



Portada del trabajo

Proyecto: Plataforma de Streaming "**Netflix**"

Curso: Programacion III

Asignatura: Programacion III

Profesor: Kelyn Tejada

Alumno: Johan Michael Herrera Rodriguez

Matricula: 2022-1018

Fecha: 04/08/2024

Contenido:

A. Requisitos Iniciales

- Nombre del proyecto de software
- Tecnología para aplicar
- Objetivo del proyecto
- Alcance del proyecto
- Cronograma del proyecto
- Definición de funcionalidades del primer Release
- Requerimientos del sistema para el primer Release

B. Equipo Metodología Scrum

- Definición de tareas a ejecutar
- Definición del equipo de trabajo (roles, habilidades, etc.)
- Herramientas a utilizar
- Definición de épicas

C- Acciones: En base a lo definido en el punto A y B desarrollar lo siguiente:

- Definir las fechas para cada ceremonia de Scrum.
- Afinar al menos 10 historias de usuarios (criterios de aceptación obligatorios).
- Asignar los puntos de historia.

D- Demostración:

- Luego de desarrollar lo anteriormente descrito en el punto A-B-C Adjuntar un video donde se pueda ver el incremento.

2- Plan de Pruebas

Realizar un plan de pruebas que contenga lo siguiente puntos, con al menos 20 casos de pruebas que sean obligatorios dentro de cualquier tipo de sistema (ejemplo Inicio de sesión de usuario, creación de usuarios, creación de roles, etc.):

Importante (los casos de pruebas deben estar relacionados con las HU que fueron definidas para este proyecto.

- 1- Lista de requerimientos funcionales y no funcionales de acorde a las historias de usuarios. (mínimo de 10 HU).
- 2- Definir cuáles son los criterios de aceptación de las pruebas.
- 3- Definir cuáles son los criterios de rechazo en las pruebas.
- 4- Comentar sobre las herramientas de pruebas que estarían usando y justificar respuesta.
- 5- Estimar el tiempo que duraría la ejecución de pruebas. (elaborar cronograma de trabajo).
- 6- Elaborar plantillas para cada caso de pruebas.
- 7- Elaborar una plantilla con los equipos de pruebas y sus responsabilidades.
- 8- Elaborar un plan de automatización de pruebas.
- 9- Ejecutar y demostrar el plan de automación de pruebas.

Introducción

El desarrollo de plataformas de streaming de contenido ha revolucionado la forma en que consumimos entretenimiento. Netflix, como líder en este campo, ha establecido un estándar en términos de calidad, accesibilidad y experiencia de usuario. Este proyecto tiene como objetivo replicar y mejorar las funcionalidades esenciales de Netflix mediante la aplicación de la metodología Agile-Scrum. A través de un enfoque estructurado y colaborativo, se busca desarrollar una plataforma de streaming que ofrezca una experiencia de usuario intuitiva y atractiva, asegurando al mismo tiempo la seguridad y escalabilidad del sistema.

Título del Proyecto:

Creación de una Plataforma de Streaming: Netflix

Descripción:

El presente trabajo detalla el desarrollo de una plataforma de **Netflix** llamada " **Netflix** ", siguiendo la metodología Agile-Scrum. **Netflix** permitirá a los usuarios ver películas, series y documentales en alta calidad, con recomendaciones personalizadas y contenido exclusivo.

A. Requisitos Iniciales

- Nombre del proyecto de software

El nombre del proyecto de software debe ser atractivo, fácil de recordar y reflejar la esencia del producto. Para nuestra plataforma de Streaming, hemos elegido el nombre:

Nombre del Proyecto: **Netflix**

Descripción del nombre:

- **Net:** Hace referencia a Internet, indicando que la plataforma opera en línea.
- **Flix:** Es una abreviatura coloquial de "flicks" (películas), sugiriendo que la plataforma se centra en películas y series.

• **Tecnología para aplicar**

Front-end:

- **React:** Una biblioteca de JavaScript para construir interfaces de usuario. Es ideal para crear aplicaciones web rápidas y escalables.
- **HTML5 y CSS3:** Para estructurar y estilizar las páginas web.
- **JavaScript:** Para la funcionalidad interactiva del sitio web.

Back-end:

- **Node.js:** Un entorno de ejecución para JavaScript que permite construir aplicaciones de red escalables.
- **Express.js:** Un framework web para Node.js que facilita el desarrollo de aplicaciones web y APIs.

Base de datos:

- **MongoDB:** Una base de datos NoSQL que permite almacenar datos en un formato flexible y escalable, ideal para manejar grandes volúmenes de contenido multimedia.

Autenticación y autorización:

- **OAuth 2.0:** Un protocolo de autorización estándar para asegurar las transacciones y la autenticación de usuarios.

Almacenamiento en la nube:

- **AWS S3:** Un servicio de almacenamiento de objetos de Amazon Web Services que ofrece una durabilidad y escalabilidad excepcionales para almacenar contenido multimedia.

Entrega de contenido:

- **AWS CloudFront:** Una red de entrega de contenido (CDN) que distribuye contenido globalmente con baja latencia y alta transferencia de datos.

