

Fullstack Web Bootcamp

Frontend PRO

01/12/2025

Índice

Introducción	03
Bundlers	12
TypeScript	21
Proyecto	30

Introducción





Desarrollador Fullstack, exalumno (reincidente) de KeepCoding, amante de Typescript y todo el ecosistema Javascript.
Entusiasta de la formación y divulgación en todo aquello que esté relacionado con la tecnología y el emprendimiento.

Requisitos

- Tener soltura con VSCode. Vamos a crear, mover y cambiar archivos.
- Tener soltura con GIT
- Comprender los conceptos del módulo de desarrollo frontend con Javascript
- Entender y saber utilizar los ESModules de Javascript (import / export).
- Entender los scripts de NPM.

Objetivos

- Entender el ecosistema de herramientas en el que me puedo apoyar en mi día a día.
- Cómo trabajar en frontend de una manera más profesional.
- Entender arquitecturas frontend.
- Saber lo que es un bundle, build o una versión empaquetada.
- Trabajar con preprocesadores de CSS (SASS).
- Trabajar con Typescript

¿Qué construiremos?

Proyecto para lograr nuestros
objetivos



S O B R E L O S J U E G O S

La Copa Mundial de Quidditch se ha celebrado cada cuatro años desde 1473. La competencia se lleva a cabo entre los equipos de Quidditch que representan a los países de todo el mundo.

El número de países participantes varía de torneo en torneo. Cualquier país puede inscribir un equipo dentro de los doce meses posteriores a la final anterior.

P R Ó X I M A E D I C I Ó N

12

5

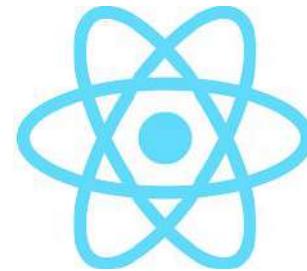
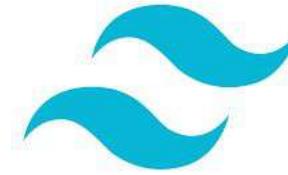
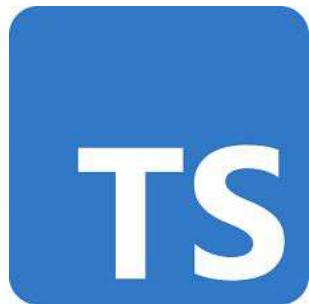
43

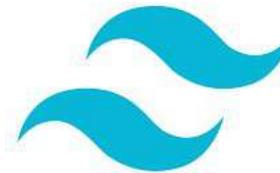
42

¿Qué utilizaremos?

Ecosistema Frontend Moderno

keep coding

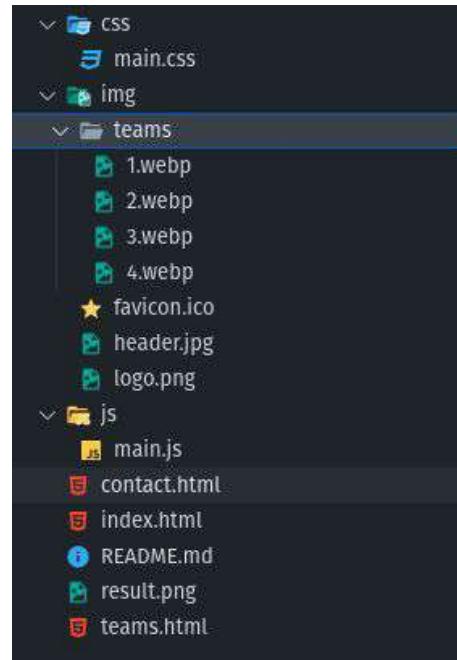
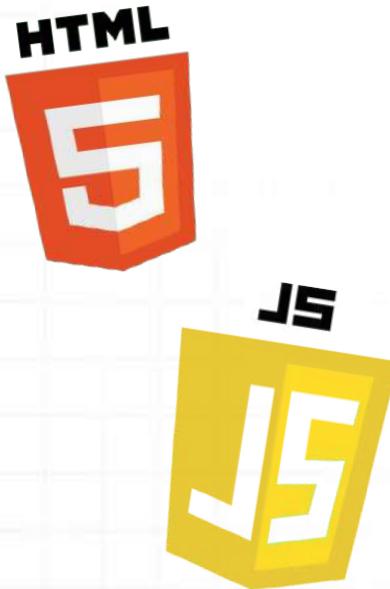




Bundlers



¿Os suena?



HTML + CSS + JS

¿Es lo correcto?

