

Recordá poner a grabar
la clase



Hedy

Calilegua



React

React es una librería que permite construir interfaces de usuario a partir de piezas individuales llamadas componentes, creando tus propios componentes y combinándolos para crear pantallas completas, páginas y aplicaciones.

Los 3 conceptos más importantes de React son:

(En orden de importancia)

- **Componentes**
- **Estado**
- **Props**

Limitaciones de React

React tiene limitaciones que generalmente son solucionadas por los desarrolladores al instalar y configurar manualmente paquetes de terceros.

Estas son algunas de las cosas que no podemos hacer únicamente con React:

- Routing
- Server Side Rendering
- Data Fetching
- Styling
- Optimizaciones
- Typescript (depende de qué bundler se utiliza para crear la aplicación web, por ejemplo: Vite, Create React App, entre otros)

Nota importante

Estas no son las únicas librerías o paquetes que pueden utilizarse para solucionar los problemas mencionados. Son solamente ejemplos específicos.

Routing

React Router: [Home v6.22.1 | React Router](#)

React Router es una biblioteca de enrutamiento para React que permite manejar la navegación y la URL en una aplicación de una sola página (SPA) de manera declarativa.

Proporciona un conjunto de componentes de React que permiten definir rutas en la aplicación y renderizar componentes diferentes según la ruta actual en la URL del navegador.

SSR (Server Side Rendering)

NextJS: [Rendering: Server-side Rendering \(SSR\) | Next.js \(nextjs.org\)](https://nextjs.org/docs/basic-features/server-side-rendering)

Hay dos entornos donde las aplicaciones web pueden renderizarse: el cliente y el servidor. En el cliente, el código de la aplicación se ejecuta en el navegador web del usuario final, mientras que en el servidor, el código se ejecuta antes de que se envíe la respuesta al navegador

Data Fetching

SWR: [Biblioteca React Hooks para la obtención de datos – SWR](#)

(Stale-While-Revalidate) es una biblioteca de obtención de datos basada en hooks que tiene como objetivo proporcionar una mejor experiencia de usuario al mejorar la capacidad de respuesta y el rendimiento percibido de las aplicaciones web.

Styling

NextJS: [Building Your Application: Styling | Next.js \(nextjs.org\)](https://nextjs.org/docs/app/building-your-application/styling)

- **CSS Global:** Simple de usar y familiar para aquellos con experiencia en CSS tradicional, pero puede llevar a paquetes de CSS más grandes y dificultad para administrar estilos a medida que la aplicación crece.
- **Módulos CSS:** Crea clases de CSS con ámbito local para evitar conflictos de nombres y mejorar la mantenibilidad.
- **Tailwind CSS:** Un marco de trabajo CSS centrado en utilidades que permite diseños personalizados rápidos al componer clases de utilidad.
- **Sass:** Un preprocesador CSS popular que extiende CSS con características como variables, reglas anidadas y mixins.
- **CSS-in-JS:** Incrusta CSS directamente en tus componentes de JavaScript, permitiendo estilos dinámicos y con ámbito.

Optimizaciones

SWR: [Biblioteca React Hooks para la obtención de datos – SWR](#)

Los componentes integrados abstraen la complejidad de implementar optimizaciones comunes de la interfaz de usuario. Estos componentes son:

Imágenes: Construidas sobre el elemento nativo ``. El componente de Imagen optimiza las imágenes para mejorar el rendimiento mediante la carga diferida y el redimensionamiento automático de imágenes según el tamaño del dispositivo.

Enlaces: Construidos sobre las etiquetas nativas `<a>`. El componente de Enlace prefetches (precarga) páginas en segundo plano para transiciones de página más rápidas y suaves.

Scripts: Construidos sobre las etiquetas nativas `<script>`. El componente de Script te brinda control sobre la carga y ejecución de scripts de terceros.

Typescript

NextJS: [Configuring: TypeScript | Next.js \(nextjs.org\)](https://nextjs.org/docs/basic-features/typescript)

Soporte mejorado para typescript

- instalación automática
- configuración automática
- Typescript Plugin
- Type Checker

Hedy