«Talento Tech»

React JS

Clase 13

Buenos Aires Ciudad

Clase N° 13 | Estilización con Bootstrap o styled-components

Índice:

- Introducción a Bootstrap o styled-components para estilizar componentes.
- Creación de un diseño básico y responsive.
- Aplicación de estilos en componentes (botones, formularios, productos).

Objetivos de la Clase:

- Optimizar el diseño y la responsividad de la aplicación utilizando **Bootstrap** y **styled-components**.
- Mejorar la experiencia de usuario con **React Icons** y **React Toastify**.
- Aplicar ajustes visuales y de interacción para hacer la interfaz más atractiva y funcional.
- Implementar mejoras básicas de **SEO** con **React Helmet** para optimizar la visibilidad en buscadores.
- Dejar la aplicación casi lista para su despliegue en un servidor.

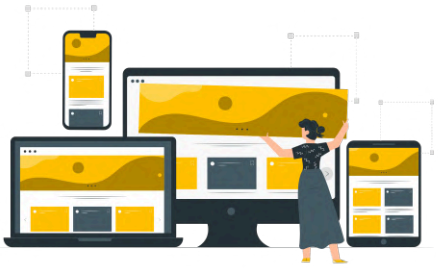
Introducción a Bootstrap y styled-components

Hasta ahora, hemos utilizado **Bootstrap y Bootswatch** como base para la estilización de nuestra aplicación. Gracias a estas herramientas, hemos podido aplicar estilos fácilmente sin necesidad de escribir mucho código CSS personalizado. Sin embargo, en esta etapa del proyecto, es importante optimizar la forma en que aplicamos los estilos para lograr un mejor control sobre la apariencia de nuestra aplicación.

Para esto, revisaremos dos enfoques diferentes:

- Mejorar la utilización de **Bootstrap** para asegurarnos de que los elementos sean completamente responsivos.
- Introducir **styled-components**, una librería que nos permite definir estilos directamente dentro de nuestros componentes de React, evitando la necesidad de archivos CSS externos y haciendo que el código sea más modular y reutilizable.

Ajustes de responsividad con Bootstrap



Uno de los problemas más comunes en una aplicación web es que, aunque se vea bien en pantallas grandes, no siempre se adapta bien a dispositivos móviles. Para asegurarnos de que nuestra aplicación tenga un diseño responsivo, revisaremos el uso del **sistema de grillas** de Bootstrap. Este sistema nos permite dividir el diseño en columnas y hacer que los elementos se ajusten de manera automática dependiendo del tamaño de la pantalla.

Por ejemplo, en la lista de productos, si usamos solo una columna, se verán muy grandes en pantallas grandes y muy pequeños en dispositivos móviles. En su lugar, podemos usar las clases de Bootstrap para que en pantallas pequeñas haya una sola columna (`col-12`), en pantallas medianas haya dos (`col-md-6`) y en pantallas grandes haya tres (`col-lg-4`).

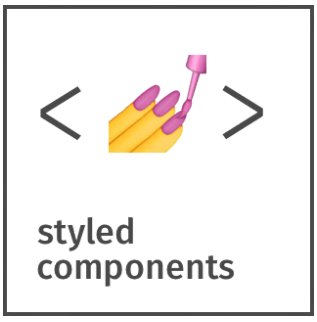
Un ejemplo de código para estructurar correctamente la lista de productos usando el sistema de grillas sería el siguiente:

```
<div className="container">

  <div className="row">

    <div className="col-12 col-md-6 col-lg-4">

      <div className="card">

        <img src="producto.jpg" className="card-img-top img-fluid"
alt="Producto" />

        <div className="card-body">

          <h5 className="card-title">Producto 1</h5>

          <p className="card-text">$1000</p>

          <button className="btn btn-primary w-100">Comprar</button>

        </div>

      </div>

    </div>

  </div>

</div>
```

Otro punto clave es la **Navbar**. Para hacer que la barra de navegación sea completamente responsiva, podemos usar el componente `navbar-toggler` de Bootstrap, que permite que el menú se colapse en dispositivos móviles y se expanda en pantallas más grandes.

Una vez que revisemos estos aspectos, cada estudiante deberá aplicar estos ajustes en su propia aplicación, revisando cada sección para asegurarse de que todo se vea bien en distintos tamaños de pantalla.

