



Buenos
Aires
Ciudad

Agencia de Habilidades
para el Futuro

<Talento
Tech />

React JS

Clase 14 | Diseño Responsivo y UX

¡Les damos la bienvenida!



Vamos a comenzar a grabar la clase.

Índice

Estilización con Bootstrap o styled-components

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

Diseño Responsivo y UX

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

Despliegue de la Aplicación

- Despliegue de la aplicación en plataformas como Vercel o Netlify.
- Pruebas de funcionamiento en el entorno de producción.
- Revisión de los pasos de despliegue y configuración.

Objetivos de la Clase

- 1** Aplicar el enfoque mobile-first para mejorar la adaptabilidad de la aplicación en distintos dispositivos.
- 2** Aplicar las mejores prácticas de diseño UI/UX para crear interfaces atractivas, fáciles de usar y que proporcionen una experiencia positiva al usuario
- 3** Revisar la aplicación y hacer ajustes que optimicen la interacción del usuario, asegurando que la aplicación sea funcional y agradable tanto en dispositivos móviles como en escritorios.
- 4** Agregar paginación en el componente AllProductos para optimizar la carga y visualización de los productos.

Diseño Mobile-First y Adaptabilidad de la Aplicación

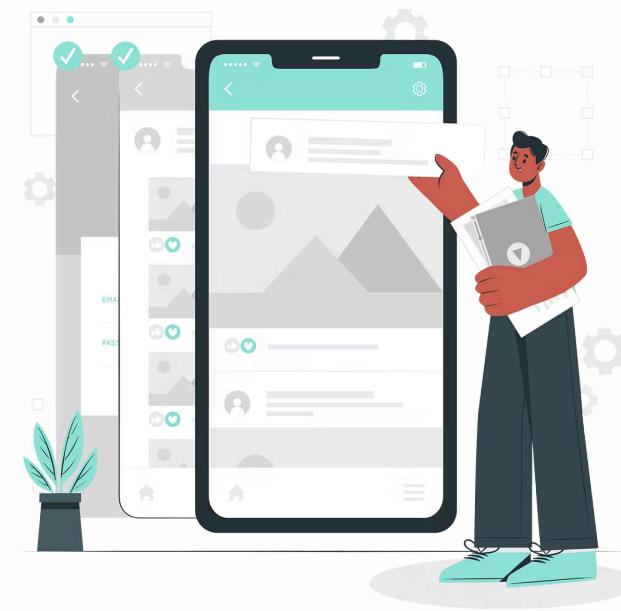
Diseño Mobile-First

¿Qué es?

Enfoque de diseño que prioriza dispositivos móviles y luego adapta a pantallas más grandes.

¿Por qué es importante?

Más usuarios acceden desde móviles. Optimiza la experiencia en pantallas pequeñas.



Beneficios del Diseño Mobile-First



Optimiza el rendimiento

Mejora la velocidad y eficiencia en dispositivos móviles.



