UNIVERSIDAD DEL VALLE

MANUAL TECNICO DE VIDEOPLUS

Integrantes:

Chura Condori Wilder

Docente:

Ing. Henry Miranda Ordonez

Asignatura:

Programación Web I

Grupo:

“A” Segundo Semestre

La Paz-Bolivia

Versión 1.0.0

Manual Técnico del Proyecto VideoPlus

1. Introducción

VideoPlus es una plataforma web para explorar series y animes en línea. Permite a los usuarios navegar por contenido actualizado, buscar títulos específicos, ver detalles de cada anime y reproducir capítulos directamente desde el navegador. Cuyo público objetivo son usuarios ver series y animes de forma rápida.

2. Objetivos

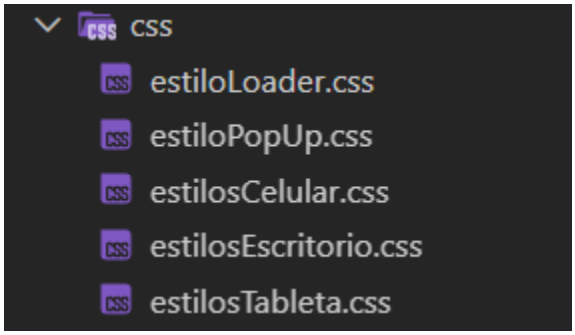
- Ofrecer una interfaz interactiva y responsiva para la visualización de series y animes.
- Proporcionar funcionalidades como búsqueda en tiempo real, filtrado por tipo, género, temporada y estado.
- Permitir la gestión básica de usuarios mediante login y registro local.

3. Tecnologías utilizadas:

- HTML: Utilizado para armar la estructura de nuestra página de manera semántica.
- CSS: Utilizado para dar los estilos y la responsividad del Proyecto mediante Queries..
- JavaScript: Nos ayudó en el manejo del DOM de manera dinámica.
- JIKAN: API externa, que nos proporciona las series y animes actualizados hasta el momento.
- Almacenamiento: LocalStorage para gestión de usuarios

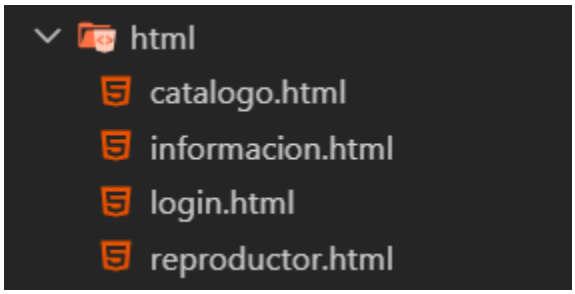
4. Gestion de archivos del Sistema

La gestión de archivos del proyecto esta organizado según tipo y funcionalidad



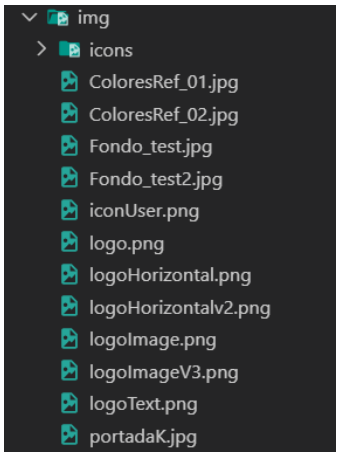
Carpeta CSS: En esta carpeta están los estilos utilizados:

- EstiloLoader: Contiene los estilos del circulo de carga mientras se espera la información.
- EstiloPopUp: Marca el estilo de popUp cuando se nos confirma algún formulario, como el de inicio de sesión o registro
- EstiloCelular: Se realiza la parte responsiva para el móvil
- EstiloEscritorio: Estilo principal de la página adaptada para navegadores de escritorio.
- EstiloTableta: Se realiza la parte responsiva para las tabletas.

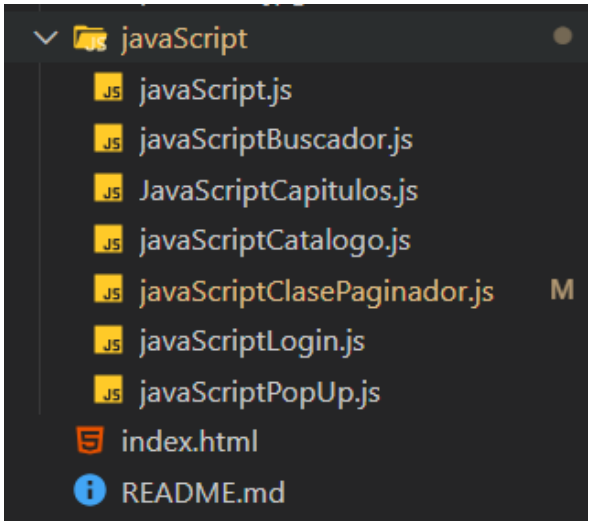


En la carpeta HTML se encuentran las paginas por las cuales navegara el usuario:

- Catalogo: Muestra todas las series en una página, para que el usuario pueda tener mayor visibilidad de las series disponibles
- Información: Brinda información de la serie que escogió el usuario, tanto como su titulo, descripción y capítulos
- Login: Pagina usada para el inicio de sesión o registro de un usuario
- Reproductor: Brinda la posibilidad de reproducir la serie o cambiar entre capítulos



La carpeta img es utilizada para tener las imágenes que se utilizaran en el proyecto



La carptera de JavaScript es para la lógica, uso de la API Jikan y manejo del DOM:

- JavaScript: realiza la lógica principal de cargar los animes top y recomendaciones.
- JavaScriptBuscador: Realiza la lógica para el buscador, tanto para mostrar los resultados o no.
- JavaScriptCapitulos: Se usa para la obtención y seccionamiento de los capítulos que el usuario selecciono ver
- JavaScriptCatalogo: Usado para mostrar el catalogo de todos los animes disponibles.
- JavaScriptClasePaginador: Clase base para generalizar las funciones de crear las portadas, realizar la lógica de los paginadores y actualización de la lista de series.
- JavaScriptLogin: Se realiza la lógica de autenticación, tanto del incio de sesión y el registro de manera local.
- JavaScriptPopUp: Logica para la aparición de un popup, con la posibilidad de aumentar mas funciones a su botón.

