



UNIVERSIDAD PRIVADA DEL VALLE
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN SISTEMAS
INFORMÁTICOS

PROYECTO DE FIN DE SEMESTRE DE PROGRAMACION WEB I

MANUAL TECNICO

PAGINA WEB DE COMPRA DE

VIDEOJUEGOS

KFCZONE

ESTUDIANTE: YAMIL DYLAN LUNA QUISPE

DOCENTE: HENRY MIRANDA ORDOÑEZ

PARALELO: B

La Paz - Bolivia

2025

INDICE GENERAL

MANUAL TÉCNICO DE LA PAGINA WEB KFCZONE	4
1. INTRODUCCIÓN GENERAL	4
2. DESCRIPCIÓN DEL SISTEMA.....	4
3. OBJETIVOS.....	4
4. Público Objetivo	4
5. Contexto del Sistema	5
6. Flujo General del Sitio Web	5
7. Estructura del Proyecto	5
8. Desarrollo del Frontend	6
9. Despliegue del Sistema	7
10. DISEÑO	7
10.1 Tipografía	7
10.2 Paleta de Colores.....	7
10.3 Wireframes	8
10.4 Responsividad	9
11. ESTILOS Y SCRIPTS GLOBALES	10
11.1 Reset del Documento	10
11.2 Variables Globales.....	11
11.3 Contenedor	11
11.4 Encabezado y Navegación.....	11

11.4.1 Zona de Acciones	11
11.5 Pie de Página	12
11.6 Estilos Responsivos.....	12
11.7 Menú Hamburguesa	12
11.8 Script Global	13
11.10 Consideraciones Técnicas.....	13
12. CONCLUSIONES	14
13. BIBLIOGRAFÍA	14

MANUAL TÉCNICO DE LA PAGINA WEB KFCZONE

1. INTRODUCCIÓN GENERAL

Se presenta un manual técnico de la página web KFCZONE, página web orientada a la visualización, exploración, consulta y simulación de compra de videojuegos. El propósito principal de este documento es describir de forma estructurada la arquitectura, funcionamiento, componentes, navegación y aspectos técnicos que conforman el sistema.

El documento proporciona una visión completa de los módulos que componen el sistema, la relación visual entre páginas, los mecanismos de interacción implementados mediante JavaScript y el conjunto de estilos globales aplicados.

2. DESCRIPCIÓN DEL SISTEMA

KFCZONE es una plataforma web estática que organiza y presenta contenido relacionado con videojuegos. El sistema no vende productos reales, sino que simula un entorno de catálogo, donde el usuario puede visualizar juegos, explorar categorías y utilizar un carrito de compras de carácter demostrativo.

El sistema está compuesto por seis secciones principales:

- Inicio de Sesión
- Tienda
- Ofertas
- Explorar
- Página individual de juego
- Carrito de Compras

Además, se incorpora un conjunto de estilos y scripts individuales y globales que controlan la navegación, el menú responsable y elementos compartidos en todas las páginas. KFCZONE permite consultar imágenes, nombres, precios simulados y descripciones básicas de videojuegos mediante APIs, así como realizar acciones de agregar o eliminar elementos del carrito sin necesidad de backend.

3. OBJETIVOS

- Proporcionar una guía para desarrolladores que requieran comprender la organización del proyecto, componentes visuales, scripts y estilos globales.
- Permitir una revisión estructurada de los módulos, favoreciendo la detección de errores, redundancias o necesidades de optimización.
- Garantizar consistencia visual y funcional en todas las secciones, utilizando patrones comunes de diseño e interacción.

4. PÚBLICO OBJETIVO

El sistema está destinado a personas interesadas en consultar información de videojuegos mediante una interfaz web. El usuario puede ser un aficionado a los videojuegos, un visitante casual o una persona que desea navegar un catálogo visual. También está orientado a personal técnico responsable de gestionar el sitio, agregar juegos o mantener la estructura.

