



Configurando Vite, React, TailWindCss

Parte – 6 Creación del frontend

En este documento vamos a empezar a configurar nuestro frontend. Para ellos vamos a utilizar **Vite, React y Js**.

- **Vite:** viene del francés vite que significa rápido. Es una herramienta de compilación rápida y ágil. Nos permite utilizar plantillas ya diseñadas y trabajar con diferentes framework's o lenguajes. En este caso, vamos a utilizar Vite entre otras cosas, para generar la carpeta de desarrollo del frontend (cliente) a través de un simple comando.
- **React:** Es la biblioteca para interfaces de usuario web y nativas. Veremos que nos proporciona diferentes herramientas como, por ejemplo, para crear rutas en el frontend.
- **TailWindCss:** Otro framework como puede ser el Bootstrap. Nos proporciona una serie de estilos que podemos llamar a través de las clases desde nuestro frontend y nos permite hacer, en alguna medida, puntos de control.

Lo primero que vamos a hacer es lanzar el comando:

npm create vite

Este comando va a lanzar el wizard de Vite y nos va a ir haciendo una serie de preguntas. **Lo primero que os puede pedir es actualizar el vite, decirle que si que queréis.**

Nos va a preguntar lo siguiente:

- **Nombre del proyecto:** le vamos a llamar "cliente" o como vosotros queráis.
- **Nos pide seleccionar un Framework:** Seleccionamos "React".
- **Nos pide una variante:** Vamos a seleccionar "javascript".

Si os fijáis en la estructura de nuestro proyecto, nos debe haber creado una carpeta denominada "cliente" o el nombre que vosotros le hayáis dado. Vite nos ha creado dentro de la carpeta toda la estructura esencial para lanzar el servidor del cliente o del frontend.

Si os fijáis, el wizard de Vite nos devuelve el prompt pero antes nos dice que ejecutemos tres comandos:

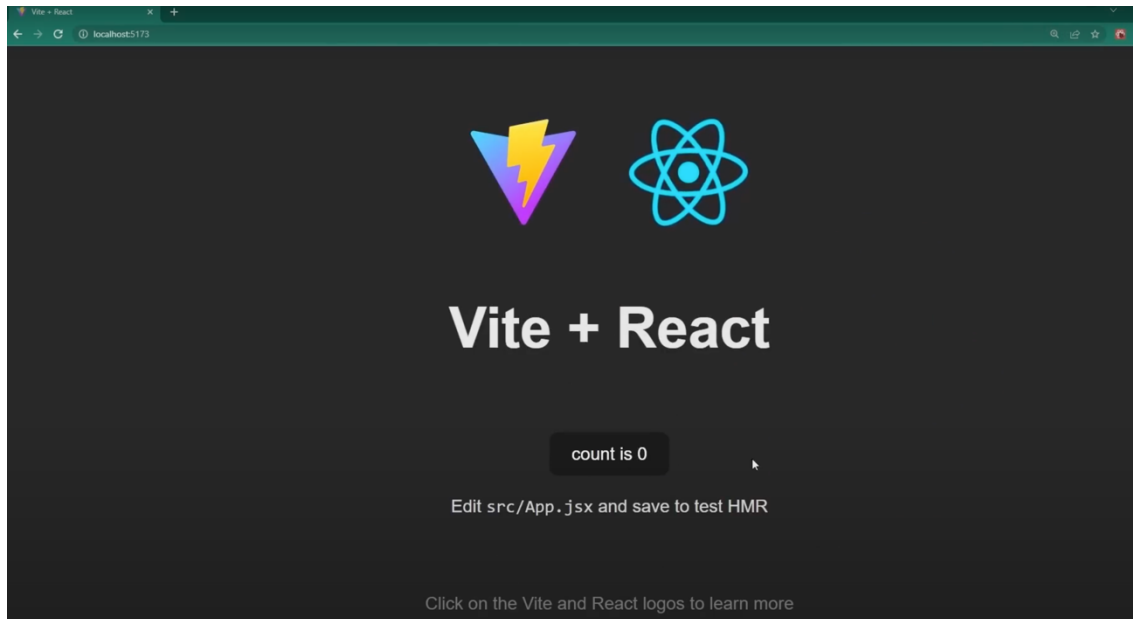


- **Primero:** Entrar dentro de la carpeta que nos ha creado Vite, **cd cliente**.
- **Segundo:** lanzar el comando **npm install**.
- **Tercero:** arrancamos el servidor del frontend a través del comando **npm run dev**.