Recomendaciones:

- **Python y TensorFlow:** Para desarrollar y desplegar algoritmos de Machine Learning que generen recomendaciones personalizadas basadas en el historial de visualización de los usuarios.

CI/CD (Integración Continua/Despliegue Continuo):

- **Jenkins:** Una herramienta de automatización de código abierto para facilitar la integración continua.
- **Docker:** Para crear, enviar y ejecutar aplicaciones en contenedores.
- **Kubernetes:** Para la orquestación de contenedores y la gestión de aplicaciones en contenedores a gran escala.

Pruebas:

- **Jest:** Un framework de pruebas para aplicaciones JavaScript.
- **Selenium:** Una herramienta para la automatización de pruebas de aplicaciones web.

Diseño:

- **Figma:** Una herramienta de diseño de interfaces colaborativa basada en la web.
 - **Adobe XD:** Un software de diseño de experiencias de usuario.
- **Objetivo del proyecto**

El objetivo del proyecto es desarrollar una plataforma de Streaming de contenido multimedia llamada **Netflix** que ofrezca a los usuarios una experiencia de visualización de alta calidad. Los objetivos específicos son:

Objetivo General:

Crear una plataforma de Streaming que permita a los usuarios ver una amplia variedad de películas, series, documentales y contenido original, ofreciendo una experiencia de usuario intuitiva, recomendaciones personalizadas y alta calidad de transmisión.

Objetivos Específicos:

1. Proveer un catálogo diverso de contenido:

- Ofrecer una amplia gama de películas, series, y documentales de diferentes géneros y categorías.
- Incluir contenido original exclusivo para atraer y retener a los usuarios.

2. Facilitar la navegación y búsqueda de contenido:

- Desarrollar una interfaz de usuario intuitiva que permita a los usuarios explorar y buscar contenido fácilmente.
- Implementar filtros y opciones de búsqueda avanzada para ayudar a los usuarios a encontrar el contenido que desean.

3. Implementar un sistema de reproducción de video robusto:

- Crear un reproductor de video que soporte múltiples formatos y calidades, con controles básicos como play, pause, avance, y retroceso.
- Asegurar una transmisión fluida y de alta calidad, incluso en conexiones de red variables.

4. Desarrollar un sistema de recomendaciones personalizadas:

- Utilizar algoritmos de Machine Learning para ofrecer recomendaciones basadas en el historial de visualización de cada usuario.
- Permitir a los usuarios marcar contenido como favorito y recibir recomendaciones relevantes.

5. Garantizar la seguridad y privacidad de los datos del usuario:

- Implementar un sistema de autenticación seguro y manejo de datos personales.
- Proteger la información del usuario y garantizar el cumplimiento con las normativas de privacidad.

6. Ofrecer soporte multiplataforma:

- Desarrollar aplicaciones y una interfaz web que sean accesibles desde diferentes dispositivos, incluyendo computadoras, tablets, y smartphones.
- Garantizar una experiencia de usuario consistente en todas las plataformas.

7. Proveer una infraestructura escalable y confiable:

- Utilizar tecnologías en la nube para gestionar el almacenamiento y la entrega de contenido de manera eficiente.

- Asegurar que la plataforma pueda escalar para manejar un gran número de usuarios simultáneos.

- **Alcance del proyecto**

El alcance del proyecto define las funcionalidades y características que se desarrollarán en las distintas fases del proyecto para asegurar que la plataforma Netflix cumpla con los objetivos establecidos. El alcance se dividirá en dos fases principales: la primera fase (MVP Producto Mínimo Viable) y las futuras fases para mejorar y expandir la plataforma.

Primera Fase (MVP):

Funcionalidades esenciales:

1. Registro y autenticación de usuarios:

- Sistema de registro de nuevos usuarios.
- Inicio de sesión y autenticación segura.
- Recuperación de contraseña.

2. Navegación y búsqueda de contenido:

- Interfaz de usuario intuitiva para explorar el catálogo.
- Barra de búsqueda para encontrar contenido por título, género, etc.
- Página de detalles del contenido con información relevante.

3. Reproducción de video:

- Reproductor de video con controles básicos (play, pause, avance, retroceso).
- Ajustes de calidad de video.
- Soporte para subtítulos y varios idiomas.

4. Perfiles de usuario:

- Creación de múltiples perfiles por cuenta.
- Personalización de perfiles (nombre, avatar).
- Historial de visualización y listas de seguimiento por perfil.

5. Sistema de recomendaciones básicas:

- Algoritmo inicial de recomendaciones basado en el historial de visualización.
- Sección de "Recomendado para ti" en la página principal.

6. Infraestructura básica:

- Servidores y almacenamiento en la nube (AWS S3).
- CDN (AWS CloudFront) para entrega rápida de contenido.
- Sistema de gestión de contenido (CMS) para administrar el catálogo.

Futuras Fases:

Mejoras y funcionalidades adicionales:

1. Recomendaciones avanzadas:

- Implementación de algoritmos de Machine Learning más sofisticados.
- Personalización avanzada basada en preferencias y comportamiento del usuario.