No existen roles de usuario, autenticación, perfiles ni bases de datos. La interacción es libre, sin restricciones, y se centra únicamente en la navegación y visualización de contenido.

5. CONTEXTO DEL SISTEMA

El sistema surge como respuesta a la necesidad de disponer de una plataforma moderna y accesible para visualizar catálogos de videojuegos. Actualmente, la consulta de productos y entretenimiento digital se realiza principalmente desde aplicaciones web, lo que requiere interfaces que sean claras, visuales, rápidas y adaptables a distintos dispositivos.

KFCZONE implementa un modelo basado en navegación horizontal entre secciones enlazadas desde un encabezado global. La experiencia se centra en:

- consulta de listados
- exploración visual mediante tarjetas
- interacción mediante botones
- simulación del proceso de selección y carrito

El sistema no requiere servidor, bases de datos o procesamiento externo, ya que todas sus funciones trabajan del lado del cliente.

6. FLUJO GENERAL DEL SITIO WEB

El flujo de navegación comienza cuando el usuario accede a cualquier página. Desde la barra superior se despliega el menú principal con enlaces a:



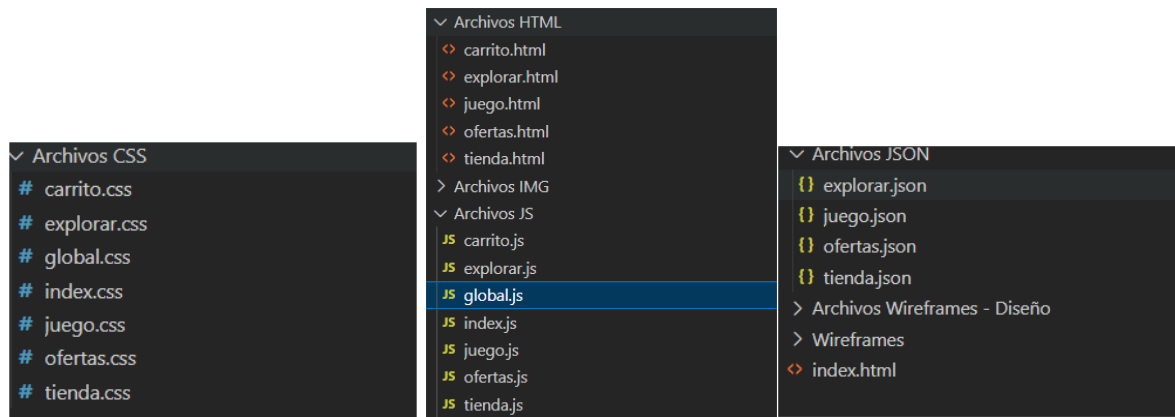
- Tienda
- Ofertas
- Explorar
- Juego
- Carrito

Al seleccionar una sección se carga una interfaz compuesta por tarjetas de videojuegos, botones de acción y elementos visuales. El usuario puede:

1. Navegar entre secciones
2. Visualizar listados
3. Abrir página individual de un juego
4. Agregar o quitar elementos del carrito
5. Volver al menú y seguir navegando

7. Estructura del Proyecto

El proyecto está organizado en carpetas y archivos independientes según funciones, manteniendo separación entre contenido, estilos, scripts y recursos visuales. La estructura general es:



Cada archivo HTML representa una pantalla del sistema. Las hojas de estilo determinan la apariencia visual. Los archivos JavaScript contienen la lógica de interacción, generación dinámica de tarjetas, comportamiento del menú y gestión del carrito y finalmente los archivos JSON guardan información de juegos en caso de que no se lograra conectar con APIs.

8. Desarrollo del Frontend

El desarrollo del frontend se realizó con tecnologías del lado del cliente:

- HTML para estructura semántica
- CSS para diseño visual y responsivo
- JavaScript para interacción

Las páginas utilizan etiquetas semánticas como header, main y footer. La navegación se gestiona mediante enlaces y menú hamburguesa en dispositivos móviles. Las tarjetas de juegos fueron generadas bajo un patrón común para mantener consistencia visual.