En la carpeta raíz también encontramos dos archivos:

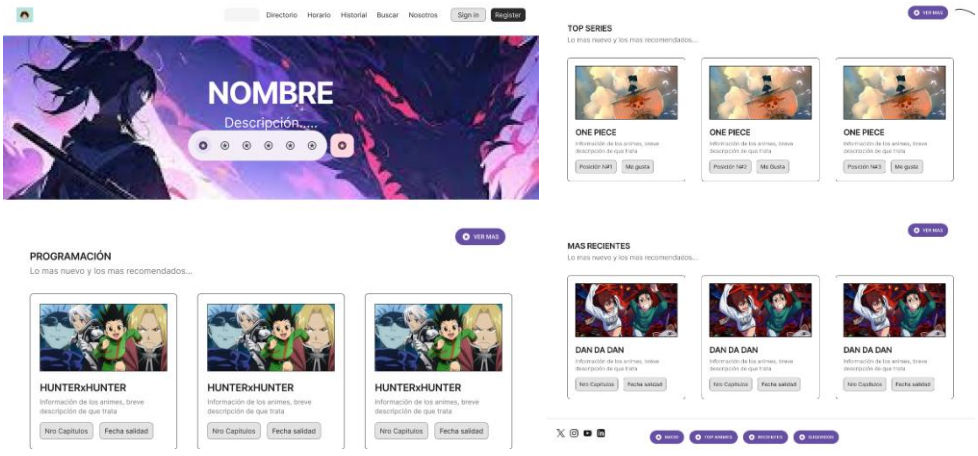
- Index: Que es la página principal, estando afuera para que GitHub Page lo reconozca y lo publique.
- README: Archivo con especificaciones y descripcion referente al proyecto.

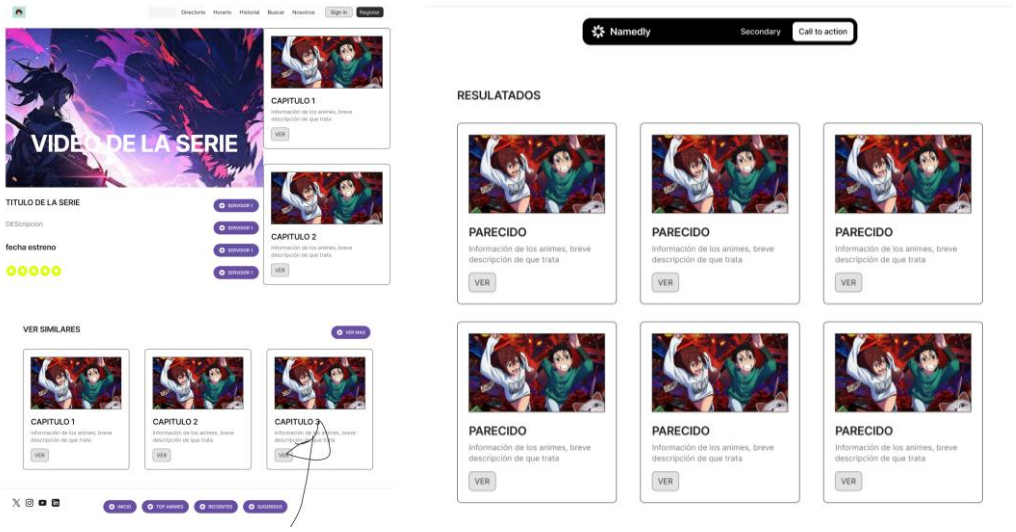
5. Flujo simple de la aplicación:

1. El usuario ingresa a la página principal.
2. Se cargan las tarjetas de series/animes usando la API de Jikan.
3. El usuario puede buscar, filtrar y paginar el contenido.
4. En caso de que presione ver más, podrá ver el catálogo de los animes actuales
5. Al seleccionar un anime, se muestra la información detallada y el listado de capítulos.
6. Al seleccionar el capítulo que desea ver se le pasara al reproductor.

6. Diseño de Interfaces

Prototipo de la interfaz de usuario





El prototipo nos ayuda a ver dónde ira cada elemento de manera eficiente o cuales podemos borrar, así mismo darnos un vistazo para la parte responsiva y como lo estructuramos.

La paleta de colores utilizada en este caso son los siguiente:

```
    --color-primario: #162936;
    --color-secundario: #3b5265;
    --color-terciario: #27e9b5;
    --color-terciario-A50: #27e9b58b;
    --color-cuarternario: #051824;

    --color-background-section: #062335;
```

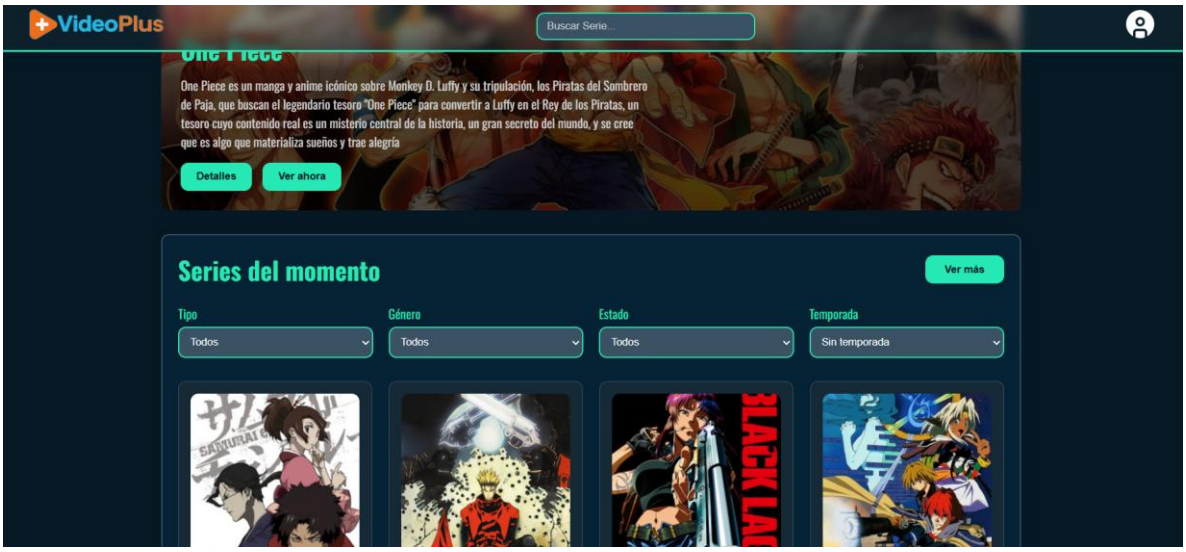
Son constantes para en un futuro por arreglarlo o cambiarlos

