Taller Interactivo: Diseño Responsive en React

Julian F. Latorre

1 Introducción

Este taller está diseñado para enseñar los principios del diseño responsive en React de manera participativa. Los estudiantes trabajarán en grupos, asumiendo diferentes roles para resolver un desafío de diseño.

2 Objetivo

Crear una página web responsive utilizando React que se adapte a diferentes tamaños de pantalla y dispositivos.

3 Roles

1. **Diseñador UX/UI:** Encargado de crear wireframes y mockups para diferentes tamaños de pantalla.
2. **Desarrollador Front-end:** Responsable de implementar la estructura y estilos CSS.
3. **Desarrollador React:** Encargado de crear y gestionar los componentes React.
4. **Experto en Media Queries:** Responsable de implementar las media queries para lograr el diseño responsive.
5. **Tester:** Encargado de probar la aplicación en diferentes dispositivos y reportar problemas.

4 Actividad

1. Dividir la clase en grupos de 5 personas, asignando un rol a cada miembro.
2. Presentar el desafío: Crear una página de inicio para una tienda en línea con las siguientes secciones:
    - Encabezado con logo y menú de navegación

- Banner principal
- Sección de productos destacados
- Pie de página con información de contacto

3. Cada grupo tiene 90 minutos para completar el desafío.
4. Los grupos presentarán sus soluciones y explicarán sus decisiones de diseño.

5 Proceso

1. **Fase de Diseño (20 minutos):** El Diseñador UX/UI crea wireframes para móvil, tablet y desktop.
2. **Fase de Desarrollo (50 minutos):**
   - El Desarrollador Front-end crea la estructura HTML y CSS base.
   - El Desarrollador React implementa los componentes necesarios.
   - El Experto en Media Queries ajusta los estilos para diferentes tamaños de pantalla.
3. **Fase de Pruebas (20 minutos):** El Tester verifica la funcionalidad en diferentes dispositivos y reporta problemas.
4. **Presentaciones (5 minutos):** Cada grupo presenta su solución a la clase brevemente en 5 minutos.

6 Evaluación

Los proyectos serán evaluados según los siguientes criterios:

- Diseño atractivo y usable en todos los tamaños de pantalla
- Correcta implementación de componentes React
- Uso efectivo de media queries y técnicas de CSS responsive
- Funcionamiento sin errores en diferentes dispositivos
- Calidad de la presentación y explicación del proceso

7 Implementación de Lazy Loading para Imágenes

Para mejorar el rendimiento de la tienda en línea, implementaremos lazy loading para las imágenes de los productos. Sigue estos pasos:

1. **Instalar dependencias:** Ejecuta el siguiente comando en la terminal:

```
npm install react-lazy-load-image-component
```

2. **Importar el componente:** En el archivo del componente de producto, agrega la siguiente importación:

```
import { LazyLoadImage } from 'react-lazy-load-image-component';
import 'react-lazy-load-image-component/src/effects/blur.css';
```

3. **Reemplazar las etiquetas de imagen:** Sustituye las etiquetas ¡img¿ por el componente LazyLoadImage:

```
<LazyLoadImage
  src={product.imageUrl}
  alt={product.name}
  effect="blur"
  width={300}
  height={200}
/>
```

4. **Ajustar estilos:** Asegúrate de que los estilos CSS se apliquen correctamente al nuevo componente.

5. **Implementar placeholders:** Crea imágenes de baja resolución para usar como placeholders mientras se cargan las imágenes completas.

6. **Probar el rendimiento:** Utiliza herramientas como Lighthouse, WebPageTest o el inspector del navegador para medir la mejora en el tiempo de carga de la página.

8 Recursos

- Documentación de React: `https://react.dev/learn`
- Guía de Flexbox: `https://css-tricks.com/snippets/css/a-guide-to-flexbox/`
- Guía de Grid: `https://css-tricks.com/snippets/css/complete-guide-grid/`
- React Lazy Load Image Component: `https://www.npmjs.com/package/react-lazy-load-image-component`