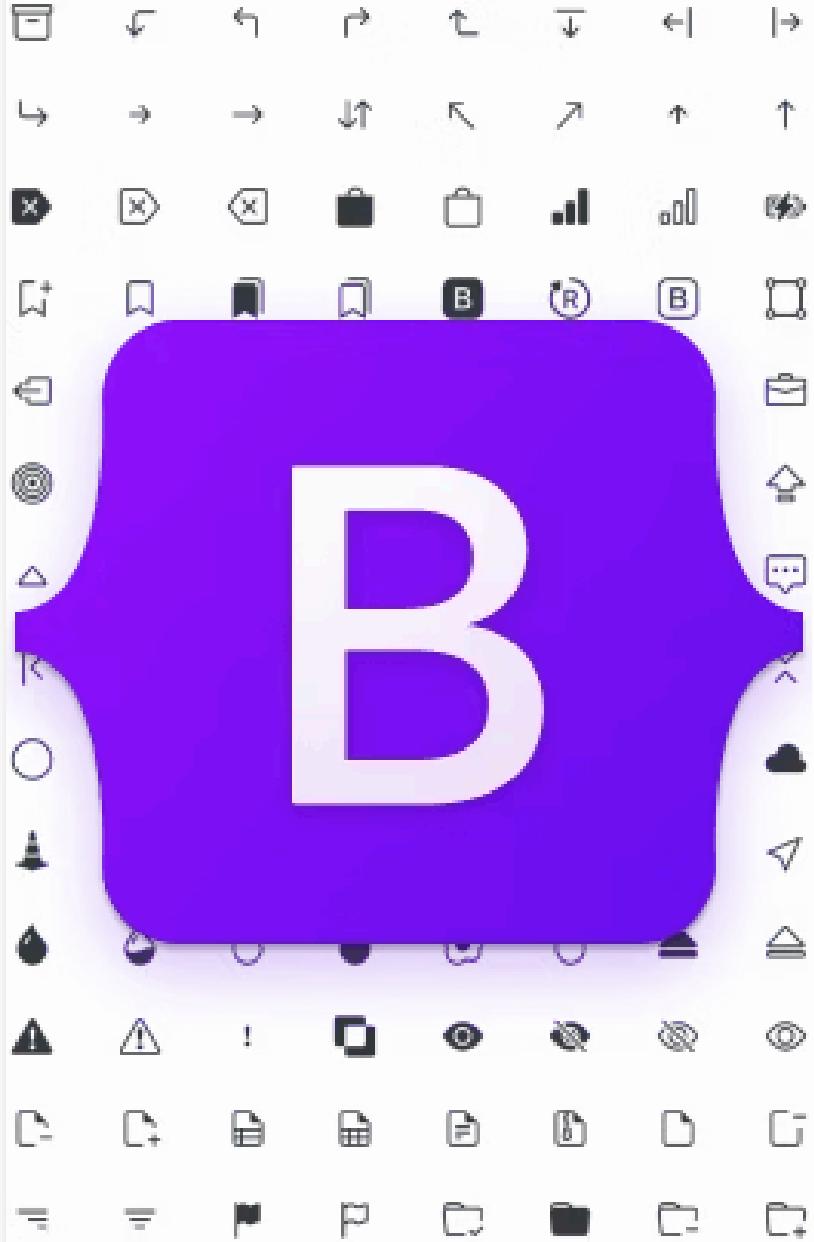
Prioriza la usabilidad

Enfoca la experiencia en pantallas pequeñas primero.



Mejora SEO

Favorece el posicionamiento en buscadores móviles.



Implementación con Bootstrap

Sistema de Grillas

Bootstrap ofrece un sistema de grillas responsivas para estructurar la interfaz.

Clases Responsivas

Utiliza clases como col-12, col-md-6, col-lg-4 para adaptar el diseño.

Flexibilidad

Permite ajustar la distribución según el tamaño de la pantalla.

Ejemplo: distribución de productos en diferentes tamaños de pantalla

```
<div className="container">
  <div className="row">
    <div className="col-12 col-md-6 col-lg-4">
      <div className="card">
        
        <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>
```

Explicación de la Grilla

1 Móviles

Los productos ocupan toda la pantalla (col-12).

2 Tablets

Los productos se organizan en 2 columnas (col-md-6).

3 Escritorio

Los productos se distribuyen en 3 columnas (col-lg-4).



Prueba de Adaptabilidad



Paso 1

Abre las DevTools en tu navegador (F12).

Paso 2

Activa el modo de dispositivo (Ctrl + Shift + M).

Paso 3

Cambia el tamaño de la pantalla para ver cómo se adapta el diseño.

Mejores Prácticas de Diseño UI/UX

Mejores Prácticas de Diseño UI/UX

UX se refiere a cómo se siente el usuario al interactuar con la aplicación. Un diseño deficiente puede alejar a los usuarios, mientras que una interfaz clara y bien organizada mejora la conversión y retención. **Principios clave de UX/UI:**



Simplicidad

Diseñar interfaces sin elementos innecesarios.



Consistencia

Mantener el mismo estilo en botones, tipografías y colores.