Los estilos incluyen:

- variables de color
- tipografías
- botones
- efectos hover
- layout mediante flexbox o grid
- sistema responsivo en base a media queries

Los scripts controlan:

- generación dinámica de contenido
- agregar y eliminar elementos del carrito
- acciones de botones
- animaciones de interfaz
- menú móvil

El sistema no requiere backend. Las acciones se procesan directamente en el navegador.

9. Despliegue del Sistema

El sistema puede ser desplegado en cualquier servicio de hosting web estático. Los archivos HTML, CSS, JS e imágenes pueden alojarse sin procesamiento de servidor. Es adecuado para:

- GitHub Pages
- Netlify
- Vercel
- Hostinger

El despliegue consiste en copiar todo el proyecto a un directorio público. El sistema está diseñado para funcionar sin configuraciones adicionales, ya que no requiere PHP, bases de datos.

El uso de un hosting estático presenta ventajas técnicas:

- no requiere servidor dedicado
- no existen dependencias o librerías externas
- se actualiza fácilmente trasladando archivos nuevos
- mantiene estructura clara en carpetas

10. DISEÑO

Se utilizaron colores oscuros predominantes con acentos en rojo para elementos interactivos como botones y precios. El sitio mantiene una identidad visual constante en todas las secciones.

10.1 Tipografía

Se utiliza tipografía sin serif moderna, adecuada para pantallas, con pesos diferenciados entre títulos, texto de botones y descripciones. La elección permite buena lectura en formatos reducidos como móviles.

10.2 Paleta de Colores

Los colores principales aplicados son:

- rojo intenso para elementos activos
- gris claro para textos secundarios
- fondo oscuro para mejorar contraste
- negro en áreas estructurales

Estos colores se aplican en:

- encabezado
- botones

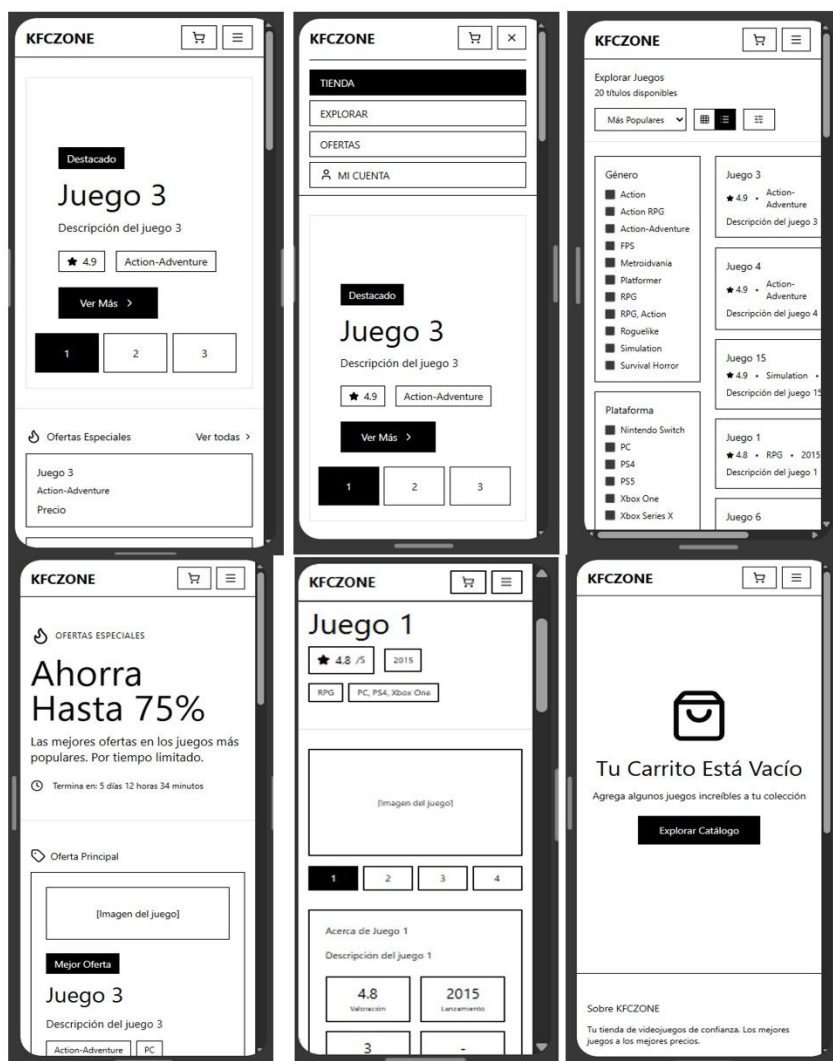
- tarjetas
- menús

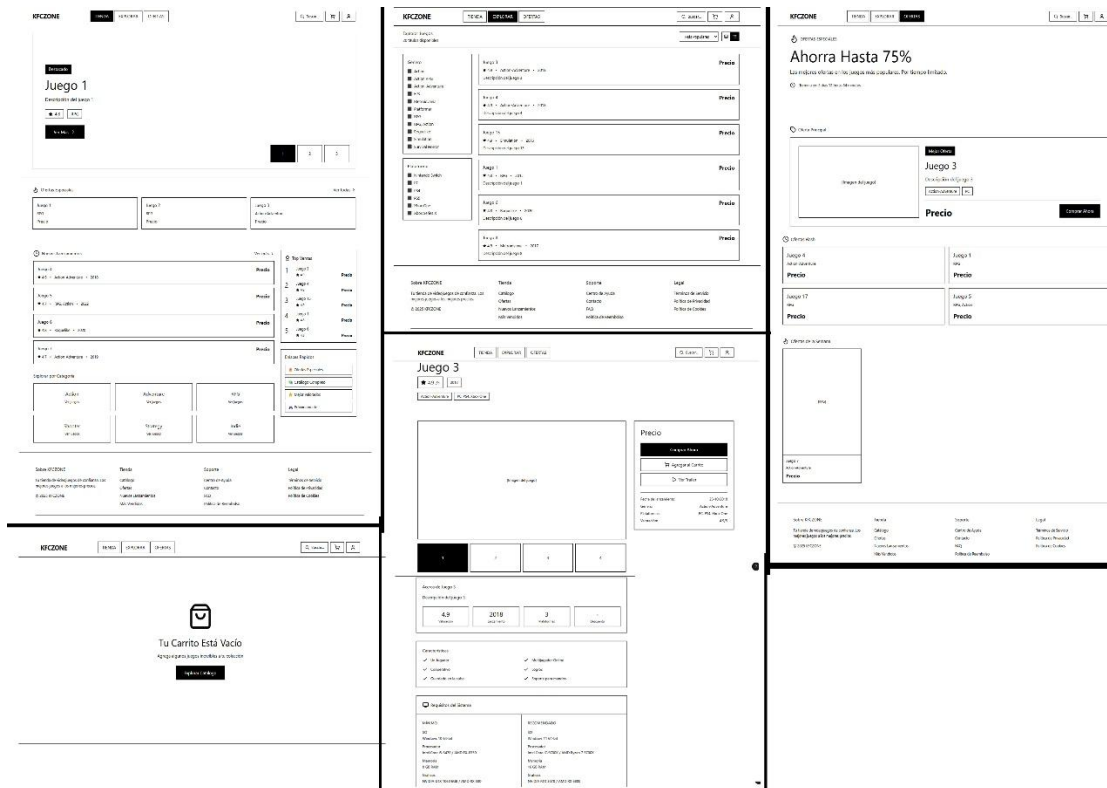
La combinación mantiene coherencia visual en todas las secciones.