2. Funcionalidades sociales:

- Comentarios y valoraciones de contenido por parte de los usuarios.
- Listas de reproducción compartidas entre usuarios.

3. Aplicaciones para dispositivos móviles y Smart TVs:

- Desarrollo de aplicaciones nativas para iOS y Android.
- Integración con plataformas de Smart TV y dispositivos de Streaming como Roku y Apple TV.

4. Soporte para múltiples idiomas y subtítulos:

- Expansión del soporte de idiomas para atraer a una audiencia global.
- Subtítulos y doblajes en varios idiomas.

5. Sistema de suscripción y facturación:

- Implementación de planes de suscripción mensual/anual.
- Integración de sistemas de pago seguros.

6. Escalabilidad y rendimiento:

- Optimización de la infraestructura para soportar un gran número de usuarios simultáneos.
- Monitoreo y mejora continua del rendimiento de la plataforma.

- **Cronograma del proyecto**

El cronograma del proyecto detalla las fases y actividades necesarias para completar el desarrollo de la plataforma Netflix, distribuidas a lo largo de un período específico. Cronograma detallado para la implementación del MVP (Producto Mínimo Viable) y las fases futuras.

- **Cronograma del Proyecto (MVP):**

Fase	Duración	Descripción
Planificación	2 semanas	Investigación de mercado, definición de objetivos y alcance, selección de tecnologías.
Diseño	3 semanas	Diseño de la arquitectura del sistema, diseño de la interfaz de usuario (UI) y experiencia de usuario (UX).
Desarrollo (MVP)	8 semanas	Desarrollo del backend y frontend, implementación de funcionalidades clave como registro, autenticación, navegación, reproducción de video, y perfiles de usuario.
Pruebas	2 semanas	Pruebas unitarias, de integración y de usabilidad. Aseguramiento de la calidad y corrección de errores.
Despliegue	1 semana	Configuración de la infraestructura en la nube, despliegue del MVP, monitoreo inicial.
Monitoreo y ajustes	2 semanas	Monitoreo del rendimiento y comportamiento del sistema en producción, ajustes y optimizaciones necesarias.

- **Cronograma del Proyecto (Fases Futuras):**

Fase	Duración	Descripción
Mejoras de recomendaciones	4 semanas	Implementación de algoritmos de Machine Learning avanzados para recomendaciones personalizadas.
Funcionalidades sociales	3 semanas	Desarrollo de funcionalidades de comentarios, valoraciones y listas de reproducción compartidas.
Aplicaciones móviles y Smart TV	6 semanas	Desarrollo de aplicaciones nativas para iOS, Android y plataformas de Smart TV.
Soporte multilingüe	3 semanas	Expansión del soporte de idiomas y subtítulos.

Sistema de suscripción y facturación	4 semanas	Implementación de planes de suscripción y sistemas de pago seguros.
Optimización de escalabilidad y rendimiento	4 semanas	Optimización de la infraestructura para soportar un gran número de usuarios simultáneos, mejora continua del rendimiento.

• Definición de funcionalidades del primer Release

En el primer Release (MVP) de Netflix, se implementarán las funcionalidades esenciales que permitirán a los usuarios registrar sus cuentas, explorar el catálogo de contenido, reproducir videos y recibir recomendaciones básicas. A continuación, se detallan estas funcionalidades:

Funcionalidades del MVP:

1. Registro y autenticación de usuarios:

- **Registro de usuarios nuevos:** Permitir a los usuarios crear una cuenta proporcionando su correo electrónico y contraseña.
- **Inicio de sesión:** Permitir a los usuarios autenticarse en la plataforma con su correo electrónico y contraseña.
- **Recuperación de contraseña:** Permitir a los usuarios recuperar su contraseña a través de su correo electrónico.

2. Navegación y búsqueda de contenido:

- **Interfaz de usuario intuitiva:** Una interfaz que permita a los usuarios explorar el catálogo de contenido de manera fácil y eficiente.
- **Búsqueda de contenido:** Barra de búsqueda que permita a los usuarios encontrar contenido por título, género, y otros filtros.
- **Página de detalles del contenido:** Página que muestre información detallada sobre cada título, incluyendo sinopsis, elenco, duración y calificaciones.

3. Reproducción de video:

- **Reproductor de video:** Un reproductor con controles básicos como play, pause, avance y retroceso.
- **Calidad de video:** Ajustes que permitan a los usuarios seleccionar la calidad de reproducción según su conexión a Internet.
- **Subtítulos y doblajes:** Soporte para subtítulos en varios idiomas y opciones de doblaje (si están disponibles).

4. Perfiles de usuario:

- **Creación de perfiles:** Permitir a los usuarios crear múltiples perfiles bajo una misma cuenta.
- **Personalización de perfiles:** Permitir a los usuarios personalizar sus perfiles con un nombre y un avatar.
- **Historial de visualización:** Cada perfil mantendrá su propio historial de visualización y lista de seguimiento.

