

NIVEL 1 REACT

Parte A: Crear y ejecutar el proyecto

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\juanf\OneDrive\Escritorio\nivel1_react> npm create vite@5.1.0 my-react-app -- --template react
Need to install the following packages:
create-vite@5.1.0
Ok to proceed? (y) y

> npx
> cva my-react-app react

↳ Select a framework: » React
↳ Select a variant: » TypeScript

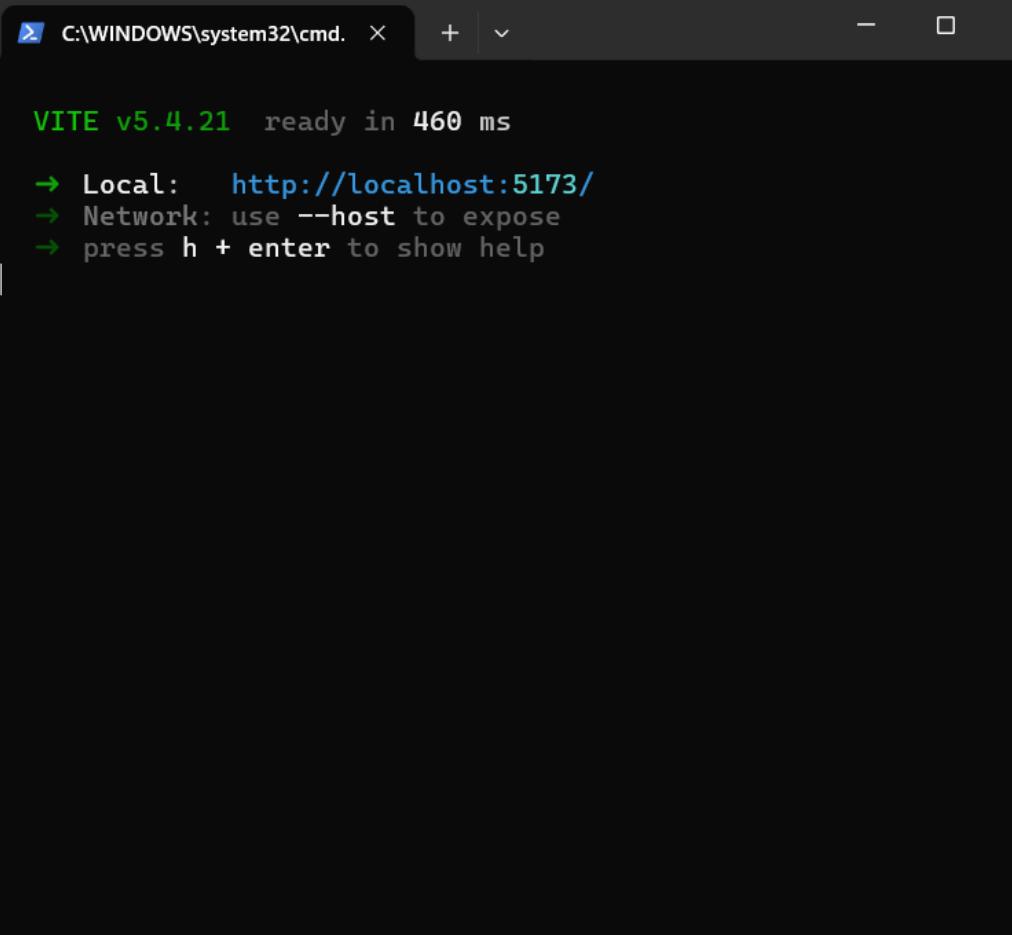
Scaffolding project in C:\Users\juanf\OneDrive\Escritorio\nivel1_react\my-react-app...

Done. Now run:

cd my-react-app
npm install
npm run dev

PS C:\Users\juanf\OneDrive\Escritorio\nivel1_react> |
```

En esta captura podemos observar como, ya dentro de nuestra carpeta creada "nivel1_react" y en ella en la terminal, ejecutamos los comandos proporcionados por la profesora para inicializar nuestro proyecto react



```
VITE v5.4.21 ready in 460 ms

→ Local: http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

En esta captura, observamos como después de poner todos los comandos, tenemos lanzado nuestro proyecto en el puerto 5173 y está disponible. Para acceder a este, debemos darle Control + Click encima o poniendo el http que nos aparece en nuestro buscador Chrome.