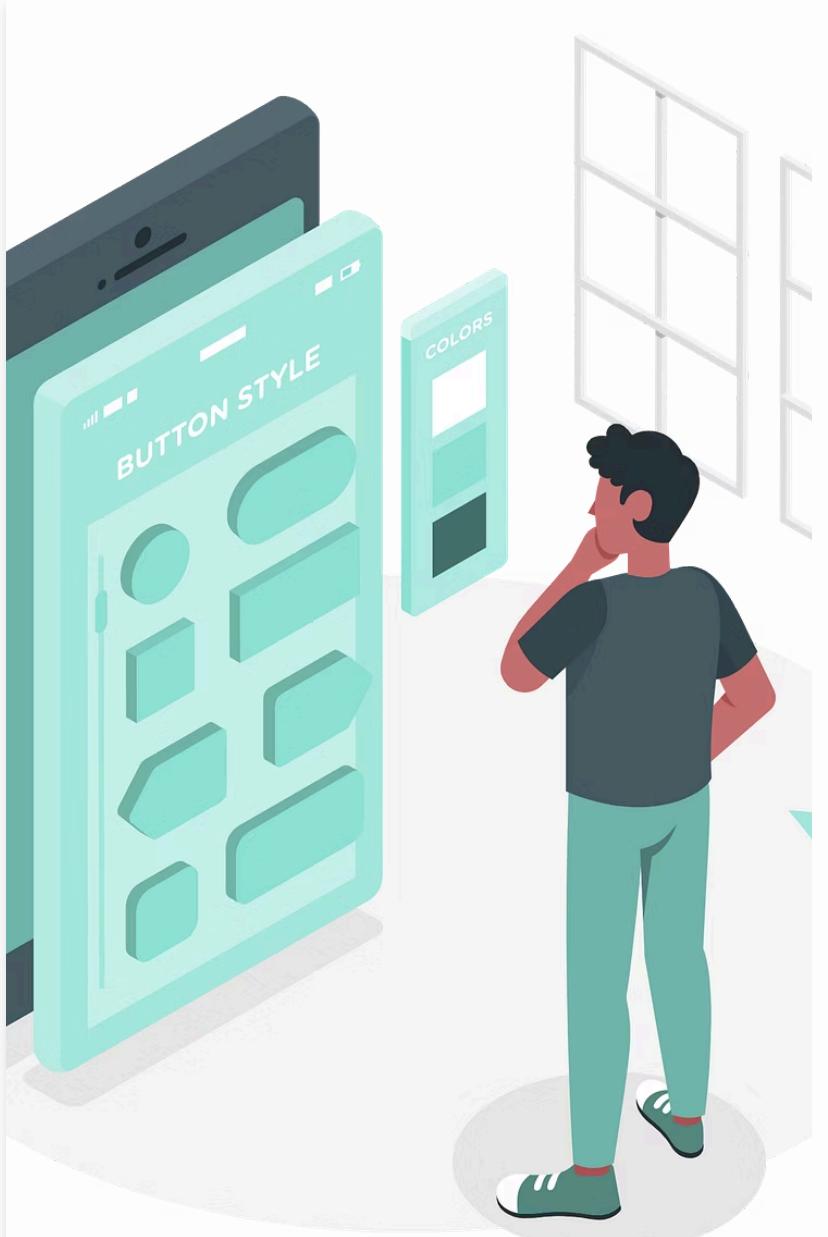
Accesibilidad

Garantizar que la aplicación sea fácil de usar para todos.



Feedback Visual

Mostrar estados de carga, confirmaciones y errores.



Botones Accesibles

Características

Los botones deben ser grandes y fáciles de presionar en pantallas táctiles.

