

UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ



INGENIERÍA EN SOFTWARE

Periodo 2025-2

CICLO DE DESARROLLO DEL SOFTWARE

Documentación App Web Streaming

AUTORES:

García Álvarez Jhon Jairo

Loor Zambrano Cristell Geovanna

DOCENTE:

Ing. Zambrano Vélez Jonathan David

CHONE-ECUADOR

2025

1. APP WEB “VivaStream”

Se desarrollará una Aplicación Web de Streaming “VivaStream” como proyecto académico integral que demostrará las capacidades del desarrollo web full-stack utilizando tecnologías modernas. Esta plataforma web permitirá a los usuarios explorar, visualizar y gestionar un catálogo diverso de películas a través de una interfaz intuitiva y responsive.

El sistema representará una aplicación web completa que integra backend robusto con Django, frontend interactivo con HTMX y diseño moderno con Tailwind CSS, sirviendo como demostración práctica de desarrollo web contemporáneo.

La plataforma contará con un catálogo organizado de películas clasificadas por géneros cinematográficos, permitiendo a los usuarios explorar contenido de forma intuitiva. Cada película estará almacenada en la base de datos con metadatos completos incluyendo título, descripción, género, duración, imagen promocional y enlace de streaming. Los usuarios podrán acceder al contenido mediante un sistema de reproducción integrado que soportará formatos modernos de video.

El proceso de reproducción iniciará automáticamente al seleccionar cualquier película del catálogo, con un reproductor que incluirá controles estándar de play, pausa, volumen y pantalla completa. El sistema registrará automáticamente el tiempo de reproducción para implementar la funcionalidad de "Continuar viendo".

Para acceder a todas las funcionalidades, los usuarios deberán registrarse en el sistema mediante un formulario de registro que capturará información básica como nombre, email y

contraseña. Una vez registrados, podrán iniciar sesión mediante credenciales seguras validadas por el sistema de autenticación de Django.

Cada usuario contará con un perfil personal donde podrá gestionar su información básica y acceder a sus secciones personalizadas. El sistema mantendrá sesiones seguras que permitirán la navegación prolongada sin necesidad de reautenticación constante, implementando medidas de seguridad estándar para proteger los datos de los usuarios.

Los usuarios autenticados tendrán la capacidad de marcar películas como favoritas mediante un sistema de guardado de un clic. Todas las películas marcadas como favoritas se almacenarán en una sección especial dentro del perfil de usuario, permitiendo acceso rápido al contenido preferido. Esta funcionalidad incluirá opciones para agregar y remover favoritos desde cualquier vista donde aparezca la película.

El sistema de favoritos se integrará de manera cohesiva con el catálogo principal, mostrando indicadores visuales que permitan identificar rápidamente qué películas han sido marcadas por el usuario actual.

La plataforma implementará un sistema comprehensivo de seguimiento de visualización que registrará automáticamente el progreso de reproducción de cada película. Cuando un usuario pause o detenga la reproducción, el sistema guardará el timestamp exacto para permitir reanudar posteriormente desde el mismo punto.

En la página principal, los usuarios autenticados verán una sección especial de "Continuar viendo" que mostrará las películas que han sido parcialmente visualizadas, con indicadores de progreso que faciliten la reanudación inmediata.

La página de inicio presentará un diseño optimizado para descubrimiento de contenido, organizando las películas en secciones por categorías cinematográficas. Cada categoría mostrará un carrusel horizontal de películas correspondientes, con imágenes promocionales y títulos claramente visibles.

La navegación entre categorías será fluida e intuitiva, implementada con HTMX para permitir transiciones suaves sin recarga completa de página. Los usuarios podrán filtrar el contenido por género mediante un sistema de navegación lateral o superior que se adaptará responsiveamente a diferentes tamaños de pantalla.