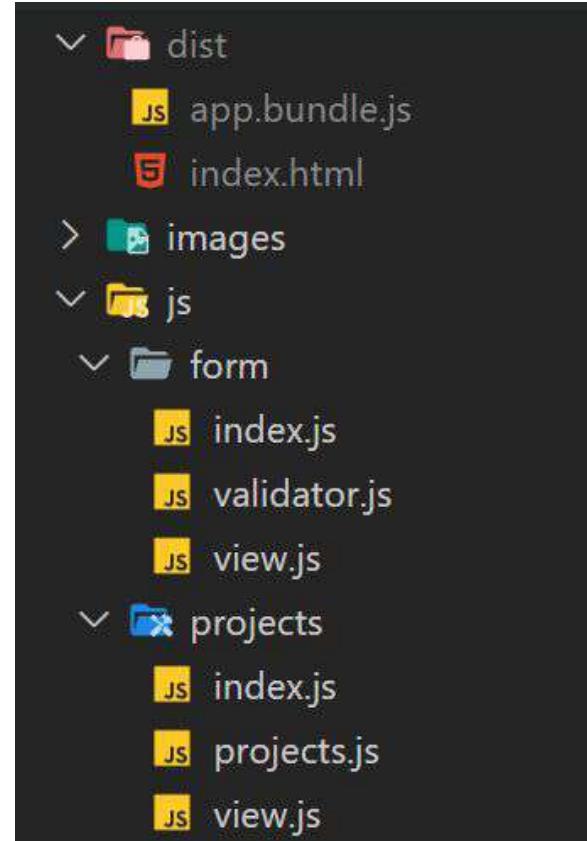
- Son los lenguajes que entiende el navegador 
- Ocupan poco espacio 
- Genera código desordenado 
- Si queremos optimizar, duplicamos código 
- Es difícil de mantener 
- Implica mucha gestión de recursos si queremos optimizar

- Nos tenemos que encargar de la retrocompatibilidad 



¿Solución? Bundlers

Un JavaScript Bundler nos permite empaquetar de forma rápida un conjunto de clases o funciones de JavaScript, CSS y archivos estáticos que se encuentran diseminadas en ficheros independientes como un único fichero de tal forma que en el navegador pueda ser cargado como una única referencia de forma rápida y transparente para el desarrollador.



Timeline

Gulp

Transformación de archivos

Webpack

Empaquetador y optimizador

Vite / Parcel

Entornos de desarrollo profesionales

Parcel

The zero configuration build tool for the Web

Parcel combines a great out-of-the-box development experience with a scalable architecture that can take your project from just getting started to massive production application.



Talk is cheap

Show me the code



Instalar parcel

```
npm init -y
```

```
npm i -D parcel
```



Estructura básica

- node_modules/
- package.json
- package-lock.json
- src/
 - index.html



TypeScript



¿Os suena?

```
✖ ▼Uncaught TypeError: result.split is not a function
    at Object.add (<anonymous>:8:19)
    at testing (<anonymous>:11:20)
    at <anonymous>:12:3
        add      @ VM10084:8
        testing @ VM10084:11
        (anonymous) @ VM10084:12
```

JavaScript

¿Es factible en producción?

- Es el lenguaje que interpretan los navegadores 
- Los navegadores modernos adoptan el estándar relativamente rápido 
- Los errores solo son detectables en ejecución 
- No podemos definir la estructura de los datos 
- No tiene una buena experiencia de desarrollo* 
- No incorpora funciones de autocompletado y documentación de código* 



¿Solución? TypeScript

TypeScript is a strongly typed programming language that builds on JavaScript, giving you better tooling at any scale.



Tipado de variables

```
1 const fullName: string = "Harry Potter";
2 const age: number = 11;
3 const isMuggle: boolean = false;
4 const hogwartsHouse: string = "Gryffindor";
```



Tipado de funciones

```
1 function greetStudent(  
2   name: string, age: number,  
3   house: string  
4 ): string {  
5   return `Hello ${name}, welcome to ${house} at Hogwarts!`;  
6 }
```



Tipado de objetos

```
1 const student: {
2   name: string;
3   age: number;
4   house: string;
5 } = {
6   name: 'Harry Potter',
7   age: 11,
8   house: 'Gryffindor',
9 }
```



Interfaces

```
1 interface IStudent {  
2     name: string;  
3     age: number;  
4     house: string;  
5     greet(): string;  
6 }  
7
```



Talk is cheap

Show me the code



Proyecto



Proyecto

Estado actual

Partimos de una landing page estática sin ningún tipo de funcionalidad ni lógica.

Únicamente implementa el logo y el nombre en el HTML y una hoja de estilos en CSS

Harry Potter

Proyecto

Requisitos

- Crear tres páginas funcionales: home, equipos y contacto.
- Todo el contenido debe ser responsive y mobile first.
- En la home deberemos tener un “countdown” de los días hasta los próximos juegos.
- En la página de equipos debemos mostrar los escudos de las cuatro casas.
- Vía api, debemos obtener y mostrar los avatares de los personajes de esas casas.
- El formulario de contacto debe validar:
 - Campos obligatorios.
 - Email correctamente formateado.
- En la página de contacto debe haber un mapa con la localización de los juegos utilizando leaflet.





S O B R E L O S J U E G O S

La Copa Mundial de Quidditch se ha celebrado cada cuatro años desde 1473. La competencia se lleva a cabo entre los equipos de Quidditch que representan a los países de todo el mundo.

El número de países participantes varía de torneo en torneo. Cualquier país puede inscribir un equipo dentro de los doce meses posteriores a la final anterior.

P R Ó X I M A E D I C I Ó N

12

5

43

42

Talk is cheap

Show me the code



Descanso

Volvemos 21:08



Gracias



ngomez@codiara.com

<https://www.linkedin.com/in/nauelg/>

keep coding



www.keepcoding.io



cursos@keepcoding.io



(+34) 916 33 1779