5. Sistema de recomendaciones básicas:

- **Algoritmo de recomendaciones:** Un algoritmo básico que genere recomendaciones basadas en el historial de visualización del usuario.
- **Sección de "Recomendado para ti":** Una sección en la página principal que muestre contenido recomendado para cada perfil.

6. Infraestructura básica:

- **Almacenamiento en la nube:** Utilización de AWS S3 para almacenar el contenido multimedia.
 - **CDN (Content Delivery Network):** Implementación de AWS CloudFront para la entrega rápida de contenido a nivel global.
 - **Sistema de gestión de contenido (CMS):** Un sistema que permita a los administradores cargar, gestionar y categorizar el contenido en la plataforma.
-
- **Requerimientos del sistema para el primer Release**

Para garantizar que el primer Release (MVP) de Netflix funcione de manera eficiente y segura, se deben cumplir ciertos requerimientos del sistema. Estos requerimientos abarcan tanto hardware como software, así como aspectos de rendimiento y seguridad.

Requerimientos de Hardware:

1. Servidores:

- **Servidor de aplicaciones:** Servidores con capacidad para manejar múltiples peticiones simultáneas.
- **Servidor de base de datos:** Servidores optimizados para almacenamiento y consulta rápida de datos.
- **CDN (Content Delivery Network):** Servidores distribuidos globalmente para la entrega rápida de contenido multimedia.

2. Almacenamiento:

- **Almacenamiento en la nube:** Uso de servicios como AWS S3 para almacenar grandes volúmenes de contenido multimedia.
- **Redundancia y respaldo:** Mecanismos para asegurar la disponibilidad y recuperación de datos en caso de fallos.

Requerimientos de Software:

1. Backend:

- **Node.js:** Plataforma para construir el servidor de aplicaciones.
- **Express.js:** Framework para la construcción de APIs y manejo de rutas.
- **MongoDB:** Base de datos NoSQL para almacenar datos del usuario, historial de visualización y metadatos de contenido.

2. Frontend:

- **React:** Biblioteca para construir la interfaz de usuario.
- **HTML5 y CSS3:** Tecnologías para estructurar y estilizar las páginas web.
- **JavaScript:** Lenguaje de programación para añadir interactividad y funcionalidades dinámicas.

3. Infraestructura en la nube:

- **AWS EC2:** Instancias de computación en la nube para alojar servidores de aplicaciones.
- **AWS S3:** Servicio de almacenamiento de objetos para contenido multimedia.
- **AWS CloudFront:** CDN para entrega rápida de contenido.
- **AWS RDS:** Servicio de base de datos gestionado, si se requiere una base de datos relacional adicional.

4. Seguridad y autenticación:

- **OAuth 2.0:** Protocolo de autorización para la autenticación de usuarios.
- **HTTPS:** Protocolo seguro para la transferencia de datos entre el cliente y el servidor.
- **Firewalls y encriptación:** Para proteger los datos del usuario y asegurar la infraestructura.

Requerimientos de Rendimiento:

1. Tiempo de respuesta:

- El tiempo de respuesta para la carga de páginas y reproducción de contenido debe ser inferior a 2 segundos.

2. Escalabilidad:

- La arquitectura del sistema debe permitir la escalabilidad horizontal para manejar el crecimiento de la base de usuarios.

3. Capacidad de usuarios concurrentes:

- El sistema debe soportar al menos 10,000 usuarios concurrentes durante el primer Release.

Requerimientos de Seguridad:

1. Autenticación y autorización:

- Implementación de un sistema seguro de inicio de sesión y gestión de sesiones.
- Protección contra ataques comunes como SQL injection, Cross-Site Scripting (XSS), y Cross-Site Request Forgery (CSRF).

2. Protección de datos:

- Encriptación de datos sensibles tanto en tránsito como en reposo.
- Políticas de privacidad y manejo de datos personales conforme a regulaciones (como GDPR).

3. Monitoreo y alertas:

- Implementación de sistemas de monitoreo para detectar y alertar sobre posibles fallos y ataques.
- Uso de herramientas como AWS CloudWatch para el monitoreo de infraestructura.

B. Equipo Metodología Scrum

- Definición de tareas a ejecutar

Para simplificar el proceso y centrarnos en las funcionalidades clave (inicio de sesión, creación de usuarios, creación de roles, búsqueda de contenido y navegación), reduciremos el número de sprints y agruparemos las tareas en 3 sprints esenciales.

Sprint 1: Configuración inicial y autenticación de usuarios

1. Planificación del proyecto:

- Definir objetivos y alcance del MVP.
- Seleccionar tecnologías y herramientas.

2. Configuración del entorno de desarrollo:

- Configurar servidores y entornos de desarrollo.
- Implementar control de versiones (Git).

3. Diseño y desarrollo de la autenticación:

- Diseñar y desarrollar el sistema de registro de usuarios.
- Implementar el inicio de sesión y autenticación segura.
- Crear funcionalidad para la recuperación de contraseñas.

Sprint 2: Creación de roles y gestión de usuarios

1. Creación de roles:

- Definir roles de usuario (administrador, usuario estándar, etc.).
- Implementar lógica de roles y permisos.

