

Webpack

DEV.F
DESARROLLAMOS(PERSONAS);

dev

Webpack

Es un empaquetador de módulos para crear entornos de desarrollo y bundles para ambientes productivos.

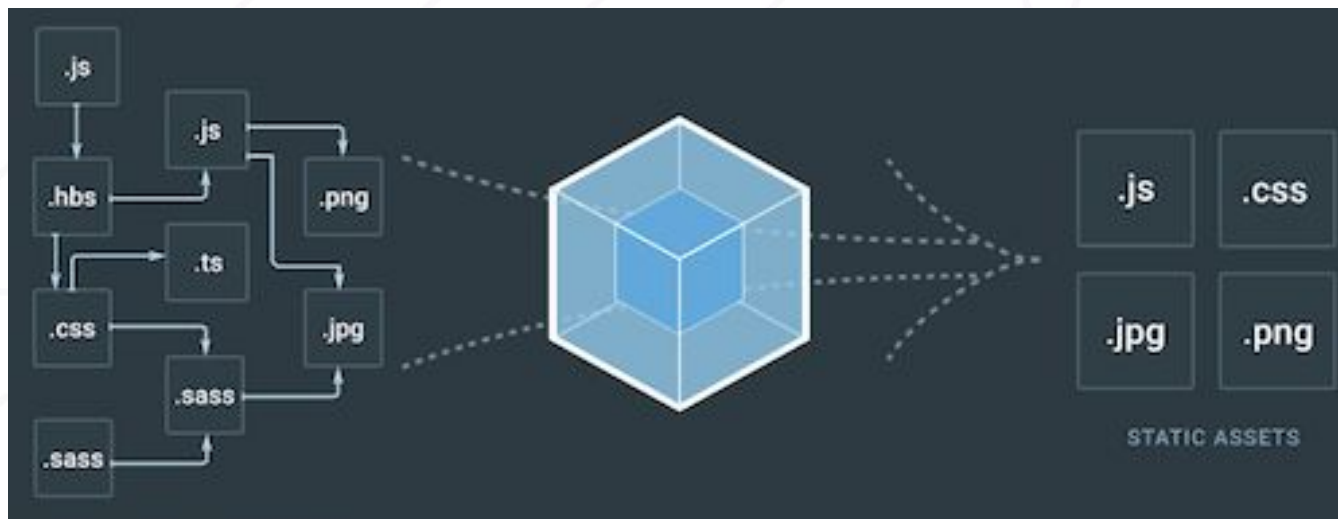
Trabaja con archivos estáticos de front como son:

- HTML.
- CSS.
- JavaScript.
- Archivos estáticos.
- Imágenes.
- Fuentes.
- Iconos.



Uso de webpack

- Builder de proyectos front end (html, css, js).
- Administrador de la configuración local y de producción.
- Es utilizado por los CLI de herramientas como Vue, React y Angular.
- Minificación de código.
- Simplifica la importación y exportación de módulos.
- **create-react-app usa webpack, babel y otras herramientas por detrás.**



¿Por qué se utiliza?

- Las herramientas de desarrollo front end modernas, utilizan sintaxis especial sobre el código escrito en html, css y js. Por ejemplo:
 1. **Vistas** => **HTML** => **React (jsx)** ó **Angular, Vue**
 2. **Estilos** => **CSS** => **Sass, Scss, LESS y Stylus**
 3. **Interactividad** => **JS (es5)** => **ES6 y posterior + frameworks y librerías de front end**