El sistema incluirá un panel de administración robusto desarrollado con el Django Admin personalizado, que permitirá la gestión completa del catálogo de películas. Los administradores podrán agregar nuevas películas mediante formularios intuitivos que capturen todos los metadatos necesarios, incluyendo la subida de imágenes promocionales y enlaces de video.

Las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) estarán disponibles para todo el contenido, permitiendo modificar información de películas existentes, actualizar categorías y gestionar el estado de publicación del contenido.

El panel de administración incluirá módulos especializados para la gestión de usuarios registrados, permitiendo visualizar listados completos, editar perfiles, y en casos necesarios, desactivar cuentas. Los administradores podrán monitorizar las sesiones activas y el comportamiento general de los usuarios dentro de la plataforma.

El sistema registrará logs de actividad que permitirán tracking de acciones importantes realizadas tanto por usuarios comunes como administradores, proporcionando trazabilidad para mantenimiento y solución de problemas.

El corazón de la plataforma utilizará Django como framework principal, implementando una estructura MTV (Model-Template-View) organizada y escalable. Los modelos de datos definirán las entidades principales: User, Movie, Genre, Favorite, WatchHistory, entre otras, con relaciones bien establecidas que garantizan integridad referencial.

Las vistas manejarán la lógica de presentación y procesamiento de requests, implementando class-based views para operaciones CRUD y function-based views para endpoints HTMX. El sistema de URLs estará organizado modularmente para mantener un código limpio y mantenable.

La interfaz de usuario aprovechará HTMX para crear interactividad del lado del cliente sin la complejidad de frameworks JavaScript pesados. Esto permitirá actualizaciones parciales de la UI para acciones como agregar favoritos, filtrar contenido y cargar más películas, todo sin recargas completas de página.

El diseño visual implementará Tailwind CSS como framework de estilos, garantizando una interfaz moderna, responsive y consistentemente diseñada. El sistema utilizará componentes reutilizables y un design system unificado que mantenga coherencia visual throughout toda la aplicación.

La plataforma incorporará un sistema de búsqueda en tiempo real implementado con HTMX, permitiendo a los usuarios encontrar películas por título o descripción de forma instantánea. Los resultados se actualizarán dinámicamente mientras el usuario escribe, proporcionando una experiencia de búsqueda fluida y moderna.

El sistema de paginación inteligente manejará grandes volúmenes de contenido, cargando adicionalmente mediante técnicas de infinite scroll o paginación tradicional, según las necesidades específicas de cada vista. Todas las interacciones prioritarias contarán con estados de loading y manejo de errores elegante que mantengan al usuario informado en todo momento.

Esta arquitectura técnica asegura que la Plataforma de Streaming “VivaStream” no solo cumpla con los requisitos funcionales inmediatos, sino que también establezca una base sólida para posibles expansiones futuras y mejoras incrementales.

2. Análisis de requisitos

Requisitos Funcionales:

2.1. Gestión de Videos

- El sistema podrá registrar usuarios y asignar roles (administrador, usuario estándar, invitado).
- El administrador podrá:
 - Subir, editar y eliminar videos en la plataforma.
 - Configurar metadatos de los videos (título, descripción, categorías, etiquetas, miniatura).
 - Programar videos para publicación futura.
 - Gestionar comentarios o reseñas asociadas a cada video.
- Subida de videos a una plataforma de Stream o subido de manera local con:
 - Transcodificación automática a múltiples calidades (480p, 720p, 1080p).
 - Generación de thumbnails automáticos.
 - Encriptación opcional de videos privados.
- Listado de catálogo de videos:
 - Filtro por categorías, etiquetas y popularidad.
 - Búsqueda rápida con autocompletado.
 - Paginación o scroll infinito según preferencia.
- Reproductor funcional con HTMX:
 - Control de play/pause, volumen, subtítulos y calidad de reproducción.
 - Soporte para reproducción continua y reproducción automática del siguiente video.

- Indicador visual de buffering y errores de carga.