2. Gestión de perfiles:

- Crear funcionalidad para la creación y personalización de perfiles.
- Implementar el historial de visualización y listas de seguimiento.

Sprint 3: Búsqueda y navegación de contenido

1. Diseño de la interfaz de usuario (UI):

- Crear wireframes y prototipos.
- Desarrollar la interfaz de navegación y búsqueda de contenido.

2. Desarrollo del backend:

- Crear endpoints para la búsqueda y gestión de contenido.
- Integrar con la base de datos para la gestión del catálogo de contenido.

3. Integración y pruebas:

- Conectar el frontend con la API para la búsqueda y navegación de contenido.
- Realizar pruebas unitarias y de integración.
- Ajustar y optimizar la interfaz de usuario.

- **Definición del equipo de trabajo (roles, habilidades, etc.)**

Para asegurar un desarrollo eficiente y efectivo del proyecto, es crucial tener un equipo bien definido con roles claros y habilidades adecuadas.

Roles del equipo Scrum:

1. Scrum Máster:

- **Responsabilidades:** Facilitar las reuniones de Scrum, eliminar impedimentos, asegurar que el equipo siga las prácticas de Scrum, y servir de enlace entre el equipo y las partes interesadas.
- **Habilidades:** Experiencia en metodologías Agile/Scrum, habilidades de comunicación, liderazgo y resolución de conflictos.

2. Product Owner:

- **Responsabilidades:** Definir y priorizar el backlog del producto, asegurar que el equipo entiende los requisitos del cliente, y tomar decisiones sobre la dirección del producto.
- **Habilidades:** Conocimiento profundo del negocio y del mercado, habilidades de comunicación, toma de decisiones y gestión de stakeholders.

3. Equipo de Desarrollo:

- **Responsabilidades:** Desarrollar las funcionalidades del producto, colaborar estrechamente para cumplir con los objetivos del sprint, y garantizar la calidad del producto.
- **Habilidades:** Conocimientos técnicos en desarrollo frontend y backend, bases de datos, pruebas, y despliegue.

Composición del equipo de desarrollo:

1. Desarrolladores Backend:

- **Responsabilidades:** Desarrollar y mantener la API, gestionar la base de datos, y asegurar la integración del sistema.
- **Habilidades:** Node.js, Express.js, MongoDB, integración de API, y conocimientos en seguridad.

2. Desarrolladores Frontend:

- **Responsabilidades:** Desarrollar la interfaz de usuario, implementar componentes UI/UX, y asegurar la integración con el backend.
- **Habilidades:** React, HTML5, CSS3, JavaScript, y diseño responsivo.

3. Especialistas en QA (Quality Assurance):

- **Responsabilidades:** Realizar pruebas unitarias, de integración y de usabilidad, y garantizar la calidad del producto antes de su despliegue.
- **Habilidades:** Automatización de pruebas, pruebas manuales, herramientas de testing (como Selenium o Jest), y atención al detalle.

4. DevOps/Ingeniero de Infraestructura:

- **Responsabilidades:** Configurar y gestionar la infraestructura en la nube, asegurar el despliegue continuo, y monitorear el rendimiento del sistema.
- **Habilidades:** AWS (EC2, S3, CloudFront), Docker, Kubernetes, y herramientas de CI/CD (como Jenkins).

• Herramientas a utilizar

Para el desarrollo del proyecto utilizando la metodología Agile-Scrum, es fundamental contar con un conjunto de herramientas que faciliten la colaboración, gestión del proyecto, desarrollo, pruebas, y despliegue.

Herramientas de gestión del proyecto y colaboración:

1. Azure Devops:

- **Uso:** Gestión de tareas y sprints, seguimiento de progreso, creación de historias de usuario y épicas.
- **Beneficios:** Integración con otras herramientas de desarrollo, personalización y reportes detallados.

2. Confluence:

- **Uso:** Documentación del proyecto, almacenamiento de conocimientos y colaboración en tiempo real.
- **Beneficios:** Integración con Jira, fácil de usar para toda la organización, soporte para documentos colaborativos.

3. Slack:

- **Uso:** Comunicación en tiempo real entre los miembros del equipo.
- **Beneficios:** Integraciones con Jira, GitHub, y otras herramientas, canales organizados por temas.

Herramientas de desarrollo:

1. Visual Studio Code:

- **Uso:** Editor de código fuente para el desarrollo frontend y backend.
- **Beneficios:** Extensiones para Node.js, React, integración con Git, depurador integrado.

2. GitHub:

- **Uso:** Control de versiones, colaboración en el código, revisión de código.
- **Beneficios:** Integración con CI/CD, gestión de repositorios, fácil de colaborar con pull requests.

3. Postman:

- **Uso:** Pruebas de API y desarrollo de servicios web.
- **Beneficios:** Fácil de usar para probar y documentar API, colaboración en equipos.

Herramientas de infraestructura y despliegue:

1. AWS (Amazon Web Services):

- **Uso:** Infraestructura en la nube, servicios de almacenamiento, entrega de contenido.