Introducción a styled-components



Styled-components es una librería que nos permite definir estilos dentro de los mismos componentes de React, sin necesidad de usar clases de CSS. Esto ayuda a que nuestro código sea más ordenado y modular, ya que los estilos estarán directamente relacionados con los componentes en los que se aplican.

Para instalar styled-components, debemos ejecutar el siguiente comando:

```
npm install styled-components
```

Una vez instalada, podemos definir estilos de manera sencilla. Por ejemplo, si queremos personalizar un botón de compra, podemos hacerlo de la siguiente manera:

```
import styled from "styled-components";

const BotonCompra = styled.button`
  background-color: #ff5733;
  color: white;
  padding: 10px 15px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  &:hover {
    background-color: #c70039;
  }
`;

function Producto() {
  return <BotonCompra>Comprar</BotonCompra>;
}
```

El beneficio de este enfoque es que cada componente tendrá sus propios estilos sin necesidad de preocuparse por conflictos de clases en CSS. Esto es especialmente útil en proyectos grandes o cuando trabajamos en equipo.

Creación de un diseño básico y responsive

Ahora que entendemos cómo mejorar la responsividad con Bootstrap y cómo estructurar estilos con styled-components, vamos a agregar algunos detalles visuales y funcionales que harán que nuestra aplicación luzca más profesional.

Iconos con React Icons

El uso de iconos mejora la experiencia del usuario, ya que permite representar visualmente las acciones de la aplicación. En lugar de tener solo texto en los botones, podemos agregar pequeños iconos para que sean más intuitivos.

Para instalar React Icons, usamos el siguiente comando:

```
npm install react-icons
```

Luego, podemos importar un icono y usarlo dentro de un botón o cualquier otro elemento.

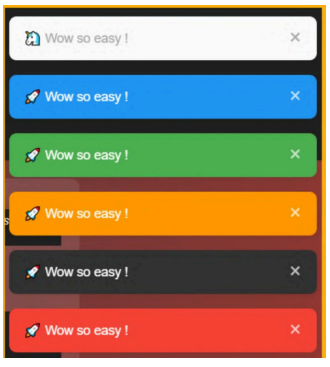
Un buen ejemplo sería agregar un icono de carrito en la navbar:

```
import { FaShoppingCart } from "react-icons/fa";

<button className="btn btn-warning">

  <FaShoppingCart /> Carrito

</button>
```

Esta simple mejora hace que la interfaz sea más clara y atractiva para los usuarios.

Notificaciones con React Toastify

Para mejorar la interacción con el usuario, podemos agregar notificaciones cuando se realiza una acción, como agregar un producto al carrito. Para esto, usaremos la



librería **React Toastify**, que permite mostrar alertas en pantalla de una manera elegante y no intrusiva.

Para instalarla, ejecutamos:

```
npm install react-toastify
```

Luego, podemos definir una notificación para cuando se agregue un producto al carrito:

```
import { ToastContainer, toast } from "react-toastify";
import "react-toastify/dist/ReactToastify.css";

const handleAddToCart = () => {
  toast.success("Producto agregado al carrito!");
};

return (
  <>
    <button className="btn btn-success" onClick={handleAddToCart}>
      Agregar al carrito
    </button>
    <ToastContainer />
  </>
);
```

Esta pequeña mejora hace que la aplicación sea más dinámica y amigable para los usuarios.

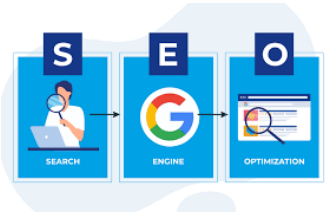
Aplicación de estilos en componentes (botones, formularios, productos)

Optimización de formularios

```
<form className="p-4 border rounded shadow">

  <div className="mb-3">

    <label className="form-label">Email</label>
```

```
    <input type="email" className="form-control" required />
  </div>
  <div className="mb-3">
    <label className="form-label">Contraseña</label>
    <input type="password" className="form-control" required />
  </div>
  <button className="btn btn-primary w-100">Ingresar</button>
</form>
```

Optimización para SEO



El **SEO (Search Engine Optimization)** es un conjunto de técnicas que ayudan a que nuestra aplicación aparezca en los resultados de búsqueda de Google. Aunque React es excelente para construir interfaces dinámicas, por defecto no es muy bueno para SEO porque la mayoría del contenido se genera en el navegador en lugar de en el servidor.

Para mejorar el SEO en React, podemos usar **React Helmet**, una herramienta que nos permite modificar las etiquetas `<title>` y `<meta>` de cada página.

