1.73s.
```

Y veremos nuestra aplicación corriendo en la URL anterior, con el mensaje *Hello Playground!*

Ya estamos preparados para desarrollar aplicaciones web complejas 😊

Nota: podemos añadir en la sección *scripts* del fichero *package.json* un nuevo elemento *start*, con el comando *parcel index.html* para poder lanzar nuestra aplicación a partir de *npm start*:

```
"scripts": {
  "start": "parcel index.html",
},
```

Ahora podemos ejecutar nuestra aplicación con *npm start*