Ejemplo

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

Mejoras en Formularios

Claridad

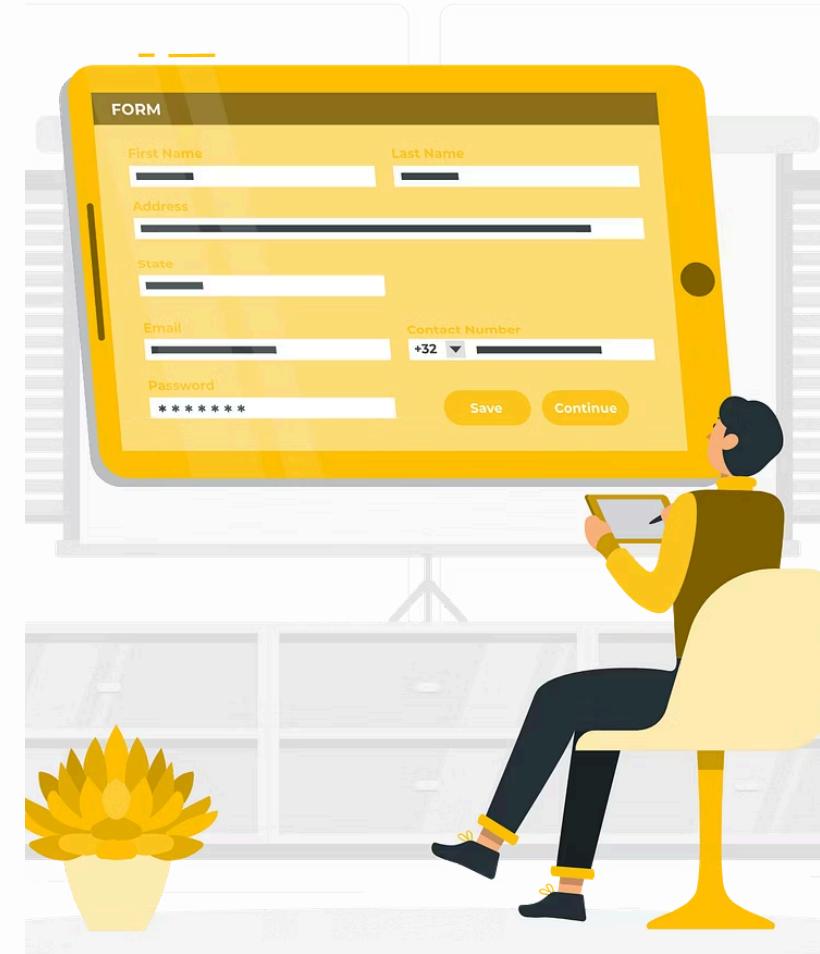
Los formularios deben ser claros, con etiquetas bien definidas.

Espaciado

Asegurar suficiente espaciado entre elementos del formulario.

Feedback

Proporcionar retroalimentación visual al usuario durante la interacción.



Ejemplo de Formulario Optimizado

```
<form className="p-4 border rounded shadow">
  <div className="mb-3">
    <label className="form-label">Email</label>
    <input type="email" className="form-control" placeholder="Ingresa tu email" required />
  </div>
  <div className="mb-3">
    <label className="form-label">Contraseña</label>
    <input type="password" className="form-control" placeholder="*****" required />
  </div>
  <button className="btn btn-success w-100">Ingresar</button>
</form>
```

Espaciado y Legibilidad

Importancia

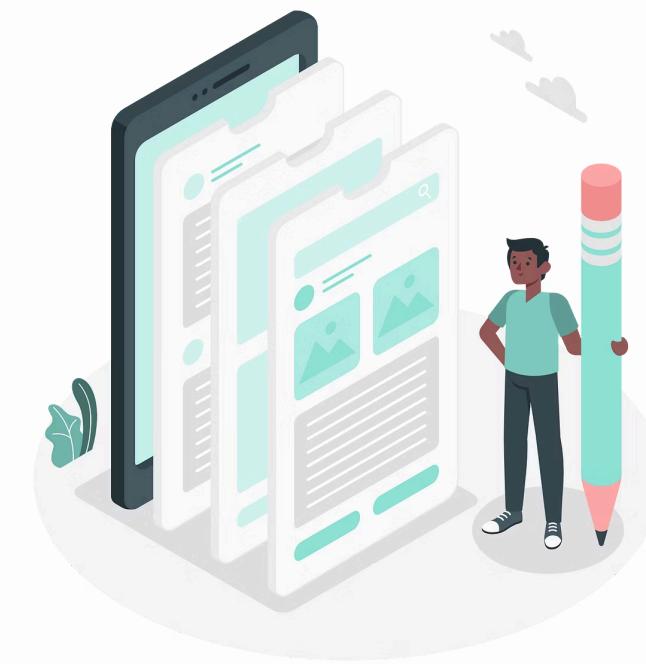
Los elementos deben tener suficiente espacio para evitar errores de clic.