Tipografía y estilos

La tipografía que utilizamos es la siguiente de GoogleFonts: Oswald

```
@import
url('https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100..90
0;1,100..900&family=Mulish:ital,wght@0,200..1000;1,200..1000&family=Oswald:w
ght@200..700&display=swap');
```

- Tamaño de títulos: 1.2rem – 1.5rem
- Tamaño de subtítulos: 0.85rem – 1rem
- Bordes redondeados y sombras suaves en tarjetas y botones
- Animaciones de hover y transiciones para botones y tarjetas

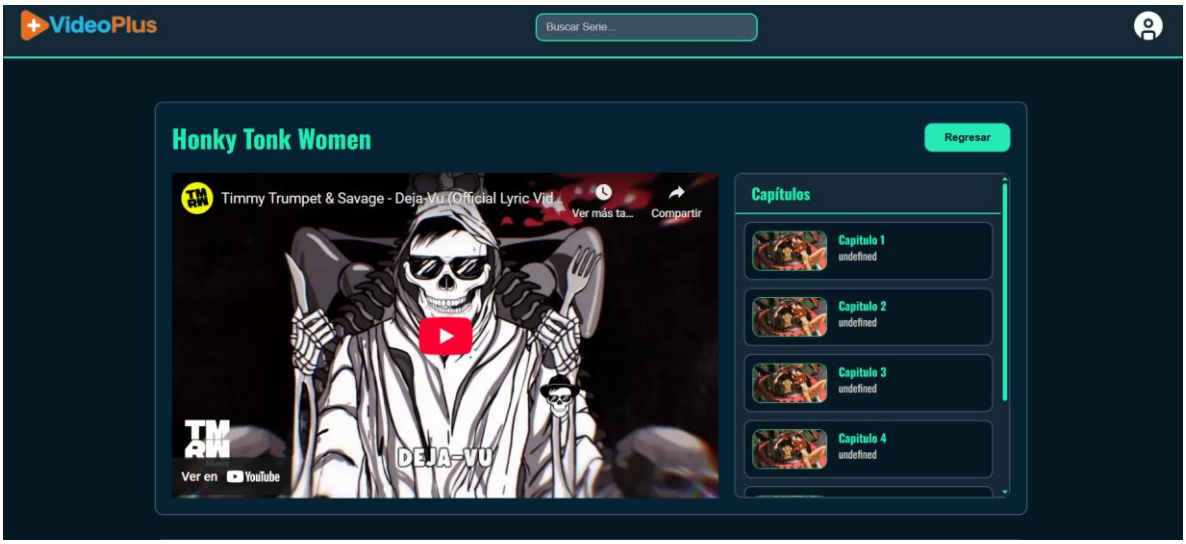


Página principal dividido en tres secciones:

- Sección destacada: donde veremos que serie es la mas vista, su descripción y sus opciones de ver o tener mas información
- Sección series del momento: mostramos que series están siendo nuevas o mas vista, teniendo la posibilidad de filtrar según el contenido que le guste
- Seccion Top Series: mostrarala las series que tienen mayor candidad de usuarios viéndola, cada tarjeta tiene su nombre y descripción

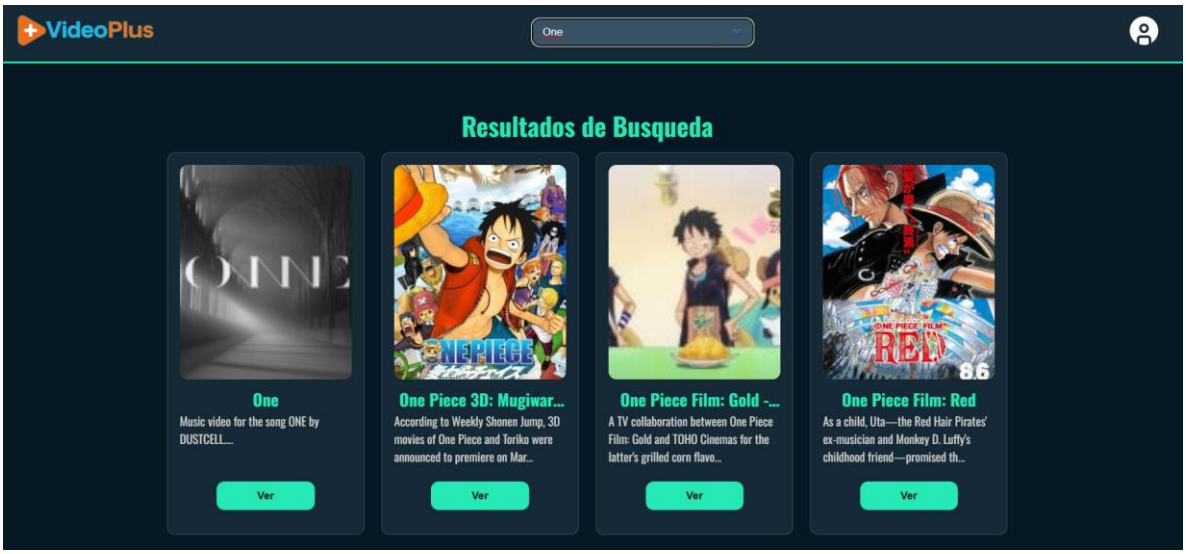
La pagina de información esta seccionado de la siguiente manera:

- Seccion Principal: donde se ve información mas detalla de la serie
- Seccion Capitulos: Muestra la cantidad de capítulos que se tiene
- Seccion Similares: Se enlista las series que son similares a la serie elegida

