



## ISWD813– Aplicaciones Web Avanzadas

FACULTAD DE INGENIERÍA DE SISTEMAS

### Desarrollo SPA con React



**DOCENTE:** Ing. Christian Suárez

**Grupo:** 2

**Estudiante:** Nicolás Omar Barragán Hidalgo

**Fecha de entrega:** 12-09/2023

## Conceptos básicos

---

Las aplicaciones de React se construyen a partir de componentes, los cuales son elementos de la interfaz de usuario (UI) que poseen su propia funcionalidad y aspecto. Un componente puede tener un tamaño tan reducido como el de un botón o ser tan extenso como una página completa. Entre las características principales de este marco de trabajo están las siguientes herramientas [1]:

### Componentes

Son la unidad fundamental de React. Son bloques de construcción reutilizables que encapsulan la lógica y la interfaz de usuario de una parte específica de una aplicación. Pueden ser tan simples como un botón o tan complejos como una página completa.

### Virtual DOM

El Virtual DOM es una representación virtual en memoria del DOM real de una página web. React utiliza esta representación para realizar un seguimiento de los elementos de la interfaz de usuario y sus estados.

### Estados y props

El estado es un objeto que almacena datos que pueden cambiar con el tiempo y que afectan la representación visual de un componente mientras que las props son datos que se pasan de un componente padre a un componente hijo.

### Renderización

React utiliza una función llamada `render()` para definir la estructura de la interfaz de usuario de un componente. Esta función devuelve un árbol de elementos React (usualmente JSX) que describe la interfaz.

## Características principales

---

### Ciclo de vida del componente

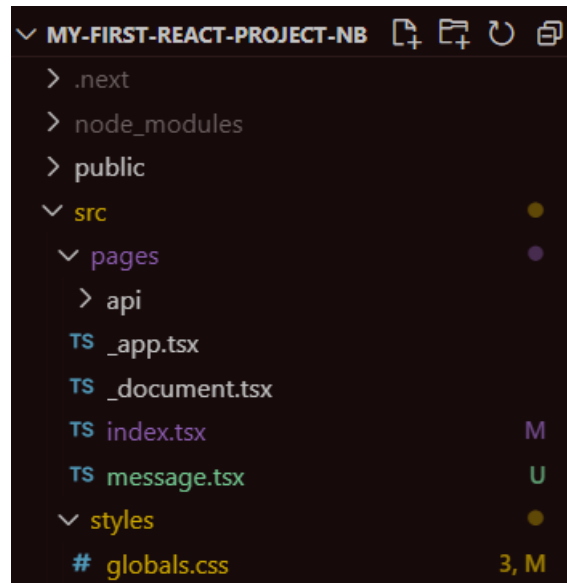
Los componentes en React pasan por un ciclo de vida que consta de diferentes etapas, cada una con métodos específicos que puedes usar para controlar el comportamiento del componente en esas etapas. Estas etapas, de manera general son [2]:

- Montaje
- Actualización
- Desmontaje
- Manejo de errores

### Unidireccionalidad de los datos

Los datos fluyen en una sola dirección, desde los componentes principales hacia los componentes secundarios. Esto facilita la gestión del estado de la aplicación y ayuda a prevenir errores de actualización no deseados.

## Estructura del proyecto



Según la estructura de la ilustración anterior lo más importante de destacar es en donde se crean los componentes. En la carpeta “src/pages” es en donde se crean los archivos que contendrán a los componentes como general se va a tener el archivo index.tsx como archivo principal. El lenguaje de programación es TypeScript pero al mismo tiempo se utiliza JSX

## JSX

También llamado JavaScript XML es una extensión de JavaScript que se utiliza en React para definir la estructura de la interfaz de usuario (UI) de una manera similar a HTML, pero dentro del código JavaScript. Algunos ejemplos de la sintaxis se pueden ver continuación:

- Para la creación de constantes:

```
const name = 'Nicolás Barragán'
```

- Para la creación de componentes:

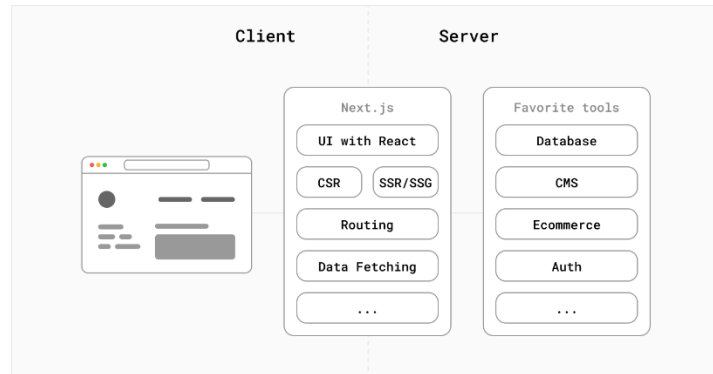
```
function Message() {  
  const name = 'Nicolás Barragán'
```

