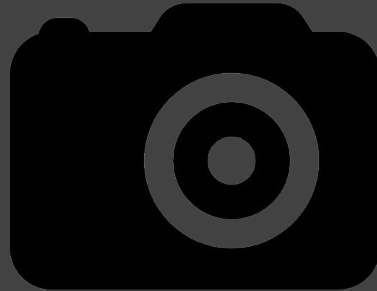


Recordá poner a grabar
la clase



Hedy

Calilegua



Next.js

Next.js es un framework de React que nos permite construir sitios web y aplicaciones web Full-Stack, amigables con el SEO y extremadamente orientadas al usuario, utilizando el framework de React.

Next.js vs React

React es una biblioteca para construir interfaces de usuario, Next.js es un framework que extiende las capacidades de React al proporcionar herramientas adicionales para el renderizado del lado del servidor, el enrutamiento, la generación de sitios estáticos, optimizaciones de rendimiento, entre otras.

Nombres de archivos reservados

NextJS nos provee nombres de archivos reservados cuando los creamos dentro del directorio app (y solamente dentro de app, es decir, en cualquier otro lado estos archivos no se van a tratar de forma especial)

Algunos ejemplos comunes son:

- **Page.js**
- **Layout.js**
- **error.js**

Y muchos más que pueden ser encontrados en la documentación oficial

[API Reference: File Conventions | Next.js \(nextjs.org\)](https://nextjs.org/docs/api-reference/file-conventions)

Routing entre paginas

Next.js no tiene una opinión predefinida sobre cómo organizar y colocar los archivos de un proyecto. Sin embargo, si se nos proporcionan dos formas de manejar las rutas de nuestro proyecto. El app router, y el directorio de páginas

En el directorio de la aplicación, la jerarquía de carpetas anidadas define la estructura de rutas y cada carpeta representa un segmento de ruta que se asigna a un segmento correspondiente en la URL.

Sin embargo, aunque la estructura de la ruta se define a través de carpetas, una ruta no es accesible públicamente hasta que se agregue un archivo `page.js` o `route.js` a un segmento de ruta.

[Routing: Project Organization | Next.js \(nextjs.org\)](#)

Componente `<Link>` vs `<a/>`

En NextJS hay dos formas de crear enlaces entre páginas (Componente Link y elemento a):

El componente `<Link>` se importa desde `next/link`.

Permite crear enlaces entre páginas de forma declarativa en la aplicación Next.js.

Ayuda a prevenir la recarga completa de la página al navegar entre diferentes rutas, ya que utiliza el enrutamiento del lado del cliente.

El elemento `<a>` de HTML se puede utilizar para crear enlaces de la manera tradicional.

Sin embargo, cuando se utiliza un elemento `<a>` común, se pierden algunas de las optimizaciones y funcionalidades proporcionadas por `<Link>`.

En conclusión: Utilizar Link cuando tenemos que cambiar entre rutas internas de nuestra aplicación y si queremos podemos utilizar el elemento a cuando queremos redireccionar al usuario a una ruta externa

Slugs (Routing dinámico)

Cuando se conocen los nombres exactos de los segmentos de antemano y queremos crear rutas a partir de datos dinámicos se pueden utilizar segmentos Dinámicos que se completan en el momento de la solicitud o se pre procesan en el momento de la construcción.

Un Segmento Dinámico puede ser creado al envolver un nombre de archivo o carpeta entre corchetes: `[nombreDelSegmento]`. Por ejemplo, `[id]` o `[slug]`.

Los Segmentos Dinámicos pueden ser accedidos desde el cliente Utilizando el hook `useRouter()` o el parámetro `params` en la declaración del componente

[Routing: Dynamic Routes | Next.js \(nextjs.org\)](https://nextjs.org/docs/routing/dynamic-routes)

Estilizado

Next.js admite diferentes formas de estilizar tu aplicación, incluyendo:

- CSS Global
- Modulos CSS
- Tailwind CSS
- Sass
- CSS-in-JS

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

Hedy