Vite nos abre el puerto **5173** para poder trabajar con este entorno de trabajo.

Para comprobar que todo funciona bien, vamos a ir a nuestro navegador y vamos a

Localhost:5173



Nos da la bienvenida y nos dice que vamos a trabajar con Vite + React.

Si abrimos el fichero **package.json** y vamos haciendo un seguimiento de cómo funciona y va leyendo Vite los distintos ficheros, llegaremos a que los primeros cambios los haremos en el fichero **App.jsx**.

Fijaros que la extensión del fichero App no es js, ahora utilizamos jsx. Este tipo de datos pertenece a Ajax.

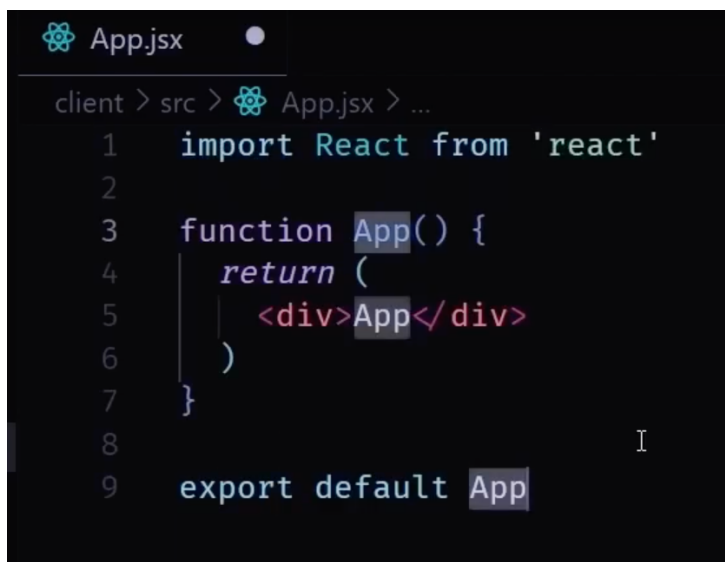
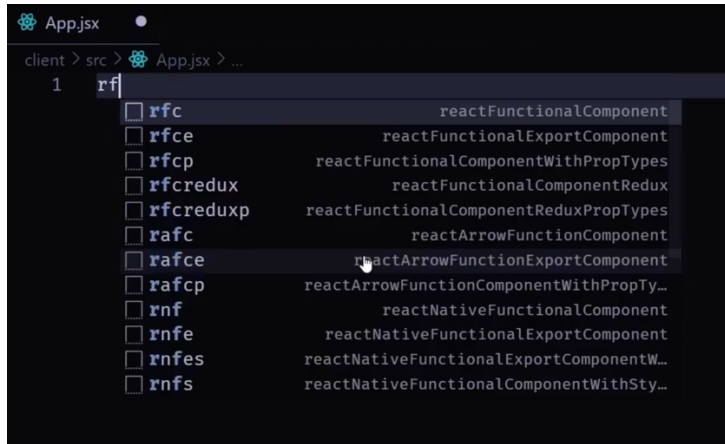
AJAX significa JavaScript asíncrono y XML (Asynchronous JavaScript and XML). Es un conjunto de técnicas de desarrollo web que permiten que las aplicaciones web funcionen de forma asíncrona, procesando cualquier solicitud al servidor en segundo plano

Vamos a editar el fichero App.jsx y vamos a borrar todas las líneas que tenemos dentro. De esta manera vamos a empezar a crear nuestra primera página de nuestro cliente.



Para trabajar de forma más rápida, podéis cargar la extensión **ES7+React/Redux/React-Native snippets**.

Esto me permite abrir un fichero en blanco, escribir la palabra **rfce** y cuando pulsamos intro, nos crea la estructura necesaria para empezar a trabajar.



En esta estructura que hemos creado, no nos hace falta importar desde 'react' pues iremos trabajando con diferentes ficheros. Eliminamos la primera fila y nos quedamos con la función App, que es la que va a cargar la web.

El fichero App.css no nos hace falta, porque vamos a utilizar tailwindcss (similar a Bootstrap). No lo vamos a utilizar en ningún otro documento que no fuera App.jsx y ya no lo referenciamos desde el mismo. Se puede borrar.