- Para enlazar datos a un template

```
return (  
  <main>  
    <h1>Hello World!</h1>  
    <h1>My name is {name} </h1>  
  </main>  
)  
}
```

## Arquitectura

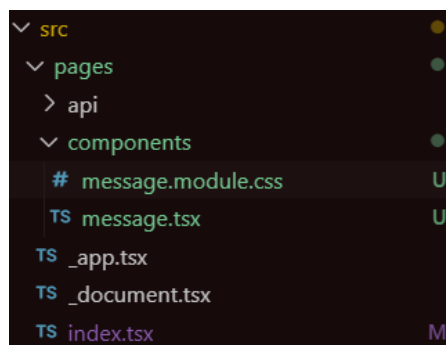
Un proyecto React trabaja a la par con Next.js, el cual es un marco de trabajo que proporciona los componentes básicos para construir aplicaciones web. Con "marco de trabajo", nos referimos a que Next.js se encarga de las herramientas y la configuración necesarias para React, además de ofrecer estructura adicional, características y optimizaciones para tu aplicación. Para una ejemplificación de como se estructura una aplicación React con Next.js se puede observar en la siguiente ilustración:



## Ejemplo

En este primer proyecto se mostrará las características más básicas de React. Para crear un proyecto React únicamente se necesita tener Node.js y utilizar el comando "npx create-next-app". Después de seguir las configuraciones se tendrá el proyecto funcional.

Primero se creará un componente "Message" que contendrá nuestro mensaje. Para esto se crea una carpeta de componentes y un archivo con la extensión. tsx. Adicionalmente se crea un modulo de CSS para utilizar clases de estilos. La estructura del proyecto es la siguiente:



Para crear un componente de acuerdo a JSX se tendrá lo siguiente:

```
function Message() {
  const name = 'Nicolás Barragán'

  return (
    <main className={styles.title}>
      <h1>Hello World!</h1>
      <h1>My name is {name}</h1>
    </main>
  )
}

export default Message;
```

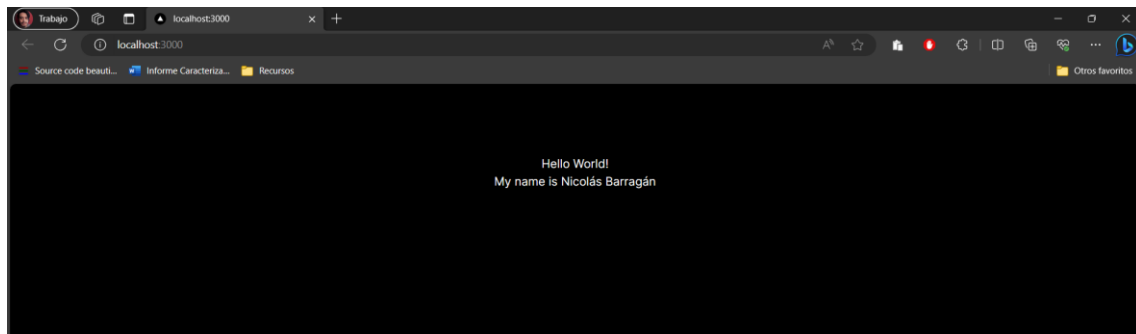
En el código anterior podemos ver la implementación de enlace de datos y el uso de módulos CSS. Los componentes siempre deben ser exportados para ser utilizados por otros componentes. En este caso se le hizo llamado en el componente principal “index.html” y es importante recalcar que siempre hay que importar los componentes a utilizar.

```
import Message from './components/message'

const inter = Inter({ subsets: ['latin'] })

export default function Home() {
  return (
    <main
      className={`flex min-h-screen flex-col items-center justify-between p-24 ${inter.className}`}
    >
      <div><Message></Message></div>
    </main>
  )
}
```

El resultado de lo anterior se puede observar en la siguiente ilustración:



## Anexos

---

Repositorio en Github

[NicolasBH9/my-first-react-project-nb: Desarrollo SPA con React \(github.com\)](https://github.com/NicolasBH9/my-first-react-project-nb)