Para instalarlo, ejecutamos:

```
npm install react-helmet-async
```

Ejemplo de implementación en una página de productos:

```
import { Helmet } from "react-helmet-async";

function Productos() {
  return (
    <>
      <Helmet>
```

```
        <title>Productos | Mi Tienda</title>
        <meta name="description" content="Explora nuestra variedad de
productos." />
      </Helmet>
      <h1>Lista de productos</h1>
    </>
  );
}
```

Nueva Tarea en Talento Lab

El cliente de Talento Lab está satisfecho con los avances del proyecto y ahora nos pide llevar la apariencia y la interacción de la aplicación al siguiente nivel. Queremos que la aplicación no solo funcione correctamente, sino que también brinde una experiencia visualmente atractiva y responsiva. Para ello, vamos a centrarnos en la optimización del diseño y la responsividad, utilizando herramientas como **Bootstrap** y **styled-components**. Además, vamos a integrar mejoras de UX/UI, como **React Icons** y **React Toastify**, para hacer la interacción más intuitiva.

Objetivos:

1. Optimizar el diseño y la responsividad de la aplicación utilizando Bootstrap y styled-components.
2. Mejorar la experiencia del usuario agregando iconos e interacciones visuales con React Icons y React Toastify.
3. Implementar ajustes visuales y de interacción para hacer la interfaz más atractiva y funcional.

4. Asegurar que la aplicación sea completamente responsiva, adaptándose a diferentes tamaños de pantalla.
5. Dejar la aplicación casi lista para su despliegue en un servidor, con los ajustes finales de SEO utilizando React Helmet.

Requerimientos:

1. **Diseñar una interfaz responsiva con Bootstrap y styled-components:**
    - Utiliza el sistema de grillas de Bootstrap para que los productos se muestren de manera eficiente en pantallas grandes, medianas y pequeñas.
    - Aplica estilos a los componentes utilizando styled-components para personalizar el diseño y hacer que el código sea más modular y limpio.

2. **Integrar iconos con React Icons:**
    - Instala React Icons y agrega iconos en botones y otros elementos interactivos. Por ejemplo, agrega un icono de carrito en la navbar para mejorar la claridad visual de la acción de compra.

3. **Agregar notificaciones con React Toastify:**
    - Integra React Toastify para mostrar notificaciones de manera elegante y no intrusiva. Por ejemplo, muestra un mensaje de éxito cuando un producto se agrega al carrito.

4. **Mejorar la accesibilidad y el SEO de la aplicación:**
    - Utiliza **React Helmet** para mejorar el SEO de la aplicación. Modifica las etiquetas `<title>` y `<meta>` para asegurar que la aplicación sea fácilmente indexada por los motores de búsqueda.

5. **Preparar la aplicación para su despliegue:**
    - Asegúrate de que todos los estilos sean responsivos y de que la interfaz funcione correctamente en diferentes dispositivos y tamaños de pantalla.
    - Revisa que la aplicación esté lista para su despliegue en un servidor, asegurando la optimización de los tiempos de carga y la experiencia del usuario en general.

Reflexión final

En esta clase, optimizamos la apariencia y la funcionalidad de nuestra aplicación para que esté lista para su despliegue. En la próxima clase, veremos cómo subirla a un servidor para que cualquiera pueda acceder a ella. ¡Estamos cada vez más cerca del objetivo final! 🚀

Materiales y Recursos Adicionales:

- ★ Bootstrap
- ★ styled-components
- ★ React Icons
- ★ React Toastify
- ★ Framer Motion

Preguntas para Reflexionar:

- ¿Cómo puedes asegurar que una aplicación React sea completamente responsiva utilizando Bootstrap y styled-components?
- En un proyecto con muchos componentes, ¿qué ventajas ofrece usar styled-components frente a usar clases CSS tradicionales o Bootstrap?
- ¿Qué aspectos de la experiencia del usuario mejoran al agregar iconos con React Icons y notificaciones con React Toastify en una aplicación?
- ¿Cómo podría la implementación de React Helmet mejorar el SEO de una aplicación React, y qué consideraciones debes tener al utilizarlo?

Próximos Pasos:

- Diseño mobile-first y adaptabilidad de la aplicación.
- Mejores prácticas de diseño UI/UX.
- Revisión de la aplicación para mejorar la experiencia de usuario.

Buenos Aires aprende

Agencia de Habilidades para el Futuro

Buenos Aires Ciudad