El fichero index.css, si que lo vamos a utilizar aunque toda las definiciones que tenemos de estilos en su interior, no nos hace falta. Vamos a borrar todo lo que está en su interior y lo utilizaremos en el momento de empezar a trabajar con tailwindcss.



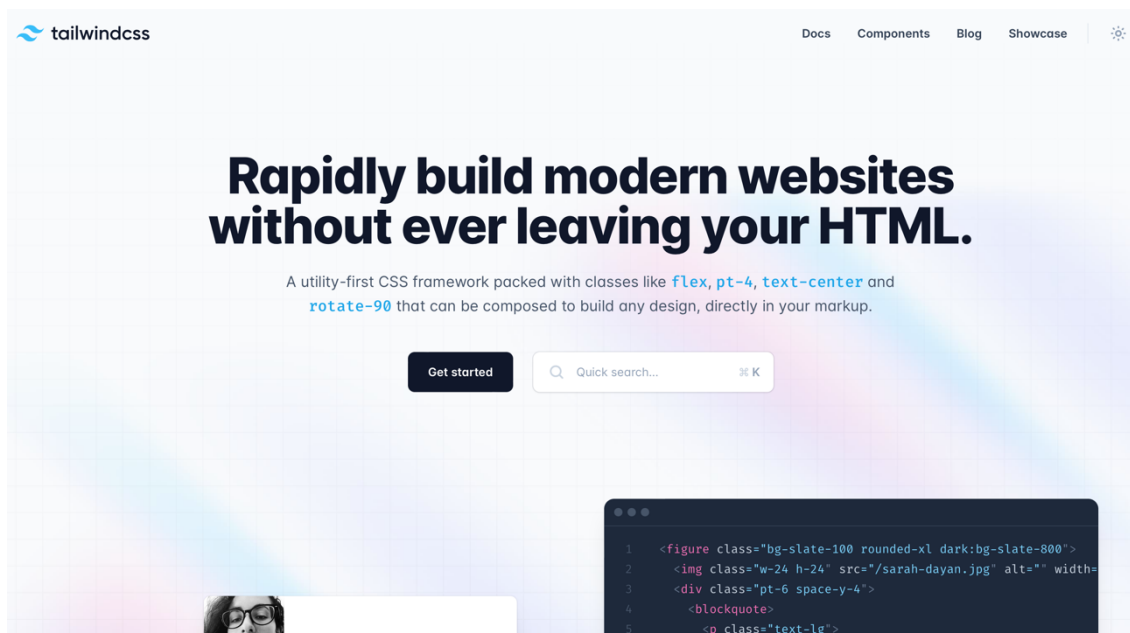
Cofinanciado por
la Unión Europea



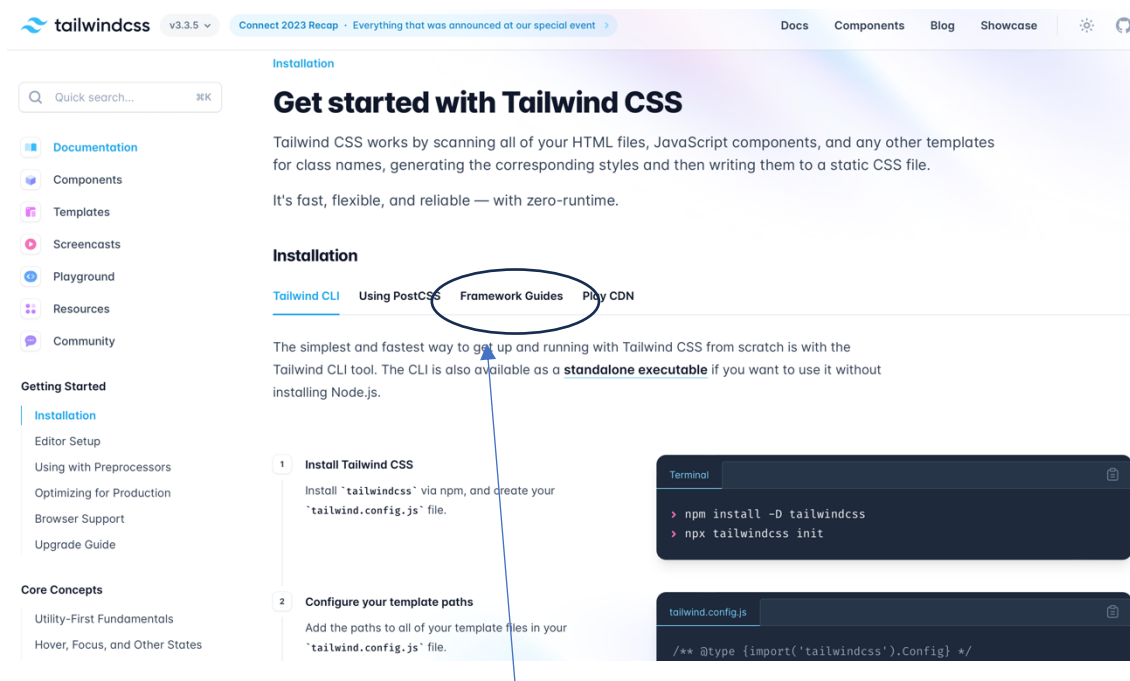
Fondos Europeos



A partir de este punto vamos a empezar a crear nuestra interface de usuario. Para ello, vamos a configurar el tailwindcss. Vamos a ir a su sitio web en el link **tailwindcss.com**



Pulsamos sobre el botón “Get started”



Seleccionamos la opción de Framework Guides



Cofinanciado por
la Unión Europea



Fondos Europeos



tailwindcss

v3.3.5

Connect 2023 Recap · Everything that was announced at our special event

Docs

Components

Blog

Showcase



Quick search...

Documentation

Components

Templates

Screencasts

Playground

Resources

Community

Getting Started

Installation

Editor Setup

Using with Preprocessors

Optimizing for Production

Browser Support

Upgrade Guide

Core Concepts

Utility-First Fundamentals

Hover, Focus, and Other States

Installation

Get started with Tailwind CSS

Tailwind CSS works by scanning all of your HTML files, JavaScript components, and any other templates for class names, generating the corresponding styles and then writing them to a static CSS file.

It's fast, flexible, and reliable — with zero-runtime.

Installation

Tailwind CLI

Using PostCSS

Framework Guides

Play CDN

Framework-specific guides that cover our recommended approach to installing Tailwind CSS in a number of popular environments.



Next.js

Full-featured React framework with great developer experience.



Laravel

PHP web application framework with expressive, elegant syntax.



Vite

Fast and modern development server and build tool.



Nuxt