Hola mundo en Html nativo

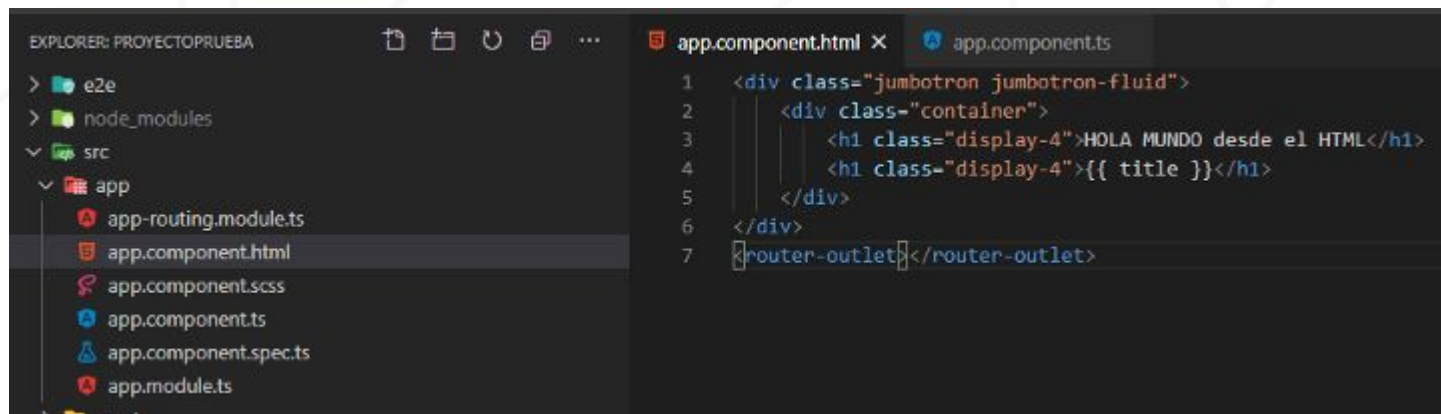


```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <meta http-equiv="X-UA-Compatible" content="ie=edge" />
7     <title>Mi primer hola mundo</title>
8   </head>
9   <body>
10    <h1>Hola mundo!!</h1>
11  </body>
12 </html>
```

Hola mundo en React

```
1 import React from 'react';
2 import { render } from 'react-dom';
3 import Hola from './componente'
4
5 render(<Hola /> , document.getElementById('root'))
6
7 /*
8  render es la funcion que se comunica con el virtualDOM y le dice que mostrar
9  en pantalla.
10
11  Recive dos parametros: Que renderizar(JSX) y en donde renderizarlos
12  */
13
```

Hola mundo en Angular

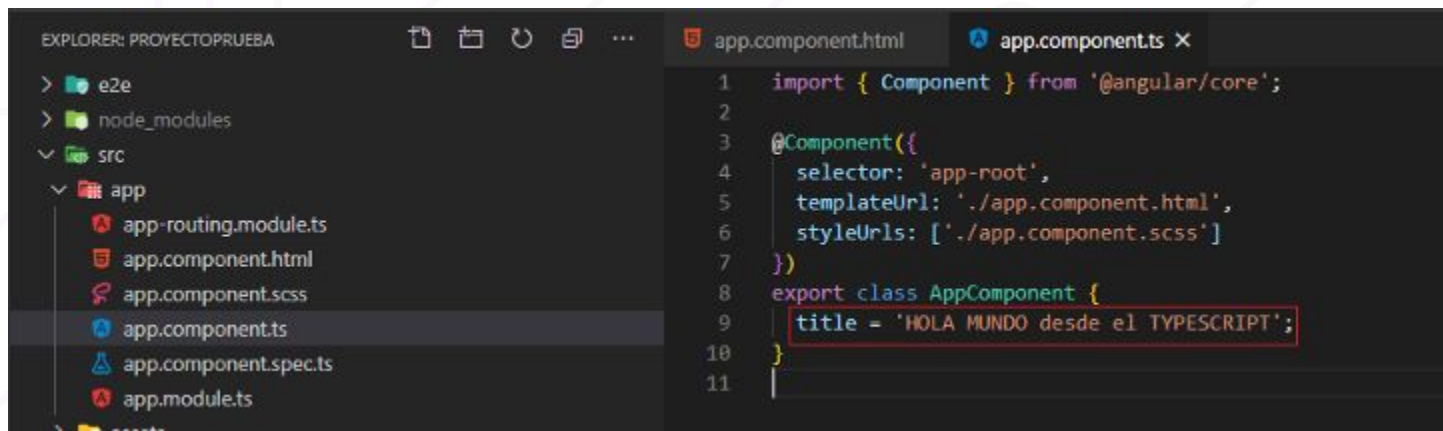


EXPLORER: PROYECTOPRUEBA

- > e2e
- > node_modules
- ✓ src
 - app
 - app-routing.module.ts
 - app.component.html
 - app.component.scss
 - app.component.ts
 - app.component.spec.ts
 - app.module.ts