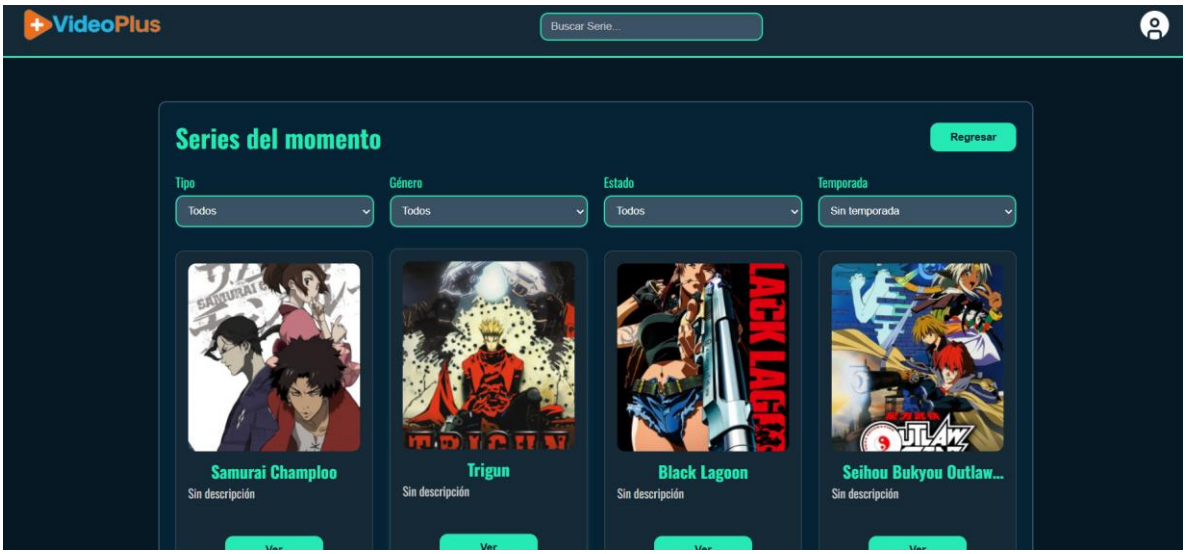


La siguiente pagina es del reproductor:

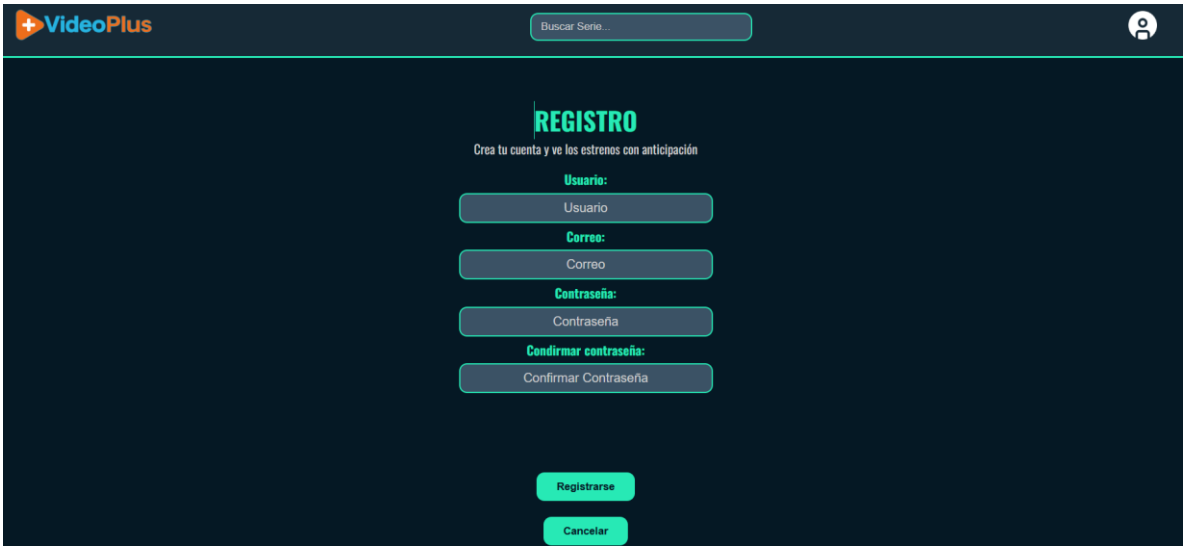
- Sección reproductor: donde vera la serie del lado izquierdo y en el derecho los capítulos si se desea saltar
- Sección Similares: Se enlista las series que son similares a la serie elegida



En la pagina de búsqueda, el usuario puede ingresar el nombre y se ira actualizando la lista de manera dinámica.



La sección de catalogo: muestra todas las series disponibles, conjuntamente con los filtros si se busca algo en especifico.



En la pagina de iniciar sesión y registro:

- Se tiene los inputs tanto en del login y registro para validarlos desde JavaScript, conjuntamente con la opción de cancelar el registro.

7. Componentes y Funcionalidades

Login / Registro

- Formulario de ingreso con validación de correo y contraseña.
- Guardado de usuarios en LocalStorage.
- Mensajes de error y confirmación mediante popup personalizado.

Buscador

- Busca animes en tiempo real usando Jikan API.
- Limitación de peticiones con debounce de 400ms.
- Muestra resultados en tarjetas dinámicas dentro de un contenedor.

Tarjetas Dinámicas

- Se crean con información obtenida de la API: imagen, título, sinopsis, botón “Ver”.
- Responsive: se adaptan en grid o flex según pantalla.
- Sincronización con loader mientras se cargan los datos.

Listado de Capítulos

- Panel lateral con scroll, listado de capítulos con número, título y miniatura.
- Título fijo en la parte superior (position: sticky).
- Compatible con pantallas móviles mediante diseño vertical.

Loader y animaciones

- Loader clásico circular en el centro del contenedor mientras se cargan tarjetas.
- Efectos de hover en tarjetas y botones.

Diseño responsive

- Uso de flexbox y media queries para adaptarse a móviles y tablets.
- Ajuste de grid de tarjetas según ancho de pantalla.
- Botón “Ver más” centrado en escritorio y a la derecha en móvil.

Resumen de las funcionalidades específicas en general

- Login / Registro con LocalStorage
- Buscador con API de Jikan (con paginación y filtros)
- Creación de tarjetas dinámicas
- Listado de capítulos y reproductor de videos
- Loader / animaciones de carga
- Responsive design (adaptación a móviles y tablets)
- Validaciones de formularios

8. Uso de la API Jikan

Jikan API:

Jikan (時間) es una API no oficial y de código abierto para la “comunidad y base de datos de anime y manga en línea más activa” : MyAnimeList.