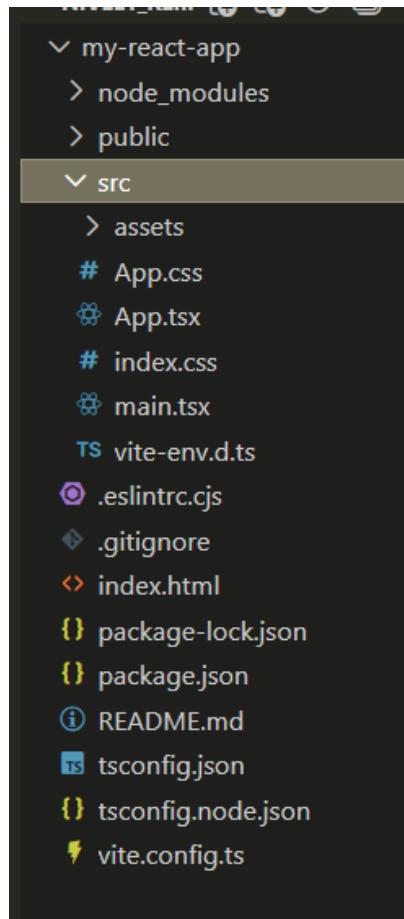
Una vez hecho la parte A, podemos pasar a la parte B del proyecto.

Parte B: Explorar la estructura del proyecto

- **vite.config.js**: Sirve como centro de control para la personalización y extensión del comportamiento de Vite en nuestro proyecto
- **package.json** y **package-lock.json**: Package.json define nuestro proyecto(dependencias en rangos de versión, scripts...), mientras que package-lock.json “bloquea” las versiones exactas de esas dependencias y subdependencias para garantizar instalaciones idénticas predecibles en cualquier máquina.