2.2. Gestión de Usuarios

- Registro y autentificación segura:
 - Validación de correo electrónico con token de confirmación.
 - Contraseñas encriptadas usando bcrypt o Argon2.
 - Protección contra ataques de fuerza bruta (rate limiting).
- Login seguro con sesiones persistentes y opción de recordar dispositivo.
- Recuperación de contraseña por email con enlace temporal.
- Logout con confirmación y limpieza de sesiones activas.
- Perfiles de usuario editables:
 - Foto de perfil, biografía, preferencias de visualización y notificaciones.
 - Historial de actividad y configuración de privacidad.

2.3. Historial de visualización

- Tracking automático de progreso de reproducción de cada video.
- Sección “Continuar viendo” en dashboard con acceso directo a videos parcialmente vistos.
- Limpieza automática de historial antiguo según política definida (ej: 6 meses).
- Opción de eliminar items específicos del historial.
- Estadísticas de visualización por usuario:
 - Tiempo total visto, videos completados, videos en progreso.
 - Reportes de tendencias y hábitos de consumo.

2.4. Sistema de favoritos

- Agregar/remover videos de favoritos desde el reproductor o el listado de videos.
- Lista organizada de favoritos: filtros por categoría, fecha de agregado o nombre.
- Compartir listas de favoritos mediante enlace o integración con redes sociales.
- Ordenamiento personalizado en favoritos (manual o automático por criterios).
- Exportación básica de lista de favoritos en CSV o JSON.
- Notificaciones opcionales cuando un video en favoritos es actualizado o removido.

Requisitos No Funcionales:

- Responsive design con Tailwind CSS optimizado para móviles, tablets y desktops.
- Tiempo de carga < 3 segundos incluso con conexión media.
- Compatibilidad con navegadores modernos: Chrome, Firefox, Edge, Safari.
- Escalabilidad: la arquitectura permitirá agregar nuevos usuarios y videos sin degradación del rendimiento.
- Seguridad: protección contra XSS, CSRF y SQL Injection.
- Disponibilidad: 99.5% uptime mínimo, con manejo de errores amigable.
- Logging y monitoreo: registros de actividad, errores y métricas de uso.
- Soporte multilenguaje opcional (internacionalización futura).

3. Justificación

Django: Fundamentos Técnicos y Ventajas

- **Modularidad y Escalabilidad:** Django sigue el principio "Don't Repeat Yourself" (DRY) y ofrece una arquitectura basada en aplicaciones reutilizables. Esto permite un desarrollo organizado donde cada funcionalidad (videos, usuarios, analytics) se encapsula en módulos independientes, facilitando el mantenimiento y la escalabilidad futura.
- **ORM Poderoso:** El Object-Relational Mapping de Django abstrae la complejidad del SQL, permitiendo trabajar con la base de datos usando Python. Esto acelera el desarrollo, reduce errores y facilita migraciones entre diferentes motores de bases de datos (SQLite, PostgreSQL, MySQL).
- **Autenticación y Seguridad Integrada:** Django incluye un sistema de autenticación robusto out-of-the-box, con protección contra vulnerabilidades comunes como CSRF, XSS, y SQL injection. Esto es crucial para una aplicación que maneja contenido protegido y datos de usuarios.
- **Panel de Administración Automático:** El admin de Django proporciona una interfaz completa para gestión de contenido sin desarrollo adicional, perfecta para administradores que necesitan gestionar videos y usuarios.

4. HTMX: Enfoque Moderno de Interactividad

Reducción de Complejidad: HTMX permite lograr interactividad del lado del cliente sin la complejidad de frameworks JavaScript pesados como React o Vue. Esto resulta en:

- Menor curva de aprendizaje
- Código más mantenible y debuggeable
- Mejor performance al evitar bundles JavaScript grandes
- Integración natural con templates Django

Tailwind CSS: Desarrollo Visual Eficiente

- Desarrollo Rápido y Consistente: Tailwind CSS permite construir interfaces modernas directamente en el HTML, eliminando la necesidad de escribir CSS personalizado. Esto acelera el desarrollo y mantiene la consistencia visual através del sistema de design tokens.
- Responsive by Default: Tailwind incluye utilidades responsive que facilitan la creación de diseños que funcionan en todos los dispositivos sin media queries complejas.
- Mantenibilidad: El enfoque utility-first de Tailwind evita problemas de especificidad CSS y genera bundles CSS optimizados automáticamente.