Las principales funciones y usos que se los dio fueron:

- Endpoint principal: https://api.jikan.moe/v4/top/anime?page=1
- Permite obtener listas de animes, información detallada, capítulos, recomendaciones y búsqueda.
- Parámetros disponibles:
    - q: texto de búsqueda
    - type: tipo de anime (tv, movie, ova, etc.)
    - genre: género
    - status: estado (airing, completed, upcoming)
    - season: temporada y año
- Manejo de errores: mostrar mensaje si la API falla o hay muchas peticiones simultáneas.
- Peticiones:
    - Para obtener la lista de series recomndadas es la siguiente, donde animeID, es el ID de la serie que deseamos

```
https://api.jikan.moe/v4/anime/${animeId}/recommendations
```

    - Para obtener la lista de las mejores series se usa la siguiente petición.

```
`https://api.jikan.moe/v4/top/anime?page=1`
```

    - Para tener el listado de series, bajo un filtro tenemos una petición simple, pero a ese tenemos que mandarle los parámetros que deseamos y lo segmentara.

```
let url = "https://api.jikan.moe/v4/anime?limit=20";
        if (tipo) url += `&type=${tipo}`;
        if (genero) url += `&genres=${genero}`;
        if (temporada) url += `&season=${temporada}`;
        if (estado) url += `&status=${estado}`;
```

    - En cuanto a las peticiones para búsqueda se realiza lo siguiente, donde mandaremos un query, es decir, los que deseamos buscar.

```
`https://api.jikan.moe/v4/anime?q=${query}&limit=12`
```

9. Códigos más utilizados y de guía

Creacion de botones para la paginación, según la cantidad que tengamos, donde:

- listBtnPages: guarda los botones, en caso de actualizar la lista se tiene que eliminar sus elementos y volverlos a crear
- listAnimesTop: Lista donde tenemos las series buscadas
- containerPages: contendor de donde estarán los botones

```
createButtonPages() {
        this.listBtnPages.forEach(btn => {
            if (this.containerPages) this.containerPages.removeChild(btn);
        });

        this.listBtnPages = [];
```

```
        for (let i = 0; i < this.listAnimesTop.length; i++) {
            let newBtn = document.createElement('button');
            newBtn.classList.add('btn-pagina');
            newBtn.textContent = i + 1;

            newBtn.addEventListener('click', () => {
                this.currentPage = i;
                this.showAnimePage(i);
            });

            this.listBtnPages.push(newBtn);

            if (this.containerPages)
this.containerPages.insertBefore(newBtn, this.btnNext);
        }
    }
```

Realizando peticiones, usamos un método asíncrono, es decir, trabaja al mismo tiempo:

- fetch: usamos este método que es para crear peticiones a las APIs que requerimos.
- await: detiene la ejecución en ese punto, donde esperara hasta que reciba una respuesta o algun fallo.
- try, cath y finally: los estamos usando por las posibilidades que la API o conexión pueden llegar a fallar, como también lo usamos para identificar si nuestra petición fue exitosa o fracaso.

```
    async GetAnimes() {
        try {
            this.showLoader();
            // const url = `https://api.jikan.moe/v4/top/anime?page=1`;
            const resp = await fetch(this.url);
            const data = await resp.json();
            console.log(data.data);
            this.createAnimeList(data.data);
        } catch (error) {
            console.error("Error al obtener recomendados:", error);
            if (this.containerSerie) this.containerSerie.innerHTML = "<p
style='color:red;'>Error al cargar los animes</p>";
        } finally {
            this.hideLoader();
        }
    }
```

Para creación de capítulos en la parte derecha y de manera vertical, usamos la concatenación para crear y contenido según necesitemos.

```
createVerticalChapters(image_url) {
```

```
        if (this.containerVertical) {
            let chapterNum = 1;
            this.listAnimesTop.forEach(element => {
                let newElement = `
                <a href="reproductor.html">
                    <li class="capitulos">
                        <img src="${image_url}">
                        <div>
                            <span class="numero-capitulo">Capitulo
${chapterNum}</span>
                            <span class="titulo-
capitulo">${element.title}</span>
                        </div>
                    </li>
                </a>
            `;
                this.containerVertical.innerHTML += newElement;
                chapterNum++;
            });
        }
    }
```

Función para buscar, si bien podemos realizar la búsqueda en tiempo real, la API Jikan, solo permite 3 peticiones simultaneas, por lo cual tenemos que enviarlo con intervalos de tiempo, como es SendaSearch usa la función de setTiemeout, 1000, lo cual limitaremos el numero de peticiones en un intervalo de 1s

```
function SendSearch(searcher){
    clearTimeout(time);
    time = setTimeout(() => {
        hacerBusqueda(searcher.value.trim());
    }, 1000);
}
```

Para evitar desbordamiento en cuanto al texto, - webkit nos brinda funcionalidades propios del navegador, en este caso permitiendo que se desfase y oculte al mismo tiempo.

```
.tarjeta-serie p {
    display: -webkit-box;
    -webkit-box-orient: vertical;
    overflow: hidden;
    text-overflow: ellipsis;
}
```

La estructura semántica que se tiene en cuanto al html es :

- Head: donde agregamos los estilos que necesitamos, más los ajustes para que los primeras primejos ajustes son del icono y nombre de la web.

```
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>VideoPlus | Anime Online</title>
    <link rel="icon" href="img/logoImageV3.png" type="image/x-icon">
    <link rel="stylesheet" href="css/estilosEscritorio.css">
    <link rel="stylesheet" href="css/estilosTableta.css">
    <link rel="stylesheet" href="css/estilosCelular.css">
    <link rel="stylesheet" href="css//estiloLoader.css">
</head>
```

