# React

## Para crear la aplicación con React ejecutamos el siguiente comando:

* npm create [vite@5.2.3](mailto:vite@5.2.3)

Que nos solicitará información para crear la estructura del proyecto.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

-node\_modules 🡪 Contiene todas las dependencias de nuestro proyecto, dentro está React, TypeScript y ReactDOM. (No tendríamos que tocar nada del directorio.)

-public 🡪 Mantiene todos los archivos que tienen que ser públicos y que cualquier usuario pudiera acceder. (Imágenes del proyecto, videos o documentos que nosotros indiquemos)

-src 🡪 Archivos del código fuente, aquí pasaremos el resto del tiempo!!.

-eslintrc.cjs 🡪 No lo vamos a tener que tocar, es usado para desarrollo.

-gitignore 🡪 Para trabajar con Git

-index.html 🡪Index principal del proyecto

-package-log.json 🡪 Guarda todo lo generado en package.json

- package.json 🡪 Contiene todas las dependencias del proyecto y algunos Script ejecutables.

Los devDependencies serán usadas mientras estemos programando, ya que en el producto final no se tendrían que ver.

-README.md 🡪Es donde se indican las instrucciones para los desarrolladores de como montar nuestro proyecto.

-tsconfig.json / tsconfig.node.json 🡪 Sirven para configurar TypeScript y no los deberíamos de tocar prácticamente nunca.

-vite.config.ts 🡪 Se tocará más adelante.

React no es el encargado de renderizar todo el contenido, para ello se emplea React-dom.

Para construir aplicaciones en la WEB se emplea react-dom.

Para móviles react-native.

Y para aplicaciones de escritorio podríamos usar una nueva biblioteca llamada react-native for Windows or MacOs, pero por el momento está en beta.

Bibliotecas populares de React

BootStrap 🡪 ReactBootStrap

(BootStrap) Nos entrega un listado de componentes que podemos usar asignándoles clases de CSS.

(ReactBootStrap) No se tienen que crear componentes y asignarles CSS, importamos directamente el componente.

Tailwind 🡪 La diferencia con BootStrap es que no nos devuelve componentes, nos dará unas Utility Clases, asignando estilo, por ejemplo:

className=” pd-8”

No se escribe CSS como se puede ver, pero nos tendríamos que aprender todos los nombres de clases.

Para solución de esto se creó la biblioteca Deisy UI.

Deisy UI 🡪 Usa Tailwind por detrás para crear componentes.

Bulma CSS 🡪 Nos entrega componentes y les asignamos estilo con CSS, es parecida a BootStrap.

React Bulma 🡪 Importar componentes como con ReactBootStrap o Deisy UI y ya vienen con los estilos asignados.

Chakra UI 🡪 Se creó específicamente para ser usada con React, también que todas las anteriores entrega componentes.

Este es diferente (Visibilidad reducida,ciegas)

La forma de trabajar con las bibliotecas va a ser más o menos parecida, habrá que acceder a la documentación, ver los componentes disponibles y copiar el código y pegarlo dentro del proyecto y modificarlo.