5. Arquitectura del Sistema

Modelo MTV (Model-Template-View) de Django:

- **Modelos:** Gestión de Datos y Lógica de Negocio

Los modelos definen la estructura de datos y contienen la lógica de negocio central.

- **Vistas:** Controladores de Lógica de Presentación

Las vistas manejan las requests HTTP y coordinan entre modelos y templates.

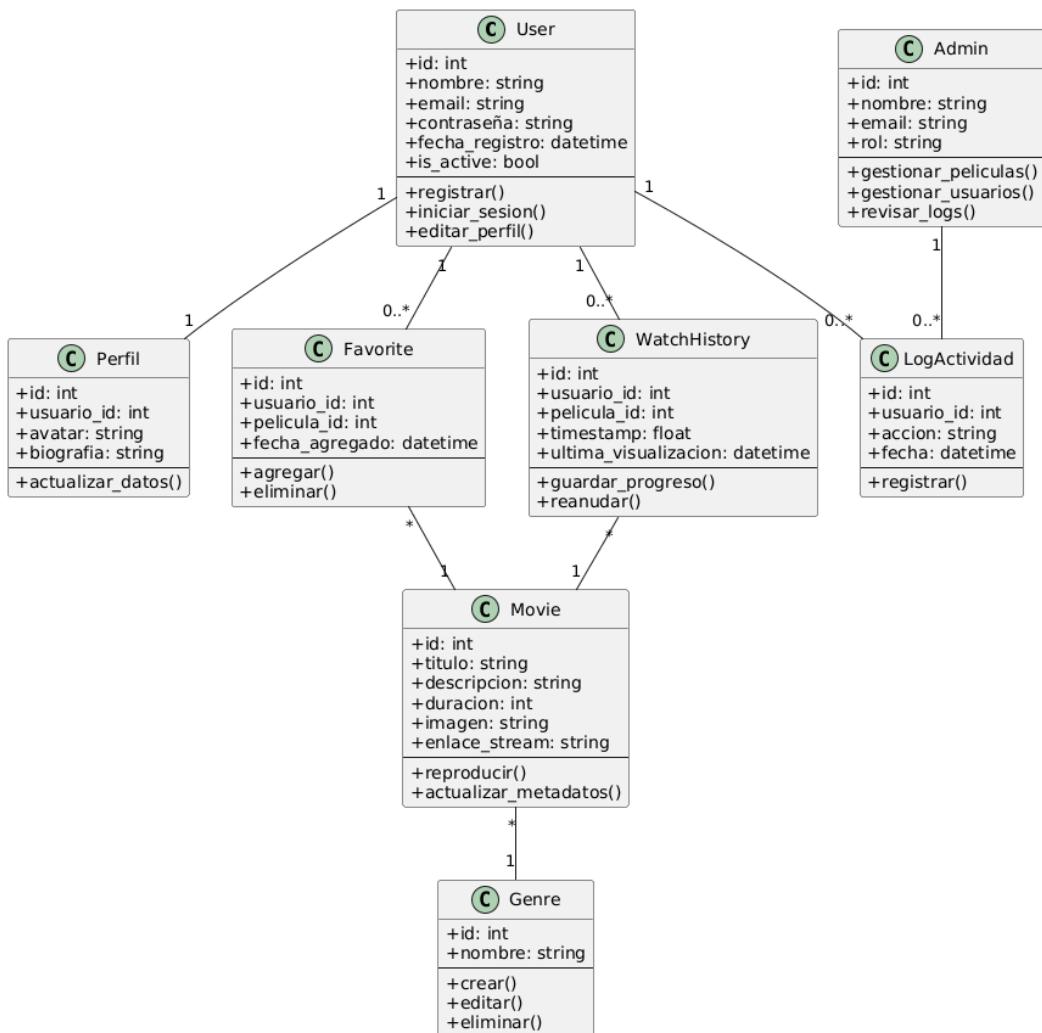
- **Templates:** Capa de Presentación con HTMX

Los templates definen la estructura HTML y contienen la lógica de presentación.