10.3 Wireframes

El diseño fue conceptualizado en tres formatos:

- escritorio
- tablet
- móvil





Las versiones se diferencian por la disposición de columnas y visibilidad de menú. El menú hamburgués aparece únicamente cuando el ancho es reducido. Las tarjetas se adaptan mediante grid.

10.4 Responsividad

El sistema responde a cambios de tamaño mediante media queries:

- reorganización de columnas
- ajuste de fuente
- menú móvil
- botones escalables

Esto garantiza que el sitio pueda ser utilizado desde:

- computadoras
- tabletas
- teléfonos

```

/* 1024px */
@media (max-width:1024px){
  .contenedor{width:calc(100% - 30px);padding:16px 0}
  .encabezado .encabezado-interno{padding:16px 18px;gap:16px}
  .navegacion-principal{gap:10px}
  .enlace-nav{padding:7px 12px;font-size:13px}
  .buscar{width:160px;font-size:13px}
  .pie-grid{grid-template-columns:repeat(3,1fr)}
}

/* 680px */
@media (max-width:680px){
  .contenedor{width:calc(100% - 20px);padding:14px 0}
  .encabezado .encabezado-interno{padding:14px 14px;gap:14px}
  .logo-KFCZONE{height:24px}
  .navegacion-principal{right:12px;top:64px;width:160px}
  .navegacion-principal a{padding:9px;font-size:13px}
}

```

sin pérdida de funcionalidad.

11. ESTILOS Y SCRIPTS GLOBALES

El sistema KFCZONE utiliza un conjunto de estilos globales que se aplican en todas las secciones del sitio. Estos estilos se definen de forma centralizada para mantener uniformidad visual, simplificar mantenimiento y reducir redundancia en los archivos CSS individuales.

Los estilos globales cumplen tres funciones principales:

- normalización visual del documento
- definición de variables reutilizables
- construcción de estructura compartida (encabezado, navegación, pie de página, menú móvil)

Junto a los estilos se implementa un script global que controla el comportamiento del menú hamburguesa utilizado en pantallas pequeñas.

11.1 Reset del Documento

Antes de aplicar estilos específicos se realiza una normalización del comportamiento de todos los elementos. El reinicio incluye:

- margin: 0
- padding: 0
- box-sizing: border-box
- altura completa para html y body

Este enfoque evita inconsistencias entre navegadores, y permite que los tamaños calculados incluyan bordes y relleno de forma predecible.

El body establece:

- tipografía base
- color general del texto
- fondo uniforme

- suavizado de fuentes para mejorar legibilidad

11.2 Variables Globales

Se definen variables CSS mediante :root, lo cual centraliza valores comunes:

- ancho máximo del contenedor
- tamaño base
- colores: primario, secundario, terciario, cuaternario
- radio de borde

Estas variables permiten mantener consistencia y facilitan ajustes globales del diseño visual.

Una modificación de color o tamaño se refleja automáticamente en todas las secciones, eliminando necesidad de repetir estilos locales.

11.3 Contenedor

La clase contenedor se utiliza en encabezados, secciones y pies de página. Controla el ancho utilizable con margen lateral y centrado automático.

Esto garantiza que el contenido esté visualmente alineado sin importar la resolución de pantalla.

11.4 Encabezado y Navegación

El encabezado es un componente estructural que contiene:

- logotipo
- navegación principal
- zona de acciones

En pantallas grandes se muestra horizontal.

Los enlaces del menú tienen:

- borde
- fondo transparente
- color secundario
- transición con escala y cambio de color en hover
- estado activo con color primario

El encabezado utiliza flexbox con separación mediante gap.

11.4.1 Zona de Acciones

La zona de acciones incluye:

- campo de búsqueda
- botones iconográficos

Los botones utilizan:

- borde
- radio estándar
- transición
- escala visual en hover

Todo el diseño es coherente con la paleta general.