- Header: Se tiene doble navegadores uno normal donde podrá usar buscador tanto en el escritorio y tablets, caso que sea móvil se desactiva la navegación horizontal para habilitar la navegación vertical que tiene otro orden, donde el buscador se encuentra en la parte inferior de logo y el icono de usuario.

```
<header>
        <nav id="navegacion">
            <div id="nav-horizontal">
                <div class="logo">
                    <a href="../index.html">
                        <img src="../img/logoHorizontalv2.png" alt="Logo de
VideoPlus">
                    </a>
                </div>
                <form class="buscador" role="search">
                    <input id="buscador" class="campoLlenar" type="search"
name="buscar" placeholder="Buscar Serie...">
                </form>
                <div class="iconUser">
                    <a href="login.html">
                        <img src="../img/iconUser.png" alt="Icono usuario">
                    </a>
                </div>
            </div>

            <div id="nav-vertical">
                <div class="nav-superior">
                    <div class="logo">
                        <a href="../index.html">
                            <img src="../img/logoHorizontalv2.png" alt="Logo
de VideoPlus">
                        </a>
                    </div>
                    <div class="iconUser">
                        <a href="login.html">
```

```
                        <img src="../img/iconUser.png" alt="Icono
usuario">
                    </a>
                </div>
            </div>
            <form class="buscador" role="search">
                <input id="buscadorMovile" class="campoLlenar"
type="search" name="buscar"
                    placeholder="Buscar Serie...">
            </form>
        </div>
    </nav>
</header>
```

- Main: En cada main tenemos diferentes secciones donde las mas importante es del contenido de búsqueda ya que, tenemos que manipular y saber cuando el usuario esta buscando o simplemente esta navegando, donde tendríamos que ocular las demás secciones, al momento de su consulta o mostrar las secciones principales cuando el usuario no está realizando ninguna búsqueda.
- Section: en cada sección en su mayoría está conformado por el siguiente
    - H3: titulo de la sección
    - Div: aunque es un contenedor y no una etiqueta semántica, lo usamos como el cuerpo donde estarán las tarjetas.
    - Button: en caso de que se requiera ver más series.
- Form filtros: utilizamos el form para contener los filtros
    - Option: Cada filtro tendrá una cantidad diferente de opciones y cada opción un valor que se asocia a los filtros de la API.

```
<main>
    <section id="contenido-busqueda" class="ocultar-menu">
        <h2 id="top-series-titulo">Resultados de Busqueda</h2>
        <div id="contenedorBusqueda-series" class="grid-series">
        </div>
        <div id="contenedor-pagina-busqueda" class="grid-paginas">
            <button id="prev" class="btn-pagina">&#10216</button>
            <!-- <button class="btn-pagina pagina-activa">1</button> -->
            <button id="next" class="btn-pagina">&#10217</button>
        </div>
    </section>
    <section class="contenido-principal seccion-destacados">
        <h2 id="titulo-destacado" class="titulo-seccion">Serie
destacada</h2>
        <article class="contenedor-portada">
            <img src="img/Fondo_test2.jpg" alt="Portada de la serie
destacada">
            <div class="texto-superpuesto">
```

```
                    <h3>One Piece</h3>
                    <p>
                        One Piece es un manga y anime icónico sobre Monkey
D. Luffy y su tripulación, los Piratas del
                    </p>

                    <div class="acciones-portada">
                        <a
href="html/informacion.html"><button>Detalles</button></a>
                        <a href="html/reproductor.html"><button>Ver
ahora</button></a>
                    </div>
                </div>
            </article>
        </section>

        <section class="contenido-principal seccion-series">
            <div class="titulo-seccion-contenedor">
                <h2 id="top-series-titulo">Series del momento</h2>
                <a href="html/catalogo.html"><button class="btn-vermas">Ver
más</button></a>
            </div>
            <form class="form-filtros">
                <div class="campo-filtro">
                    <label for="tipo">Tipo</label>
                    <select name="tipo" id="tipo">
                        <option value="">Todos</option>
                        <option value="tv">Series</option>
                        <option value="movie">Películas</option>
                        <option value="ova">Ovas</option>
                        <option value="special">Especiales</option>
                        <option value="ona">Ona</option>
                        <option value="music">Musica</option>
                    </select>
                </div>
                <div class="campo-filtro">
                    <label for="genero">Género</label>
                    <select name="genero" id="genero">
                        <option value="">Todos</option>
                        <option value="1">Accion</option>
                        <option value="2">Aventura</option>
                        <option value="4">Comedi</option>
                        <option value="8">Drama</option>
                        <option value="10">Fantasia</option>
                        <option value="22">Romance</option>
```

```
                        <option value="24">Sci-Fi</option>
                        <option value="27">Shounen</option>
                        <option value="36">Dia a dia</option>
                        <option value="37">Supernatural</option>
                    </select>
                </div>

                <div class="campo-filtro">
                    <label for="estado">Estado</label>
                    <select name="estado" id="estado">
                        <option value="">Todos</option>
                        <option value="airing">En emisión</option>
                        <option value="complete">Completado</option>
                        <option value="uncoming">Proximamente</option>
                    </select>
                </div>

                <div class="campo-filtro">
                    <label for="temporada">Temporada</label>
                    <select name="temporada" id="temporada">
                        <option value="">Sin temporada</option>
                        <option value="winter">Invierno</option>
                        <option value="spring">Primavera</option>
                        <option value="summer">Verano</option>
                        <option value="fall">Otoño</option>
                    </select>
                </div>
            </form>
            <div id="contenedor-recomendados" class="grid-
series">
            </div>
            <div id="paginas-recomendadas" class="grid-paginas">
                <button id="previo-recomendado" class="btn-
pagina">&#10216</button>
                <button id="next-recomendado" class="btn-
pagina">&#10217</button>
            </div>
        </section>

        <section class="contenido-principal seccion-series">
            <div class="titulo-seccion-contenedor">
                <h2 id="top-series-titulo">Top Series</h2>
                <a href="html/catalogo.html"><button class="btn-vermas">Ver
más</button></a>
            </div>
```

```
            <div id="contenedor-series-top" class="grid-series">
            </div>
            <div id="paginas-top" class="grid-paginas">
                <button id="previo-top" class="btn-pagina">&#10216</button>
                <button id="next-top" class="btn-pagina">&#10217</button>
            </div>
        </section>
    </main>
```

