Introducción a React y creando proyecto

En esta sección vamos a comenzar a enfocarnos en el desarrollo web con React desde cero, se procederá a crear un nuevo proyecto e imprimir **"Hola mundo"** en el navegador.

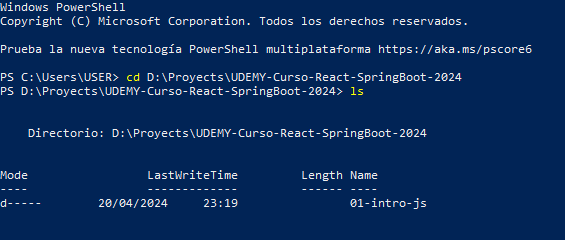
Crear un nuevo proyecto

Sigue estos pasos:

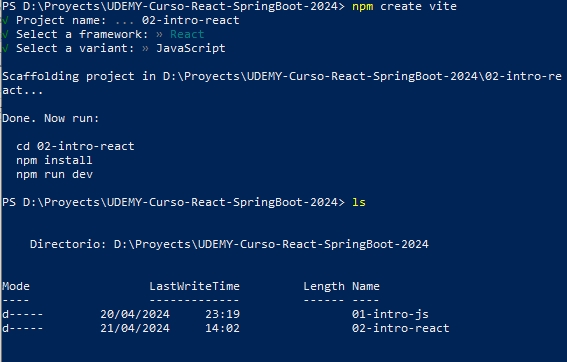
Comandos a introducir en una terminal PowerShell

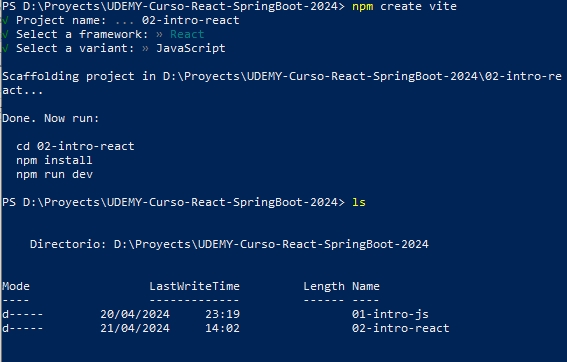
Cambia al directorio que contiene tus proyectos con el comando **cd ruta/a/la/carpeta/de/proyectos**. Asegúrate de reemplazar **ruta/a/la/carpeta/de/proyectos** con la ruta real a tu carpeta de proyectos.

Verifica si el proyecto anterior está presente utilizando el comando **ls**, el cual muestra la lista de archivos y carpetas en el directorio actual.

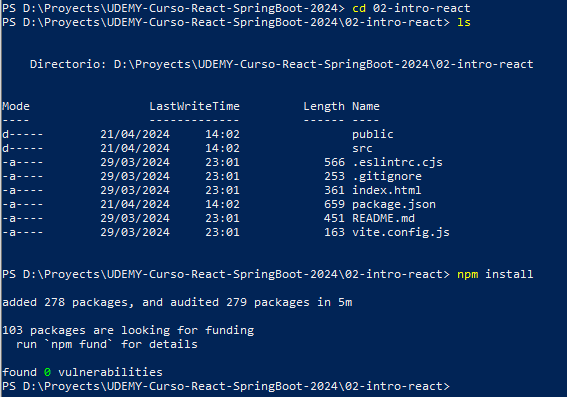


Crea un nuevo proyecto Vite con el comando **npm create vite**. Cuando se te solicita, ingresa el nombre del proyecto cómo **02-intro-react** en el campo **"Project name"**, luego selecciona **React** como el framework y **JavaScript** como la variante.





Cambia al directorio del nuevo proyecto creado con el comando **cd 02-intro-react**. Una vez dentro del proyecto, verifica la presencia del archivo **package.json** utilizando el comando **ls**. Si el archivo está presente, procede a instalar todas las dependencias enumeradas en **package.json** con el comando **npm install**. Esto generará la carpeta **node\_modules**.



Aún no cierres la terminal de Powershell.

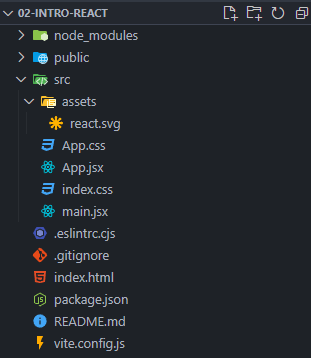
Cambiar de proyecto en Visual Studio Code

Si ya tienes abierto un proyecto anterior o cualquier otro proyecto en Visual Studio Code, sigue estos pasos para cambiar al nuevo proyecto **02-intro-react**:

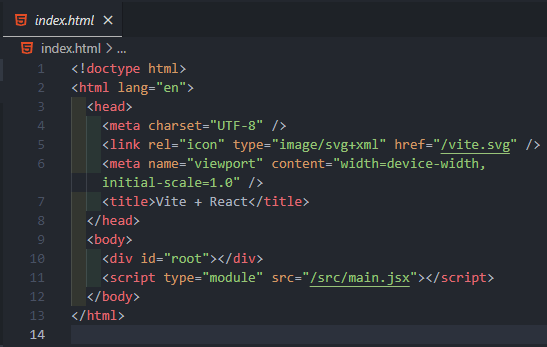
1. Cierra el proyecto actual haciendo clic en el menú **Archivo** y seleccionando **Cerrar carpeta**.
2. Luego, para abrir el nuevo proyecto, tienes dos opciones:
   1. Arrastra la carpeta **02-intro-react** y suéltala dentro de la ventana de Visual Studio Code.
   2. Haz clic en el menú **Archivo**, selecciona **Abrir carpeta** y busca la carpeta **02-intro-react**.

Estructura de archivos y carpetas

Todo nuevo proyecto generado, cuenta con la siguiente estructura de archivos y carpetas por defecto:



* **node\_modules/**: Contiene todas las dependencias del proyecto, incluyendo React, react-dom, y otras bibliotecas necesarias.
* **package.json**: Archivo de configuración que lista las dependencias del proyecto, como React y react-dom, así como otros scripts y metadatos del proyecto.
* **vite.config.js**: Archivo de configuración de Vite donde se definen las opciones y ajustes específicos para el entorno de desarrollo y producción.
* **src/**: Directorio principal que contiene todo el código fuente del proyecto:
  + **main.jsx**: Archivo principal de JavaScript que sirve como punto de entrada de la aplicación React.
  + **Archivos css**: Contienen las hojas de estilos (CSS) de la aplicación.
* **public/**: Directorio donde se almacenan los archivos estáticos accesibles públicamente, como imágenes y otros recursos.
* **index.html**: Archivo principal HTML de la aplicación. Contiene una etiqueta **<div>** con el identificador **root**, que es el punto de montaje para la aplicación React. Incluye un script de tipo **module** que importa y ejecuta el archivo **main.jsx**.



Ten en cuenta que JSX, o JavaScript XML, es una extensión de sintaxis para JavaScript que permite escribir código con una sintaxis similar a HTML/XML. En React, las etiquetas JSX se compilan a llamadas JavaScript puras para crear elementos React en el DOM. Esto proporciona una capa de abstracción que facilita la creación y manipulación de componentes de interfaz de usuario.

Dependencias React y React-Dom

Las dependencias principales del proyecto, **react** y **react-dom**, incluyen un conjunto de dependencias asociadas e independientes. Además, el archivo **package.json** distingue entre dependencias de producción (**dependencies**) y dependencias de desarrollo (**devDependencies**). Cuando construyes el proyecto para producción con Vite, solo las dependencias listadas en dependencies se incluirán en el bundle o proyecto final. Esto resulta en un proyecto más ligero y optimizado, ya que las dependencias de desarrollo no se incluirán.

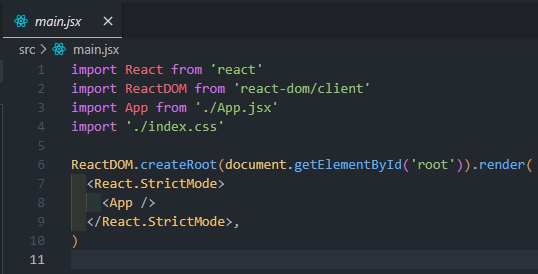
Además, Vite se encarga de compilar todo el código moderno de JavaScript (ES6+) a JavaScript nativo que es compatible con los navegadores.



Renderizado de componentes

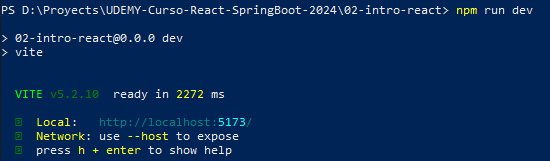
En el archivo **main.jsx**, que actúa como el punto de entrada principal de la aplicación, se realizan los siguientes pasos:

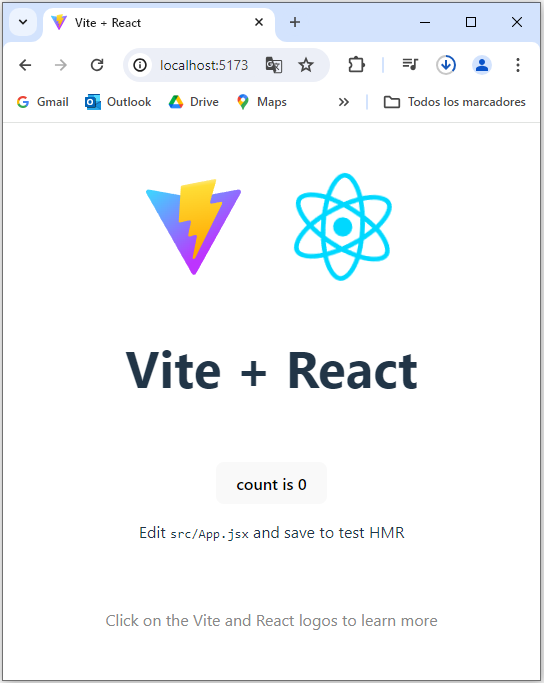
1. Se importan las bibliotecas necesarias:
   1. **React** desde el paquete **"react"**: es la biblioteca principal de React.
   2. **ReactDOM** desde el paquete **"react-dom"**: se utiliza para renderizar componentes React en el navegador en aplicaciones web. Para aplicaciones móviles, se usaría ReactNative.
2. Se importa el componente **App** desde el archivo **App.jsx**, que contiene el código JSX y JavaScript del componente.
3. El componente **App** se utiliza como componente principal y se renderiza dentro del elemento con el identificador **root**, que se encuentra en el archivo **index.html**. Todos los componentes de la aplicación se incluyen dentro de este elemento **<div>**.
4. **<React.StrictMode>** es una característica de React que ayuda a detectar prácticas desaconsejadas en la fase de desarrollo y marca warnings en la consola. Se puede omitir en la versión final de producción si se desea.



Ejecución del proyecto

Por defecto, el archivo **App.jsx** contiene el código fuente de una aplicación de tipo contador. Para ejecutar el proyecto, abre una nueva terminal en Visual Studio Code haciendo clic en el menú **Terminal**, opción **New Terminal**, escribe **npm run dev** y accede a la URL [**http://localhost:5173/**](http://localhost:5173/). También puedes ejecutar ese comando desde la terminal de PowerShell si estás ubicado en la carpeta **02-intro-react**.

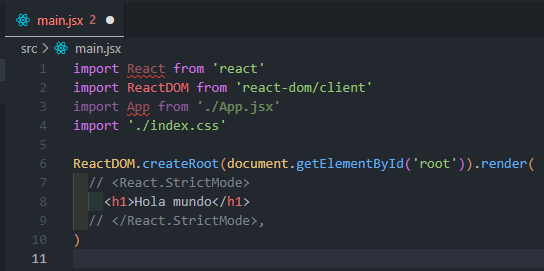




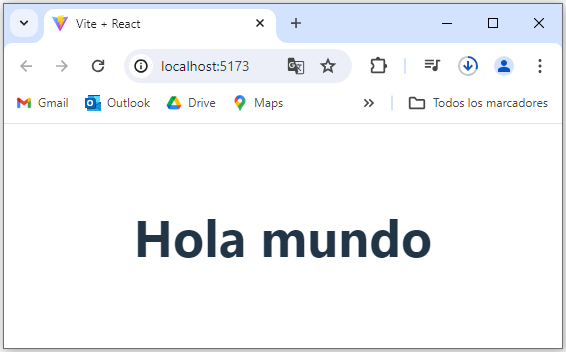
Hola mundo en React

Para mostrar un **"Hola Mundo"** en el navegador con React, sigue los siguientes pasos:

1. Abre el archivo **main.jsx**.
2. Cambia el elemento **<App />** (define el componente **App**) por **<h1>Hola Mundo</h1>**.
3. Comenta la línea **<React.StrictMode>** ya que no se utilizara en esta implementación.

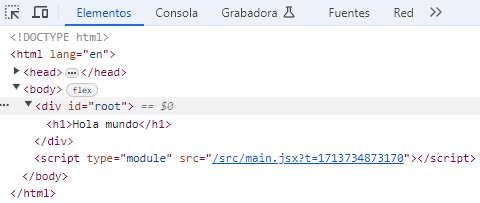


Después de hacer estos cambios, el contenido renderizado en el navegador mostrará **"Hola Mundo"** en lugar del componente **<App />**.



**Descripción detallada:**

* **ReactDOM.createRoot**: Esta función de React establece el punto de inicio o el componente principal en la página **index.html**. Localiza el **<div>** con el identificador **root** y, mediante el método **render**, comienza a anidar los componentes.
* **Estilo**: El diseño centrado del **"Hola Mundo"** se debe a las hojas de estilos que se incluyen en **index.css**.
* **Estructura del DOM**: Revisando el código fuente de la página, dentro del **<div>** con el ID **root**, se encuentran las etiquetas **<h1>**.



* **Funcionamiento de React**: React trabaja principalmente manipulando el DOM del documento HTML. Crea las etiquetas y va anidando y componiendo la aplicación con diferentes componentes reutilizables.