app.component.html

```
1 <div class="jumbotron jumbotron-fluid">
2   <div class="container">
3     <h1 class="display-4">HOLA MUNDO desde el HTML</h1>
4     <h1 class="display-4">{{ title }}</h1>
5   </div>
6 </div>
7 <router-outlet></router-outlet>
```



EXPLORER: PROYECTOPRUEBA

- > e2e
- > node_modules
- ✓ src
 - app
 - app-routing.module.ts
 - app.component.html
 - app.component.scss
 - app.component.ts
 - app.component.spec.ts
 - app.module.ts

app.component.ts

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.scss']
7 })
8 export class AppComponent {
9   title = 'HOLA MUNDO desde el TYPESCRIPT';
10 }
11
```

Hola mundo en Vue

```
<template>
  <div class="hello">
    Hello world
  </div>
</template>

<script>
  export default {
    name: "HelloWorld"
  };
</script>

<style scoped lang="scss">
</style>
```


Babel

DEV.F
DESARROLLAMOS(PERSONAS);

dev

Babel

Babel es una herramienta que **nos permite transformar nuestro código JS de última generación (o con funcionalidades extras) a un código de Javascript que cualquier navegador o versión de Node.js pueda entender.**

Babel **funciona mediante plugins** para que le indiquemos qué cosas queremos que **transforme**, por ejemplo con el plugin `babel-plugin-transform-es2015-arrow-functions` podemos decirle que transforme las arrow functions de ECMAScript 2015 a funciones normales, con `babel-plugin-transform-react-jsx` podemos hacer que entienda código de JSX y lo convierta a código JS normal.

Compiladores

DEV.F
DESARROLLAMOS(PERSONAS);

dev

Compiladores

Son herramientas que convierten el código fuente en código entendible para la máquina. Como JS no es un lenguaje compilado, sino interpretado no se requiere compilar. Sin embargo, cuando se escribe sintaxis de ES6, de frameworks u otras herramientas modernas, se requiere de un proceso para convertir todos esos archivos y formato de código en algo entendible para el navegador.

En jerga informática, al proceso anterior se le conoce como compilación o más precisamente transpiración de código.

Linters

DEV.F
DESARROLLAMOS(PERSONAS);

dev

Linters

- **Configuración base de VSCode:** Se guarda en el **settings.json** y **editor.config** de vscode y nos permite definir el estándar de codificación de nuestro editor.
- **Eslint:** Es una herramienta de resaltado de sintaxis y es customizable con reglas que se pueden definir en el archivo **.eslintrc.json**.
- **Prettier:** Es un formateador de código que puede autocorregir automáticamente nuestro código según el conjunto de reglas que establezcamos en el archivo **.prettierrc.json**.

En qué momento se pueden usar?

- Al guardar.
- Al pegar código.
- Al ejecutar un comando npm (programado por nosotros).
- Antes de hacer un commit.
- Antes de desplegar a algún entorno.

Librerías / Frameworks front end



Liberia vs Framework

Una **librería** es un paquete de código que **nos ayuda a resolver un problema concreto** por medio de scripts específicos mientras que un **framework** es un paquete de código (generalmente mas grande) que **nos permite resolver un conjunto de problemas mediante un set de herramientas**.

Ejemplos de librerías:

- React.
- Selector.
- jQuery.

Ejemplos de frameworks:

- Angular.
- Spring.
- ExpressJS.
- VueJS.
- NextJS.