6. Diagramas

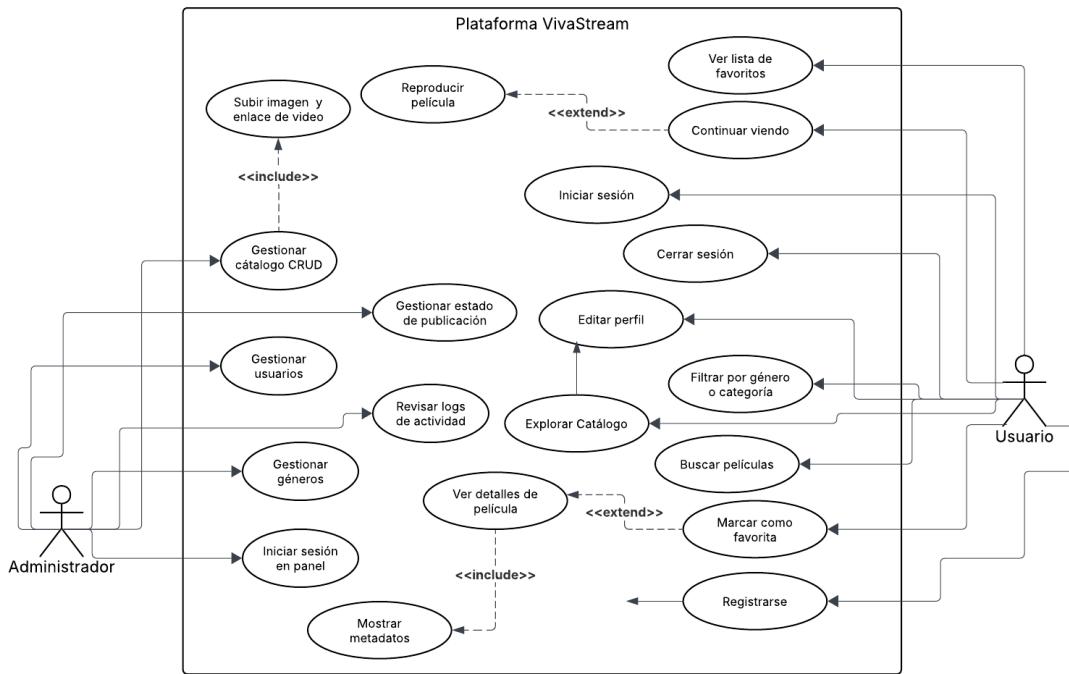
- **Diagrama de clases**

El diagrama de clases de la aplicación web *VivaStream* representa la estructura lógica del sistema, sus entidades principales y las relaciones entre ellas. Este modelo sigue la arquitectura MTV de Django, donde cada clase corresponde a un modelo de base de datos o componente funcional.



- **Diagrama de caso de uso**

El **diagrama de casos de uso** muestra las interacciones entre los actores externos (usuario y administrador) y el sistema *VivaStream*. Permite visualizar qué funcionalidades están disponibles para cada tipo de usuario.



7. Diseño de base de datos

Tabla	Campos principales	Descripción
users	id (PK), nombre, email, contraseña, fecha_registro, is_active	Almacena información básica de los usuarios.
profiles	id (PK), usuario_id (FK), avatar, biografia	Información adicional de cada usuario.
genres	id (PK), nombre	Clasificación de películas por género.

