

## Práctica

## Introducción

## Modalidad

Individual

La importancia de #Hacer con estilos en React

Autogestionada

Los estilos en un proyecto React son fundamentales para definir la apariencia y la experiencia visual de la aplicación, permitiendo que los componentes sean atractivos y fáciles de usar, esto se puede lograr mediante CSS, estilos en línea o librerías como Styled Components, React-Bootstrap, MUI entre otros.

Los PropTypes son una herramienta que ayuda a validar las propiedades pasadas a un componente, asegurando que reciban el tipo de datos correcto y mejorando la robustez del código. Esto facilita la detección de errores durante el desarrollo.

Las rutas permiten la navegación entre diferentes vistas o componentes, organizando la estructura de la aplicación y mejorando la experiencia del usuario al facilitar el acceso a diferentes secciones.

---

**Recuerda...**

Los estilos deben aplicarse correctamente para que los componentes se visualicen adecuadamente; sin un tamaño y diseño apropiados, la interfaz puede resultar confusa o poco atractiva. Además, al trabajar con rutas, es crucial gestionar adecuadamente los parámetros y la navegación para evitar errores y mejorar la experiencia del usuario.

Los estilos mejoran la apariencia visual de los componentes, mientras que los PropTypes validan las propiedades, asegurando la robustez del código. Las rutas organizan la navegación entre vistas, mejorando la experiencia del usuario. Juntos, estos elementos contribuyen a crear aplicaciones React más eficientes, mantenibles y agradables visualmente.

**Y..., ¿Qué esperamos que puedan practicar?**

Esperamos que, a través de los ejercicios sobre estilos y rutas en React, desarrolles una comprensión sólida de cómo aplicar estilos de manera efectiva para mejorar la estética de tus componentes. Además, es crucial que adquieras habilidades en la gestión de rutas para facilitar la navegación dentro de tu aplicación. Buscamos que experimentes con el renderizado condicional y la validación de propiedades con PropTypes. Al finalizar, deberías ser capaz de crear una interfaz interactiva y visualmente atractiva, optimizando la experiencia del usuario. ¡Estamos ansiosos por ver tus resultados!

**¡Comencemos! A continuación tendrán el detalla de cada una de las prácticas formativas autogestionadas.**

Práctica

#Hacer con estilos en React

Modalidad

Individual

Estilos

Autogestionada

- **Propósito:** Practicar la aplicación de estilos en React
- **Material de consulta:** Recupera los temas de la sección de "Estilos de React" sobre estilos CSS
- **¿Dudas, consultas?:** En el caso de que lo necesites y requieras orientaciones del/la docente en el foro de consultas, podrás dejar tus comentarios.

A continuación analiza las siguientes consignas y desarrolla el código necesario para completarlas:

### Propuesta A

A partir del siguiente componente StyleMe, agrega estilos en línea al elemento h1 para que tenga background "lightblue" y color "darkred":

JavaScript

Copiar código

```
import React from "react";
import ReactDOM from "react-dom";

const StyleMe = <h1>Please style me! I am so bland!</h1>;

ReactDOM.render(StyleMe, document.getElementById("root"));
```

### Propuesta B

Aplicar estilos en línea en React, puede resultar molesto si se desea usar más que solo unos pocos estilos.

Una alternativa más agradable es almacenar un objeto de estilo en una variable e inyectar esa variable en JSX.

- A partir del componente StyleMe de la actividad anterior, declara una nueva constante llamada styles y asigna los estilos background "lightblue" y color "darkred" al elemento h1

JavaScript

Copiar código

```
import React from "react";
import ReactDOM from "react-dom";

const StyleMe = <h1>Please style me! I am so bland!</h1>;

ReactDOM.render(StyleMe, document.getElementById("root"));
```



Práctica

#Hacer con estilos en React

Modalidad

Individual

Componente, estructura y ruta

Autogestionada

- **Propósito:** Practicar la confección de componente, estructura y ruta
- **Material de consulta:** Recupera los temas de la sección de "Estilos de React" sobre Rutas
- **¿Dudas, consultas?:** En el caso de que lo necesites y requieras orientaciones del/la docente en el foro de consultas, podrás dejar tus comentarios.

A continuación analiza las siguientes consignas y desarrolla el código necesario para completarlas:

#### Propuesta A

Crear un componente que permita cambiar el tema de una aplicación (por ejemplo, claro y oscuro).

Pista: puedes utilizar `useState` para manejar el estado del tema actual.

#### Propuesta B

Crear una ruta que requiera que el usuario inicie sesión antes de poder acceder a ella.

Para esto, debes agregar la sintaxis de rutas protegidas y una lógica de autenticación en la aplicación.

Práctica

#Hacer con estilos en React

Modalidad

Individual

Crear aplicaciones

Autogestionada

- **Propósito:** Practicar la creación de aplicaciones React
- **Material de consulta:** Recupera los temas de la sección de "Estilos React" sobre estilos CSS, React-Bootstrap y rutas
- **¿Dudas, consultas?:** En el caso de que lo necesites y requieras orientaciones del/la docente en el foro de consultas, podrás dejar tus comentarios.

A continuación analiza las siguientes consignas y desarrolla el código necesario para completarlas:

#### Propuesta A

Crea una aplicación que muestre una lista de destinos turísticos.

- Cada destino debe estar representado por una imagen y un nombre.
- Al hacer clic en uno de los destinos, la aplicación debe navegar a una ruta específica que muestre información detallada sobre el destino.
- Valida las props de los componentes con PropTypes.
- Utiliza los componentes de React-Bootstrap.
- En caso de que se introduzca una ruta con un destino turístico incorrecto, se debe mostrar un componente NotFound que muestre un mensaje de error y un botón para volver a inicio.

#### Propuesta B

Crea una aplicación que muestre una lista de restaurantes.

- Cada restaurante debe estar representado por una imagen y un nombre.
- Al hacer clic en uno de los restaurantes, la aplicación debe navegar a una ruta específica que muestre información detallada sobre el restaurante, incluyendo el menú y la ubicación.
- En caso de que se introduzca una ruta con un restaurante incorrecto, se debe mostrar un componente NotFound que muestre un mensaje de error y un botón para volver a inicio.