Servicios clave:

- **EC2:** Instancias de computación para alojar servidores de aplicaciones.
- **S3:** Almacenamiento de objetos para contenido multimedia.
- **CloudFront:** CDN para la entrega rápida de contenido.
- **RDS:** Base de datos relacional (si se requiere).

2. Docker:

- **Uso:** Contenerización de aplicaciones, asegurando consistencia entre entornos de desarrollo y producción.
- **Beneficios:** Facilita el despliegue y escalabilidad de aplicaciones.

3. Jenkins:

- **Uso:** Integración y despliegue continuo (CI/CD).
- **Beneficios:** Automatiza las compilaciones, pruebas y despliegue de aplicaciones.

Herramientas de pruebas y calidad:

1. Selenium:

- **Uso:** Automatización de pruebas de interfaz de usuario.
- **Beneficios:** Permite realizar pruebas end-to-end, soporta múltiples navegadores.

2. Jest:

- **Uso:** Pruebas unitarias y de integración para JavaScript.
- **Beneficios:** Fácil de configurar, rápido y con buena integración con React.

3. SonarQube:

- **Uso:** Análisis de la calidad del código.
- **Beneficios:** Detecta bugs, vulnerabilidades y deuda técnica en el código.

- Definición de épicas

Épica 1: Autenticación y Gestión de Usuarios

Descripción: Esta épica abarca todas las funcionalidades relacionadas con la autenticación de usuarios, la creación de cuentas, y la gestión de perfiles y roles.

Historias de Usuario:

1. **Registro de usuarios:** Como usuario, quiero poder registrarme en la plataforma para acceder al contenido.
2. **Inicio de sesión:** Como usuario registrado, quiero poder iniciar sesión para acceder a mi cuenta.
3. **Recuperación de contraseña:** Como usuario, quiero poder recuperar mi contraseña en caso de olvidarla.

4. **Gestión de perfiles:** Como usuario, quiero poder crear y gestionar diferentes perfiles dentro de mi cuenta.
5. **Asignación de roles:** Como administrador, quiero poder asignar roles y permisos a los usuarios para gestionar el acceso.

Épica 2: Búsqueda y Navegación de Contenido

Descripción: Esta épica incluye todas las funcionalidades necesarias para que los usuarios puedan buscar y navegar a través del catálogo de contenido disponible en la plataforma.

Historias de Usuario:

1. **Búsqueda de contenido:** Como usuario, quiero poder buscar contenido específico utilizando palabras clave.
2. **Navegación por categorías:** Como usuario, quiero poder navegar el contenido por categorías (géneros, populares, recomendados).
3. **Visualización de detalles del contenido:** Como usuario, quiero ver detalles sobre el contenido (descripción, calificación, actores).
4. **Reproducción de contenido:** Como usuario, quiero poder reproducir el contenido seleccionado sin interrupciones.
5. **Historial de visualización:** Como usuario, quiero poder ver mi historial de visualización para acceder fácilmente a contenido previamente visto.

Épica 3: Interfaz de Usuario y Experiencia de Usuario (UI/UX)

Descripción: Esta épica abarca el diseño y desarrollo de la interfaz de usuario, asegurando una experiencia de usuario intuitiva y atractiva.

Historias de Usuario:

1. **Diseño responsivo:** Como usuario, quiero que la plataforma se vea bien y funcione correctamente en diferentes dispositivos (móviles, tablets, computadoras).
2. **Interfaz intuitiva:** Como usuario, quiero que la interfaz sea fácil de navegar y entender.
3. **Personalización de la UI:** Como usuario, quiero poder personalizar ciertos aspectos de mi experiencia de usuario (temas, configuración de cuenta).
4. **Feedback y errores:** Como usuario, quiero recibir feedback claro y útil cuando ocurra un error o una acción sea exitosa.

Épica 4: Infraestructura y Despliegue

Descripción: Esta épica incluye todas las tareas relacionadas con la configuración de la infraestructura, el despliegue de la aplicación y la gestión de la infraestructura en la nube.

Historias de Usuario:

1. **Configuración de servidores:** Como desarrollador, quiero configurar los servidores para alojar la aplicación y el contenido.
2. **Despliegue continuo:** Como desarrollador, quiero automatizar el proceso de despliegue para realizar actualizaciones rápidas y eficientes.
3. **Monitoreo y alertas:** Como administrador, quiero monitorear la infraestructura y recibir alertas sobre cualquier problema para asegurar la disponibilidad y rendimiento del sistema.

Épica 5: Seguridad

Descripción: Esta épica abarca todas las tareas relacionadas con la seguridad de la plataforma, protegiendo los datos del usuario y asegurando un acceso seguro a los servicios.

Historias de Usuario:

1. **Autenticación segura:** Como usuario, quiero que mi información de inicio de sesión esté protegida contra accesos no autorizados.
2. **Encriptación de datos:** Como administrador, quiero que los datos sensibles sean encriptados tanto en tránsito como en reposo.
3. **Protección contra ataques:** Como administrador, quiero implementar medidas para proteger la plataforma contra ataques comunes (XSS, CSRF, SQL Injection).