- Footer: Utilizamos el footes para poner nuestras redes sociales y los respectivos links, esto hay que tomar en cuneta ya que se utiliza en la en todas las paginas, por lo cual se suele estar copiando de un apagina a otra.

```
<footer class="estilo-footer">
        <p>© 2025 VideoPlus - Todos los derechos reservados</p>

        <div class="sociales">
            <a href="https://www.facebook.com/share/16j6dpiWxN/"><img
src="img/icons/facebook.png" alt="Facebook"></a>
            <a href="https://github.com/perezitoPike"><img
src="img/icons/github.png" alt="GitHub"></a>
            <a href="https://www.linkedin.com/in/wilder-chura-
85812922b"><img src="img/icons/linkedin.png" alt="LinkedIn"></a>
            <a href="https://github.com/perezitoPike"><img
src="img/icons/twitter.png" alt="Twitter"></a>
            <a href="https://wa.me/+59169789500"><img
src="img/icons/whatsapp.png" alt="WhatsApp"></a>
        </div>
    </footer>
```

10. Glosario

Término	Significado
API	Servicio que devuelve datos desde un servidor.
Jikan API	API pública que permite obtener información de MyAnimeList.
Loader	Animación que indica que algo está cargando.
Paginador	Control para cambiar de página en una lista de resultados.
Tarjeta	Pequeño panel visual con imagen y datos resumidos.
Modal / Popup	Ventana emergente con información detallada.
LocalStorage	Forma de guardar datos de manera local.

11. Algunos usos y onsideraciones técnicas

- Limitar las peticiones a la API con un temporizador, en nuestro caso utilizamos un timer de 1 segundo, para no sobre saturarlo de peticiones.
- Uso de LocalStorage solo para pruebas locales, que en nuestro caso usamos con los inputs y sino pasa las restricciones se le informa mediante alerts.

- Animaciones suaves y transiciones para mejorar UX.
- Variables CSS para consistencia en colores y estilos.

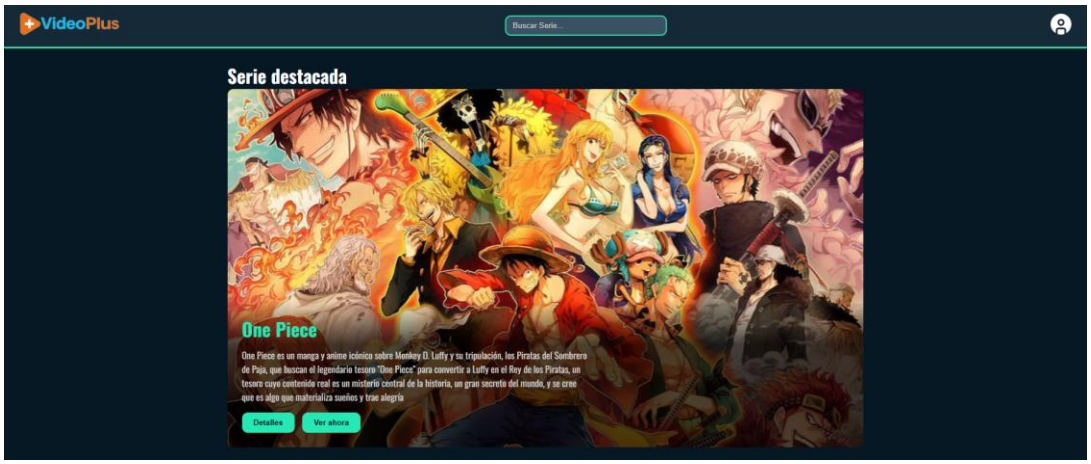
12. Conclusiones

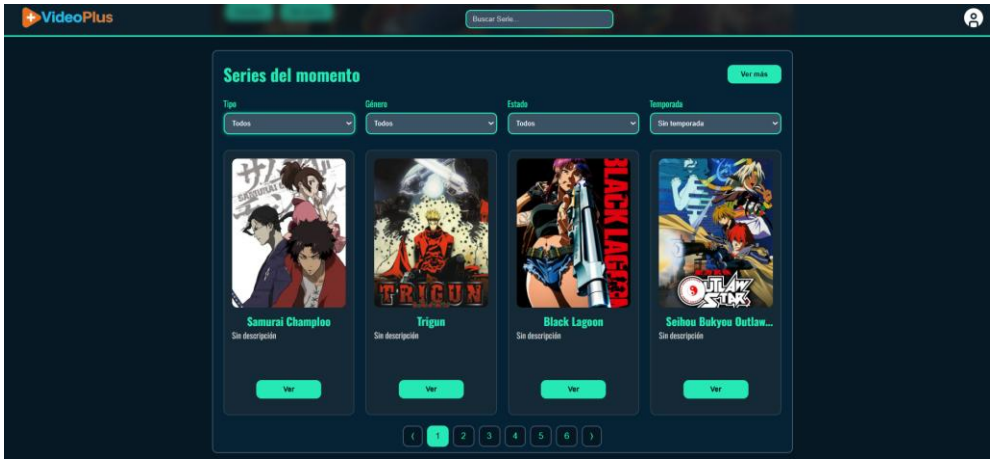
VideoPlus es un proyecto completo de visualización de series y animes que combina diseño responsivo, integración con API externa, y gestión básica de usuarios.

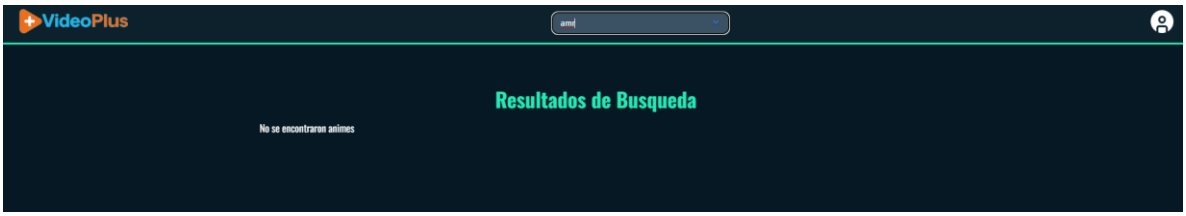
- Logros: diseño moderno, paginación, búsqueda en tiempo real, listado de capítulos, loader dinámico.
- Limitaciones: almacenamiento local solo para pruebas, dependencia de Jikan API.
- Próximas mejoras: autenticación con backend, favoritos de usuario, filtros avanzados, integración con otras APIs.