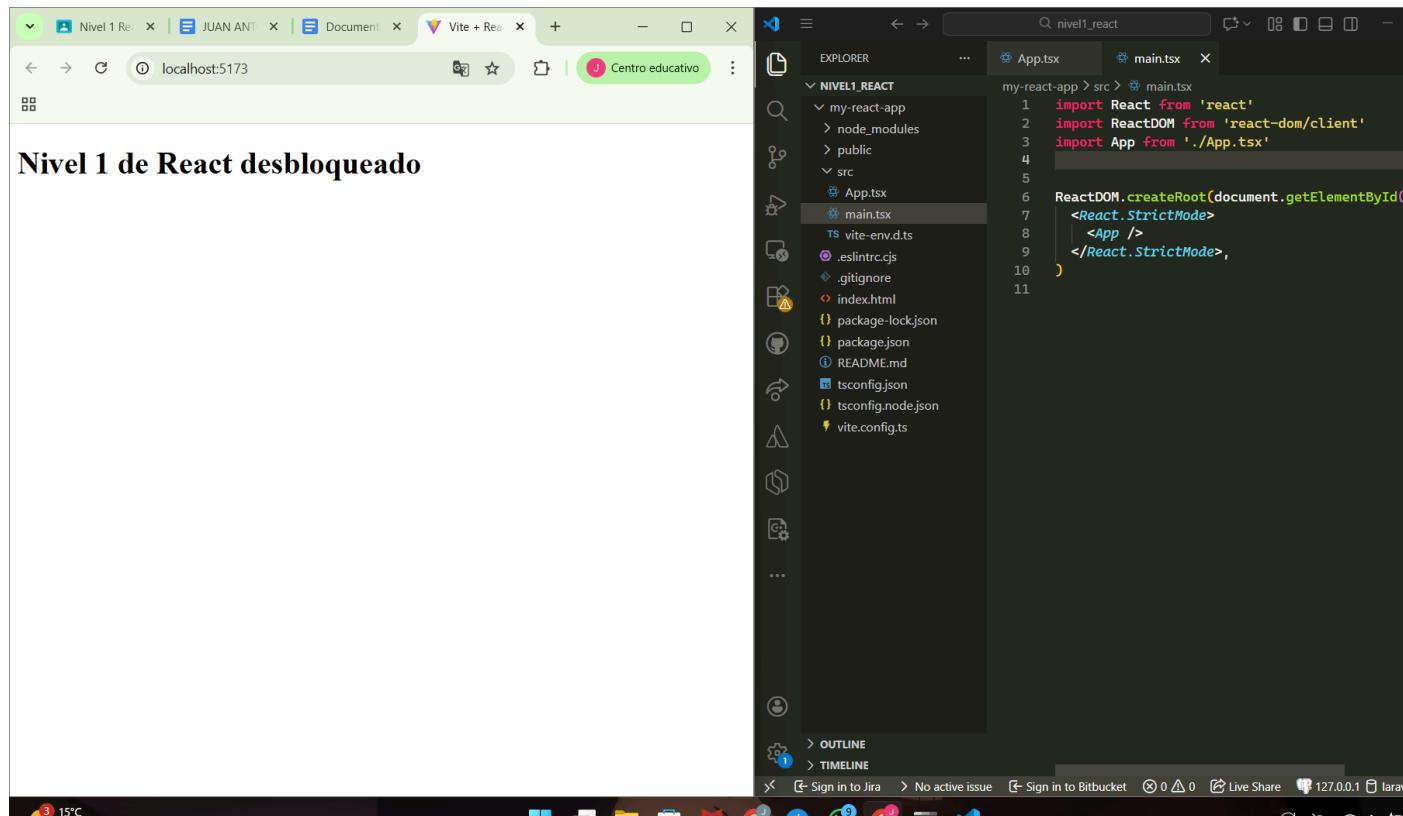
- **eslintrc.cjs**: Define qué reglas de estilo y calidad se deben aplicar, qué errores detectar y cómo formatear el código. Ubicado en la raíz del proyecto. Ubicado en la raíz del proyecto.
- **index.html**: Sirve como archivo principal o página de inicio por defecto de un sitio web. Ubicado en la raíz del proyecto.
- **src/main.jsx**: Es el punto de entrada principal de una aplicación React, un archivo fundamental que inicia y renderiza el componente raíz dentro del HTML.
- **src/App.jsx**: Es el componente principal y punto de entrada de una aplicación React, ubicado en la carpeta src (código fuente).
- **src/App.css, src/index.css, y src/assets/**: son archivos CSS para estilos en aplicaciones web (en especial React) donde App.css estiliza el componente principal, index maneja los estilos globales y assets es una carpeta dentro de src para recursos como imágenes, fuentes y archivos multimedia.

Aquí podemos observar la ubicación de las carpetas mencionadas anteriormente:



NOTA: En nuestro proyecto aparecerá “.tsx” y “.ts” ya que con las prisas hice el proyecto en TypeScript, no en JavaScript. Igualmente no es un impedimento para hacer que el proyecto funcione. Para el próximo sí que pondré JavaScript.

Parte C: Modificar app y limpiar archivos



En esta captura de pantalla podemos observar dos cosas:

- En la parte derecha (VS Code): Observamos la nueva estructura de nuestro proyecto React con sus modificaciones dichas por la profesora consistente en limpiar nuestro proyecto de archivos innecesarios.
- En la parte izquierda (Chrome): Estamos en la url de nuestro proyecto con sus modificaciones realizadas y mostrando cómo han funcionado, ya que modificando el src/App.tsx con el código proporcionado, se nos actualiza.

NOTA: Es necesario irnos a main.tsx y eliminar una línea (4) que nos hace el import del .css. Con esta modificación, solucioné el error y aseguré su correcto funcionamiento.

Preguntas extras PDF

1. ¿Qué hace ReactDOM.createRoot(...) en src/main.jsx?

ReactDOM.createRoot(...) crea el punto raíz donde React controla y renderiza toda la app dentro del DOM.

2. ¿Dónde se encuentra el elemento con id="root" y para qué sirve?

El elemento **id="root"** está en index.html y sirve como contenedor principal de la aplicación React.

3. ¿Qué comando utilizas para iniciar el servidor de desarrollo y qué URL local te muestra?

El comando es rpm run dev y muestra normalmente <http://localhost:5173/>.

4. ¿Qué cambiaste en App.jsx y por qué ya no hacen falta App.css/index.css/assets?

En **App.jsx** usé estilos inline/JSX simple, por eso ya no hacen falta App.css, index.css ni assets.