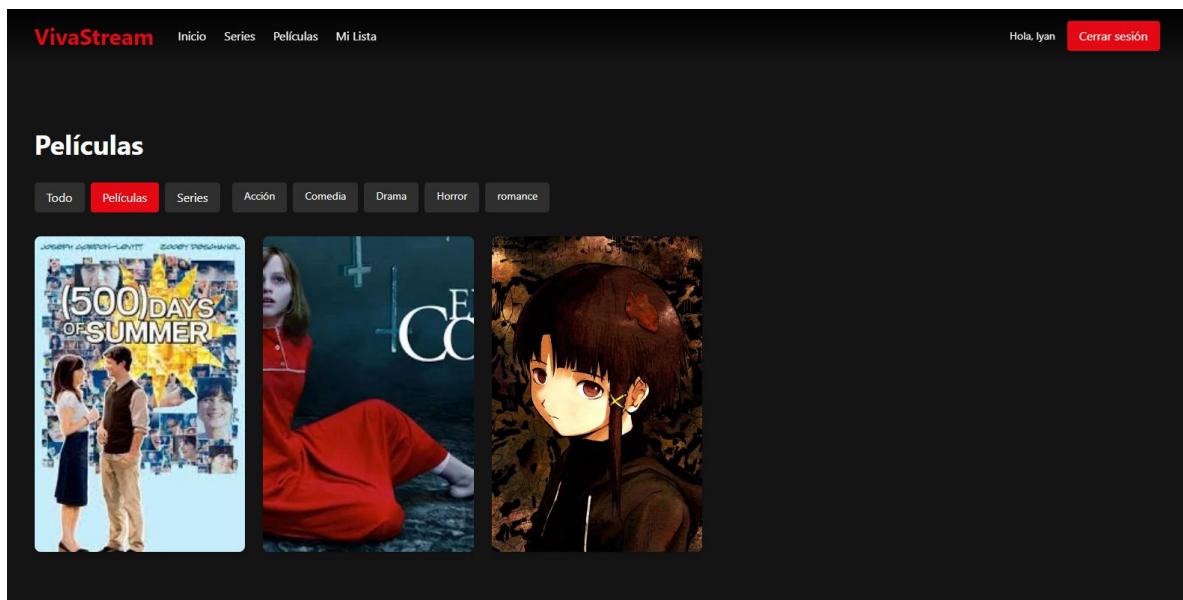
movies	id (PK), titulo, descripcion, duracion, imagen, enlace_stream, genre_id (FK)	Contiene metadatos y enlaces de películas.
favorites	id (PK), usuario_id (FK), pelicula_id (FK), fecha_agregado	Permite marcar películas como favoritas.
watch_history	id (PK), usuario_id (FK), pelicula_id (FK), ultima_visualizacion, timestamp	Guarda progreso de visualización.
admins	id (PK), nombre, email, rol	Usuarios con privilegios de administración.
Log_actividad	id (PK), usuario_id (FK), accion, fecha	Registro de acciones para auditoría.

8. Prototipo

8.1. Inicio



8.2. Catálogo



8.3. Info de película



9. Conclusión

Para concluir, VivaStream refleja un diseño completo y estructurado del sistema, desde los requisitos funcionales y no funcionales hasta los diagramas UML y el diseño de base de datos. Se logró:

- Claridad en la arquitectura: Se definieron actores, casos de uso y relaciones entre componentes para garantizar un flujo lógico de interacción.
- Optimización de datos: El modelo de base de datos relacional permite escalabilidad, integridad de información y eficiencia en consultas críticas como favoritos y progreso de visualización.
- Seguridad y auditoría: La inclusión de roles de administrador y registros de actividad garantiza control sobre el contenido y acciones del usuario.
- Base para implementación: Los diagramas UML y la estructura de la base de datos facilitan la implementación en tecnologías web modernas, asegurando que los desarrolladores puedan integrarlo de manera rápida y organizada.