11.5 Pie de Página

El pie de página se compone mediante grid:

- cuatro columnas en escritorio
- dos columnas en móviles

Incluye enlaces, información y enlaces hover con animación de escala. Los títulos tienen tipografía clara y color blanco.

Este patrón garantiza una estructura clara y uniforme para elementos informativos.

11.6 Estilos Responsivos

Los componentes globales incorporan comportamientos responsivos basados en media queries.

Cuando la pantalla está por debajo de 768px:

- aparece el botón hamburguesa
- se oculta la navegación principal
- el menú se despliega mediante click
- los enlaces se reorganizan verticalmente

El menú desplegado se posiciona sobre el contenido mediante position: absolute, asegurando visibilidad y accesibilidad.

En 640px o menos la grilla del pie de página pasa a dos columnas para mejorar lectura.

11.7 Menú Hamburguesa

El menú móvil es un elemento fundamental para pantallas reducidas. El botón hamburguesa se compone mediante tres líneas horizontales creadas con elementos span de ancho y alto fijo.

Cuando el usuario interactúa, el menú se muestra u oculta según estado de visualización.

11.8 Script Global

La lógica se implementa en un archivo js global, con referencias a:

```
// boton hamburguesa
const hamburguesa = document.getElementById("hamburguesa");
const nav = document.querySelector(".navegacion-principal");
hamburguesa.addEventListener("click", () => {

  // alternar si el menu se muestra o se oculta
  nav.style.display = nav.style.display === "flex" ? "none" : "flex";
});
```

No requiere librerías externas, frameworks ni dependencias.

El script solo controla visibilidad, no afecta navegación ni rutas internas. La interacción es simple, directa y suficiente para las necesidades del sistema.

11.9 Implementacion de APIs

GitHub Pages hospeda el HTML/CSS/JS, y APIs que están en servidores externos, el sitio puede hacer fetch() y funciona de forma perfecta.

AWG: <https://api.rawg.io/api>

YouTube: [https://www.googleapis.com/youtube/v3/...](https://www.googleapis.com/youtube/v3/)

```
// config rawg y youtube
const RAWG_KEY = "66d71aea9962478a92839e951481b374";
const RAWG_BASE = "https://corsproxy.io/?https://api.rawg.io/api";
const YT_KEY = "AIzaSyBxKNB4izVzKqU4rznBLvZYChbUYOgfVL4";
```

11.10 Consideraciones Técnicas

Los estilos globales cumplen con criterios de:

- consistencia
- mantenimiento
- escalabilidad

Su uso permite:

- reducir duplicación de código
- actualizar diseño de forma centralizada
- asegurar compatibilidad entre páginas

No existen reglas innecesarias, duplicadas o conflictivas, debido a la distribución modular:

- cada sección tiene su CSS propio

- el global establece base y estructura

Los estilos individuales extienden el global sin sobrecribir comportamiento central.

12. CONCLUSIONES

El desarrollo del sistema KFCZONE permitió integrar una plataforma funcional orientada a la visualización de videojuegos, aplicando principios de arquitectura web estática, diseño responsivo y organización modular de archivos. El uso de HTML, CSS y JavaScript fue adecuado al alcance del proyecto, permitiendo interacción básica, generación dinámica y navegación clara, sin necesidad de backend.

La estructura organizada del proyecto facilita:

- mantenimiento
- escalabilidad
- incorporación de nuevos juegos
- modificaciones visuales

El sistema cumple con el objetivo de ofrecer una interfaz accesible y coherente para navegación y consulta de contenido, siendo ideal para demostración académica o proyectos orientados a catálogo visual.

13. BIBLIOGRAFÍA

- Diseño y creación de wireframes
- <https://www.figma.com/design/3Bw79bOiiiaTdK89E7ntOT/Proyecto-Web?node-id=01&t=0qGmEZXYi8izkTRb-1>
- W3Schools.com. (s. f.). <https://www.w3schools.com/cssref/index.php>
- W3Schools.com. (s. f.-c). <https://www.w3schools.com/tags/default.asp>