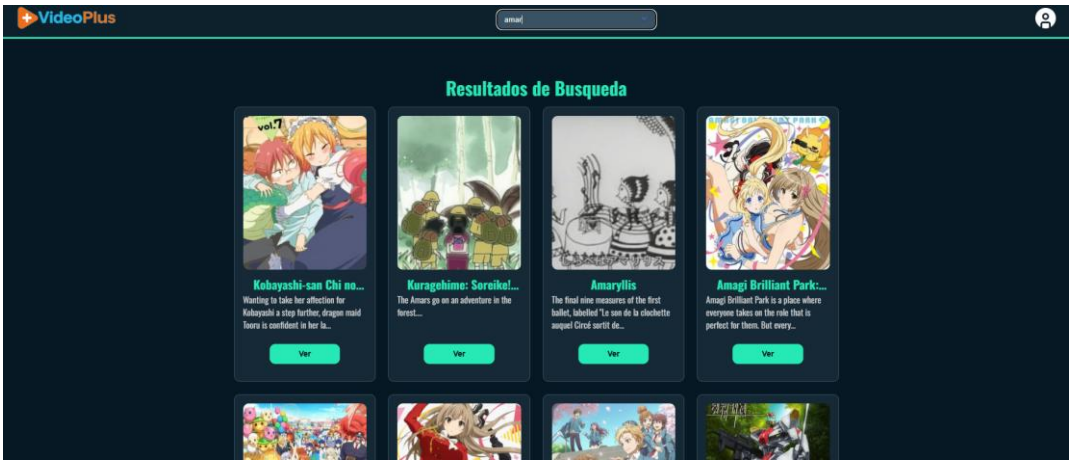
13. Anexos

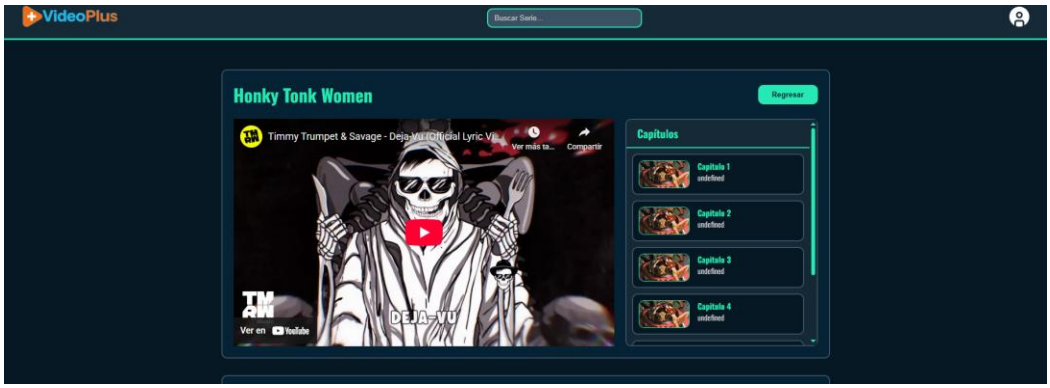
Paginas principales


VideoPlus
Buscar Serie...
Serie destacada
One Piece
One Piece es un manga y anime icónico sobre Monkey D. Luffy y su tripulación, los Piratas del Sombrero de Paja, que buscan el legendario tesoro "One Piece" para convertir a Luffy en el Rey de los Piratas, un tesoro cuyo contenido real es un misterio central de la historia, un gran secreto del mundo, y se cree que es algo que materializa sueños y trae alegría
Detalles
Ver ahora


VideoPlus
Buscar Serie...
Series del momento
Ver más
Tipo
Todos
Género
Todos
Estado
Todos
Temporada
Sin temporada
Samurai Champloo
Sin descripción
Ver
Trigun
Sin descripción
Ver
Black Lagoon
Sin descripción
Ver
Seihou Bukyou Outlaw...
Sin descripción
Ver
1 2 3 4 5 6


VideoPlus
amq
Resultados de Busqueda
No se encontraron animes


VideoPlus
amaz
Resultados de Busqueda
Kobayashi-san Chi no...
Wanting to take her affection for Kobayashi a step further, dragon maid Toora is confident in her fa...
Ver
Kuragehime: Soreike!...
The Ansars go on an adventure in the forest....
Ver
Amaryllis
The final nine measures of the first ballet, labelled "Le son de la clochette auquel Circé sortit de...
Ver
Amagi Brilliant Park:...
Amagi Brilliant Park is a place where everyone takes on the role that is perfect for them. But every...
Ver



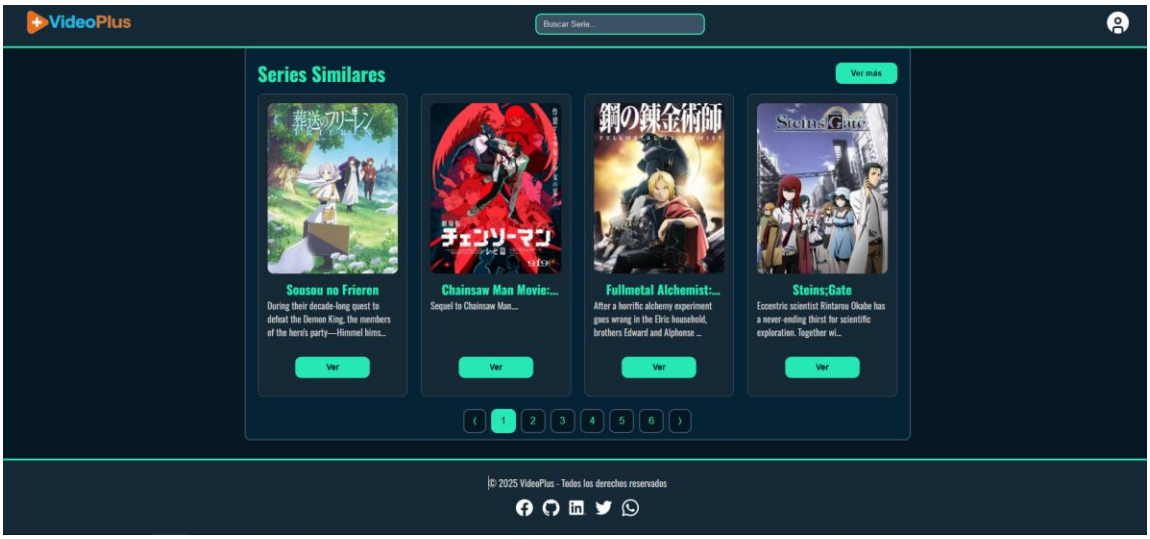


Diagrama de Flujo