2- Plan de Prueba

1. Requerimientos Funcionales y No Funcionales (Historias de Usuario)

Requerimientos Funcionales:

- **Inicio de Sesión:** El sistema debe permitir a los usuarios registrados iniciar sesión con su correo electrónico y contraseña.
- **Selección de Perfil:** Tras iniciar sesión, el sistema debe mostrar la página de selección de perfiles, y el usuario debe poder seleccionar el perfil 'Johan'.

- **Búsqueda de Contenido:** El sistema debe permitir a los usuarios buscar series o películas, como 'Dexter'.
- **Navegación por Géneros:** Los usuarios deben poder navegar por las categorías de géneros disponibles sin usar la barra de búsqueda.
- **Reproducción de Contenido:** El sistema debe permitir a los usuarios reproducir contenido seleccionado, como la serie 'SWAT', y también debe manejar la reproducción de contenido al hacer clic en el botón "Reproducir" o "Reanudar".

Requerimientos No Funcionales:

- **Tiempo de Respuesta:** Las acciones de navegación y búsqueda deben completarse en un tiempo razonable (por ejemplo, menos de 5 segundos).
- **Interfaz de Usuario:** La interfaz debe ser intuitiva y accesible, con botones claramente visibles y accesibles.
- **Seguridad:** Las credenciales del usuario deben ser manejadas de manera segura durante el proceso de inicio de sesión.
- **Compatibilidad:** El sistema debe ser compatible con navegadores modernos y versiones actuales de los sistemas operativos.
- **Escalabilidad:** El sistema debe manejar múltiples usuarios simultáneamente sin degradar su rendimiento

2. Criterios de Aceptación de las Pruebas

- **Inicio de Sesión y Selección de Perfil:** El usuario debe poder iniciar sesión con éxito, y el perfil 'Johan' debe ser seleccionado correctamente, con capturas de pantalla de la página de perfiles y la página principal.
- **Búsqueda de Contenido:** La búsqueda de 'Dexter' debe devolver resultados relevantes, y el contenido debe ser visible en la interfaz.
- **Navegación por Géneros:** Los géneros deben ser accesibles y navegables, mostrando contenido relacionado con cada categoría.
- **Reproducción de Contenido:** La serie 'SWAT' debe comenzar a reproducirse correctamente, y el botón de "Reproducir" o "Reanudar" debe funcionar según sea necesario.
- **Visualización de Detalles de Contenido:** Los detalles del contenido específico deben ser visibles y accesibles en la interfaz.

3. Criterios de Rechazo en las Pruebas

- **Inicio de Sesión y Selección de Perfil:** Si el usuario no puede iniciar sesión o el perfil 'Johan' no aparece en la selección, la prueba se considera fallida.
- **Búsqueda de Contenido:** Si la búsqueda de 'Dexter' no devuelve resultados relevantes o el contenido no está visible, la prueba se considera fallida.
- **Navegación por Géneros:** Si los géneros no están accesibles o no muestran contenido relacionado, la prueba se considera fallida.
- **Reproducción de Contenido:** Si la serie 'SWAT' no comienza a reproducirse o el botón de reproducción no funciona, la prueba se considera fallida.

- **Visualización de Detalles de Contenido:** Si los detalles del contenido específico no son visibles o accesibles, la prueba se considera fallida.

4. Herramientas de Pruebas y Justificación

Herramientas Utilizadas:

- **Selenium WebDriver:** Para la automatización de pruebas de navegador. Permite controlar el navegador para realizar acciones automatizadas y verificar la funcionalidad de la interfaz de usuario.
- **WebDriverManager:** Para gestionar la instalación y actualización de controladores de navegador. Simplifica la configuración del entorno de pruebas.
- **Python:** Lenguaje de programación utilizado para escribir los scripts de prueba, gracias a su simplicidad y potentes bibliotecas para pruebas automatizadas.

Justificación:

- **Selenium WebDriver** es ampliamente utilizado para pruebas de aplicaciones web debido a su capacidad para interactuar con los elementos de la interfaz de usuario de manera programática.
- **WebDriverManager** simplifica la gestión de controladores de navegador, evitando la necesidad de descargar e instalar manualmente.
- **Python** es elegido por su legibilidad y la disponibilidad de bibliotecas y herramientas para automatización de pruebas.

5. Estimación del Tiempo y Cronograma de Trabajo

Estimación del Tiempo:

- **Desarrollo de Pruebas:** 2-3 horas por prueba (10-15 horas en total)
- **Ejecución de Pruebas:** 1 hora por prueba (5 horas en total)
- **Revisión y Documentación:** 1 hora por prueba (5 horas en total)

Cronograma de Trabajo:

- **Día 1-2:** Desarrollo de scripts de prueba (5 pruebas)
- **Día 3:** Ejecución de pruebas (5 pruebas)
- **Día 4:** Revisión de resultados y documentación

6. Plantillas para Cada Caso de Prueba

Plantilla de Caso de Prueba:

ID de Prueba: prueba_01

Objetivo: Iniciar sesión en Netflix y seleccionar el perfil 'Johan'.