Técnicas

Usar márgenes y padding adecuados entre elementos.

Beneficios

Mejora la legibilidad y reduce la fatiga visual del usuario.



CONTACT

NAME

EMAIL

SUBJECT

AND

MESSAGE

SUBMIT

Ejemplo de Espaciado

```
<div className="p-4 my-3 border rounded shadow">
  <h3 className="mb-3">Formulario de Contacto</h3>
  <input type="text" className="form-control mb-3"
placeholder="Nombre" />
  <input type="email" className="form-control mb-3"
placeholder="Correo Electrónico" />
  <button className="btn btn-success w-
100">Enviar</button>
</div>
```

Revisión de la Aplicación para Mejorar la Experiencia de Usuario



Revisión de la Aplicación

Paso 1

Probar la aplicación en diferentes dispositivos.

Paso 2

Utilizar herramientas de prueba responsiva.

Paso 3

Ajustar el diseño según los resultados de las pruebas.

Herramientas para Pruebas Responsivas



Chrome DevTools

Modo Responsive: F12 > Ctrl + Shift + M



Responsinator

Simulador online para múltiples dispositivos



Google Mobile-Friendly Test

Herramienta de Google para evaluar la adaptabilidad móvil

Accesibilidad y Etiquetas ARIA

Las etiquetas ARIA mejoran la accesibilidad para usuarios con discapacidades.

Ejemplo

```
<button className="btn btn-primary" aria-label="Agregar al carrito">  
  Agregar  
</button>
```



Implementación de un Paginador en la vista de productos

Implementación de un Paginador en la vista de productos

Cuando se manejan grandes cantidades de productos en una tienda, mostrarlos todos en una sola página puede afectar la performance y la experiencia del usuario. La paginación permite dividir la lista de productos en páginas más pequeñas, cargando solo una parte a la vez.

Beneficios de la paginación:



Mejora el rendimiento, cargando solo los productos necesarios.



Hace que la navegación sea más organizada y fácil de usar.



Permite a los usuarios explorar los productos de manera más eficiente.

¿Cómo implementarlo en React?

1

Dividir los productos en páginas: Determinar cuántos

2

Crear botones de navegación: Permitir moverse entre páginas.

3

Actualizar la vista dinámicamente: Renderizar

productos se mostrarán por

página.

Ejemplo:

solo los productos de la
página actual.

```
import { useState } from "react";
const AllProductos = ({ productos }) => {
  const productosPorPagina = 6; // Cantidad de productos a mostrar por página
  const [paginaActual, setpaginaActual] = useState(1);
  // Calcular el índice de los productos a mostrar en la página actual
  const indiceUltimoProducto = paginaActual * productosPorPagina;
  const indicePrimerProducto = indiceUltimoProducto - productosPorPagina;
  const productosActuales = productos.slice(indicePrimerProducto, indiceUltimoProducto);
```

```
// Cambiar de página
const totalPaginas = Math.ceil(productos.length / productosPorPagina);
const cambiarPagina = (numeroPagina) => setpaginaActual(numeroPagina);
return (
  <div className="container">
    <h2 className="my-4">Todos los Productos</h2>
    <div className="row">
      {productosActuales.map((producto) => (
```

```
<div key={producto.id} className="col-12 col-md-6 col-lg-4">
  <div className="card">
    <img src={producto.imagen} className="card-img-top" alt={producto.nombre} />
    <div className="card-body">
      <h5 className="card-title">{producto.nombre}</h5>
      <p className="card-text">${producto.precio}</p>
      <button className="btn btn-primary w-100">Aregar al carrito</button>
    </div>
  </div>
</div>
))}
```

```
{/* Paginador */}
<div className="d-flex justify-content-center my-4">
  {Array.from({ length: totalPaginas }, (_, index) => (
    <button
      key={index + 1}
      className={`btn mx-1 ${paginaActual === index + 1 ? "btn-primary" : "btn-outline-primary"}`}
      onClick={() => cambiarPagina(index + 1)}
    >
```

```
        {index + 1}
      </button>
    ))}
  </div>
</div>
);
};

export default AllProductos;
```



Reflexión Final

En esta clase, optimizamos la responsividad y la experiencia de usuario, asegurando que la aplicación sea intuitiva y accesible en distintos dispositivos. Aplicamos el enfoque **mobile-first**, mejoramos la usabilidad con prácticas de UX/UI y realizamos una revisión final para detectar y corregir posibles problemas. Estos ajustes no solo mejoran la apariencia, sino que también hacen que la navegación sea más fluida y eficiente. Una aplicación bien diseñada no solo funciona correctamente, sino que también ofrece una experiencia agradable y profesional.

