**React**

**Estructura de directorios de Vite.**

Documentación: <https://es.reactjs.org/>

<https://es.reactjs.org/docs/getting-started.html>

<https://vitejs.dev/>

<https://web.dev/learn/pwa/>

<https://developers.google.com/search/docs/advanced/robots/intro?hl=es&visit_id=638003729355730019-2382005997&rd=1>

Al momento de crear una app con npm Vite, en la estructura de directorios de Vite vamos a explicar ciertas diferencias que tiene contra CRA.

1. El archivo **“package-lock.json”:** Tiene el mismo objetivo tanto en Vite como CRA, pero aquí vamos a notar que ya hay varias diferencias marcadas, en la parte de las dependencias de Vite este tiene menos dependencias. CRA instala muchas cosas para que no tengamos que configurarlas a diferencia de Vite que cuando las necesitemos, vamos a tener que instalarlas y configurarlas por nuestro lado **esto puede ser bueno o puede ser malo.**

Algo que tiene bueno Vite es como hace el Hot Module Replacement (Forma de cambiar modulos en caliente) hace el cambio tan rápido que pareciera que nunca hizo ningún cambio, al momento que nosotros guardamos el cambio, lo despliega rápidamente sin importar que tan grande sea la aplicación. Al que mencionar del **package-lock.json** al momento de usar React Native no tenemos el react-dom.

1. El archivo **“index.html”:** Este archivo con Vite se encuentra en la carpeta raíz a diferencia con CRA que se encuentra dentro de public y el contenido es muy diferente ya que en Vite no se agrega el **manifest.json** es el que informa al navegador sobre la Aplicación Web Progresiva y cómo debe proceder cuando se instala en el escritorio o en el dispositivo móvil del usuario. Esto es una de las cosas que a la gente le disgusta de Vite que es mas crudo (tiene menos configuraciones automáticas).
2. El archivo **“vite.config.js”:** Este archivo nos va a servir a nosotros si queremos hacer configuraciones propias de Vite y esto no tiene nada que ver con React. Esto es una forma que tiene Vite para trabajar, ejemplo:

// https://vitejs.dev/config/

export default defineConfig({

  plugins: [react()]

})

Al tener ahí plugins de React, en pocas palabras pone a React de manera global en toda la aplicación, en todos los componentes y no tengamos que estar importando React siempre.