

Actividad Áulica N19

CARRERA: Ingeniería en Sistemas de la Información.

MATERIA: Paradigmas y Lenguajes de Programación III.

COMISIÓN: “U” (única).

PROFESOR: Mgter. Encina Agustín.

ESTUDIANTES: Tabarez, Kiara.

PROYECTO: TQComponents

FECHA: 21-10-2025.

Repositorio: https://github.com/Kiaratabarez/Kiaratabarez-TQcomponentes_Frontend.git

Plugin Seleccionado

Nombre: Owl Carousel 2

Versión: 2.3.4

Tipo: Plugin jQuery para carruseles responsive

Sitio oficial: <https://owlcarousel2.github.io/OwlCarousel2/>

El Owl Carousel es un plugin ampliamente utilizado en proyectos web modernos debido a su estabilidad, flexibilidad y facilidad de integración. Permite crear presentaciones de imágenes o productos de manera ordenada, fluida y estéticamente agradable.

Justificación de la Selección

- La elección de Owl Carousel 2 se fundamenta en las siguientes razones:
- Es uno de los plugins más populares y maduros para jQuery, con amplia documentación y soporte comunitario.
- Posee compatibilidad total con dispositivos móviles (responsive design) y gestos táctiles (touch-friendly).
- Su integración es sencilla, requiriendo pocas líneas de código.
- Ofrece amplias opciones de personalización, incluyendo velocidad, navegación, puntos indicadores y animaciones.
- Contribuye a una interfaz más moderna, visualmente atractiva y dinámica, mejorando la interacción del usuario con el sitio.

Proceso de Implementación

Incorporación de las dependencias

En el archivo index.html, añadí los enlaces a las hojas de estilo y scripts de Owl Carousel desde un CDN confiable.

Estas dependencias las ubique dentro del <head> y antes del cierre del </body>

Estructura HTML

Reemplace el contenedor estático de productos los destacados, originalmente colocado en formato *grid*, por un contenedor dinámico compatible con Owl Carousel añadiendo la línea:

```
<div class="owl-carousel owl-theme productos-carousel"></div>.
```

Este cambio me permitió que los productos se mostrarán de manera deslizable y animada, manteniendo la estética general.

Desarrollo de la función JavaScript

Dentro del archivo scripts.js, creé la función “cargarProductosDestacadosCarousel()” con los siguientes propósitos:

- Cargar dinámicamente los productos destacados del catálogo.
- Inicializar el componente Owl Carousel con parámetros personalizados.
- Configurar navegación por flechas, puntos indicadores y comportamiento automático.

Estas opciones configuradas permiten ajustar el número de productos visibles según el ancho de pantalla, garantizando así la adaptabilidad del componente.

Personalización mediante CSS

En el archivo styles.css se incorporaron ajustes visuales personalizados, adaptando la apariencia del carrusel al estilo general del sitio.

Entre las modificaciones se incluyeron los colores, márgenes, tamaños de imagen y transiciones, buscando mantener coherencia visual con el resto del diseño de TQComponents.

Funcionalidades Implementadas

El carrusel que se incorporó posee las siguientes características:

- Reproducción automática con intervalo de 3 segundos.
- Pausa automática al pasar el cursor (hover pause).
- Navegación lateral mediante flechas.
- Puntos indicadores de posición (dots navigation).
- Adaptabilidad completa a diferentes tamaños de pantalla.
- Animaciones de transición suaves y personalizadas.

Estas funciones contribuyen a una navegación más intuitiva y atractiva, priorizando la interacción del usuario con el contenido destacado.

Resultados Obtenidos

La implementación de Owl Carousel 2 logró mejorar notablemente la presentación visual y funcional del sitio web.

El carrusel aporta dinamismo, fluidez y profesionalismo al diseño, reemplazando una disposición estática por una experiencia de desplazamiento moderno e interactivo.

Además, su integración no afectó el rendimiento del sitio, manteniendo tiempos de carga óptimos y compatibilidad con dispositivos móviles y navegadores actuales.

Conclusión

La integración del plugin Owl Carousel 2 en el proyecto me permitió elevar la calidad visual e interactiva del sitio web.

El resultado final combina eficiencia técnica, diseño adaptable y una experiencia de usuario mejorada, alineándose con las buenas prácticas de desarrollo web.

La utilización de herramientas como esta demuestra la importancia de incorporar librerías externas confiables, que potencien el desarrollo sin sacrificar rendimiento ni coherencia estética.