* npm create next-app@latest <nombre proyecto>

# /public/

Poner los archivos estáticos, como el /img/

# /pages

Aquí se van a ubicar las rutas (es como sveltkit),

Al nombrar rutas evitar usar el ‘-’

Si se crea un archivo cuyo nombre sea distinto des ‘index.jsx’, se considera una ruta a partir de esta carpeta /miruta

## /pages/MiRuta/index.jsx

Si se crea un carpeta y dentro un index.jsx entonces esto es una nueva ruta /Miruta donde se renderiza el contenido de index.jsx

export default function index(){

    return (

        <>

        Hello

        </>

    )

}

# Link

Import Link from ‘next/link’

<link href=’/ruta/mivista’ > Algo </>

# Image

Import Image from ‘next/image’;

<Image src=”..jpg” width={600} height={600} alt=”descripcion” />

La carpeta public se considera la raíz asi que no hay que incluirla

# Metadatos

Para el CEO

Import Head from “next/head”;

<Head>

<title> Mi titulo</>

<link rel=“icon” ../>

<meta …/>

</Head>

# Layout

Crear un componente (en cualquier parte, pero preferiblemente en su carpeta components) que haga de esta función,

Export default funtion MiLayout({children}){

Return (

…

{children}

)

}

# defaultProps

Para poner parámetros por defecto tambien se puede

Componente.defaultProps={

    title:"el titulo"

}

# Estilos en modo clase

/styles/miestilos.css

.container{

    max-width: 36rem;

    font-size: large;

    color: blue;

}

En el componente

import styles from '../styles/Layout.module.css'

<div className={styles.container}>

En los estilos no se pueden usar etiquetas html, solo clases

# getStaticProps

Supuestamente este metodo no se compila al frontend

export default function index({data}){

    return (

        <>

        Hello

        {

            data.map(({id,title,body}, index) => {

                return (

                    <div key={id}>

                        <h3>{id} - {title}</h3>

                        <p>{body}</p>

                    </div>)

        })

        }

        </>

    )

}

export async function getStaticProps() {

    try{

        const response = await fetch('https://jsonplaceholder.typicode.com/posts')

        const data = await response.json()

        return {

            props: { data },

            revalidate: 60 // 1 minute

            }

    }catch(error){

        console.error(error)

    }

}

Rutas dinámicas

/pages/ruta/[parametro].jsx

Parecido a sveltekit