Materiales y Recursos Adicionales



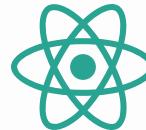
[Bootstrap](#)

Framework CSS para diseño responsivo



[styled-components](#)

Biblioteca para estilizar componentes React



[React Icons](#)

Conjunto de iconos para aplicaciones React



[React Toastify](#)

Biblioteca para notificaciones en React

Preguntas para Reflexionar

1 Mobile-First

¿Cómo mejora la experiencia del usuario al aplicar el enfoque mobile-first?

2 Navegación Intuitiva

¿Qué elementos visuales y funcionales son más importantes para una navegación intuitiva?

3 Accesibilidad

¿Cómo evaluarías si la accesibilidad de tu aplicación es adecuada para todos los usuarios?

4 Mejora Continua

¿Qué estrategias implementarías para seguir mejorando la experiencia del usuario en futuras versiones?



Próximos Pasos

- 1
- 2
- 3

Despliegue de la aplicación en plataformas como Vercel o Netlify

Pruebas de funcionamiento en el entorno de producción

Revisión de los pasos de despliegue y configuración





Ejercicios Prácticos

Nueva Tarea en Talento Lab

El cliente de Talento Lab ha solicitado un enfoque avanzado para mejorar la experiencia visual e interactiva de la aplicación, con un énfasis en la optimización para dispositivos móviles y la mejora de la usabilidad general. La aplicación debe ser completamente responsive y ofrecer una experiencia de usuario fluida y accesible, manteniendo una apariencia atractiva en diferentes tamaños de pantalla..





Ejercicio Práctico

Obligatorio

Objetivos:



Aplicar enfoque mobile-first para asegurar responsividad completa.



Implementar mejores prácticas de diseño para interfaces accesibles.



Integrar React Icons y React Toastify para mejorar la navegación.



Optimizar SEO con React Helmet para mejorar la indexación.



Ejercicio Práctico

Obligatorio

Requisitos del Proyecto:

 **Optimización Responsiva con Bootstrap y Styled-components:**

- Utiliza el sistema de grillas de **Bootstrap** para organizar los componentes de manera responsiva, adaptando la distribución de los productos y otros elementos a diferentes tamaños de pantalla (móviles, tablets, escritorios).
- Aplica **styled-components** para crear un diseño personalizado y modular, mejorando la claridad y mantenibilidad del código.

 **Interactividad Mejorada con React Icons y React Toastify:**

- Instala **React Icons** y agrega iconos en los botones de acción, como en la barra de navegación (por ejemplo, el ícono de carrito en el botón de compra).

Integra **React Toastify** para mostrar notificaciones visuales al usuario, como mensajes de éxito cuando un producto es agregado al carrito de compras.



Ejercicio Práctico

Obligatorio

Requisitos del Proyecto:

Mejora de la Accesibilidad y SEO:

- Utiliza etiquetas **ARIA** en botones y enlaces para mejorar la accesibilidad.

Usa **React Helmet** para modificar las etiquetas `<title>` y `<meta>`, mejorando el SEO de la aplicación y asegurando que sea fácilmente indexada por los motores de búsqueda.

Pruebas y Preparación para el Despliegue:

- Realiza pruebas en diferentes dispositivos para garantizar la compatibilidad.
- Revisa que todos los elementos de la interfaz se comporten correctamente en pantallas pequeñas y grandes, asegurando tiempos de carga rápidos y una experiencia de usuario fluida.

Paginación en AllProducts:

- Implementa una paginación funcional para mostrar los productos en páginas separadas.