Intuitive Vue framework for building universal applications.



Gatsby

Framework for building static sites with React and GraphQL.



SolidJS

A tool for building simple, performant, and reactive user interfaces.

Por último entramos dentro de Vite. Aquí nos encontraremos los pasos necesarios para poder trabajar con tailwindcss en nuestro proyecto.

tailwindcss

v3.3.5

Connect 2023 Recap · Everything that was announced at our special event

Docs

Components

Blog

Showcase



Quick search...

Documentation

Components

Templates

Screencasts

Playground

Resources

Community

Getting Started

Installation

Editor Setup

Using with Preprocessors

Optimizing for Production

Browser Support

Upgrade Guide

Core Concepts

Utility-First Fundamentals

Hover, Focus, and Other States

Setting up Tailwind CSS in a Vite project.

Using React

Using Vue

Using Svelte

1 Create your project

Start by creating a new Vite project if you don't have one set up already. The most common approach is to use **Create Vite**.

Terminal

```
> npm create vite@latest my-project -- --template react
> cd my-project
```

2 Install Tailwind CSS

Install `'tailwindcss'` and its peer dependencies, then generate your `'tailwind.config.js'` and `'postcss.config.js'` files.

Terminal

```
> npm install -D tailwindcss postcss autoprefixer
> npx tailwindcss init -p
```

3 Configure your template paths

Add the paths to all of your template files in your `'tailwind.config.js'` file.

tailwind.config.js

```
/** @type {import('tailwindcss').Config} */
export default {
  content: [
    './index.html',
    './src/**/*.js',
    './src/**/*.ts',
    './src/**/*.jsx',
    './src/**/*.tsx',
  ],
  theme: {
    extend: {},
  },
}
```

El primer punto "Create your Project", ya lo hemos hecho al principio así que este no nos hace falta.

Directamente vamos a empezar en el segundo y terminar en el sexto.