Pasos:

- 1- Navegar a la página de inicio de sesión de Netflix.
- 2- Introducir las credenciales de inicio de sesión.
- 3- Hacer clic en el botón de inicio de sesión.
- 4- Seleccionar el perfil 'Johan'.
- 5- Tomar capturas de pantalla en la página de selección de perfil y en la página principal.

-Resultado Esperado: El sistema debería iniciar sesión correctamente, y el perfil 'Johan' debería ser seleccionado.

Plantilla de Caso de Prueba:

ID de Prueba: prueba_02

Objetivo: Buscar la serie 'Dexter' utilizando la barra de búsqueda de Netflix.

Pasos:

- 1- Estar en la página principal de Netflix.
- 2- Ingresar 'Dexter' en la barra de búsqueda.
- 3- Confirmar que los resultados de búsqueda incluyan la serie 'Dexter'.
- 4- Tomar una captura de pantalla de los resultados de búsqueda.

Resultado Esperado: La serie 'Dexter' debería aparecer en los resultados de búsqueda.

Plantilla de Caso de Prueba:

ID de Prueba:** prueba_03

Objetivo: Navegar por la categoría de 'Géneros' y seleccionar 'Acción'.

Pasos:

1. Estar en la página principal de Netflix.
2. Navegar al menú de 'Géneros'.
3. Seleccionar la categoría 'Acción'.
4. Confirmar que se muestran títulos de la categoría 'Acción'.
5. Tomar una captura de pantalla de los títulos listados.

Resultado Esperado: Deberían mostrarse correctamente los títulos de la categoría 'Acción'.

Plantilla de Caso de Prueba:

-ID de Prueba: prueba_04

Objetivo: Reproducir la serie en Netflix.

Pasos:

1. Estar en la página principal de Netflix.
2. Buscar o navegar hasta encontrar la serie 'SWAT'.
3. Hacer clic en 'Reproducir' para iniciar la serie.
4. Confirmar que el contenido comienza a reproducirse.
5. Tomar una captura de pantalla del contenido en reproducción.

-Resultado Esperado: La serie debería comenzar a reproducirse correctamente.

Plantilla de Caso de Prueba:

ID de Prueba: prueba_05

Objetivo: Ver los detalles de un contenido específico en Netflix.

Pasos:

1. Estar en la página principal de Netflix.
2. Seleccionar un contenido específico (por ejemplo, 'Stranger Things').
3. Ver los detalles del contenido seleccionado (sinopsis, elenco, etc.).
4. Confirmar que la información mostrada sea correcta.
5. Tomar una captura de pantalla de los detalles del contenido.

Resultado Esperado: Los detalles del contenido seleccionado deberían mostrarse correctamente.

7. Plantilla de Equipos de Pruebas y Responsabilidades

Plantilla de Equipos de Pruebas:

Equipo de Pruebas:

- **Responsable de Pruebas Automatizadas:** Encargado de escribir y ejecutar los scripts de prueba automatizados.
- **Analista de Calidad:** Responsable de revisar los resultados de las pruebas y documentar los hallazgos.
- **Desarrollador:** Encargado de solucionar problemas técnicos y ajustar los scripts de prueba según sea necesario.

8. Plan de Automatización de Pruebas

Plan de Automatización de Pruebas:

- **Objetivo:** Automatizar las pruebas funcionales para verificar el comportamiento del sistema en diferentes escenarios.
- **Alcance:** Pruebas de inicio de sesión, selección de perfil, búsqueda de contenido, navegación por géneros, reproducción de contenido, y visualización de detalles de contenido.
- **Herramientas:** Selenium WebDriver, WebDriverManager, Python.
- **Estrategia:** Desarrollar scripts de prueba para cada escenario, ejecutarlos en un entorno controlado, y revisar los resultados.
- **Cronograma:** Definido en la sección 5.

Conclusión

El proyecto de desarrollo de la plataforma de streaming inspirada en Netflix busca crear una experiencia de usuario de alta calidad mediante la implementación de funcionalidades esenciales como la autenticación, gestión de usuarios, búsqueda y navegación de contenido, y reproducción de videos. Utilizando la metodología Agile-Scrum, el equipo puede trabajar de manera colaborativa y eficiente, asegurando la entrega de un producto mínimo viable (MVP) que cumpla con las expectativas del usuario y esté listo para ser escalado y mejorado en futuras iteraciones. Con el uso de tecnologías modernas y herramientas adecuadas, el proyecto está bien posicionado para alcanzar sus objetivos y ofrecer una plataforma robusta y segura.

El proceso de automatización de pruebas para la plataforma de Netflix ha sido diseñado y documentado de manera integral para garantizar la calidad y funcionalidad del sistema. A continuación, se resumen los puntos clave del plan y las pruebas realizadas:

- 1.Requerimientos Funcionales y No Funcionales: Se definieron claramente los requerimientos para la automatización de pruebas, abarcando desde la capacidad de inicio de sesión hasta la reproducción de contenido y la navegación por géneros. Además, se consideraron aspectos no funcionales como tiempo de respuesta, interfaz de usuario, seguridad, compatibilidad y escalabilidad.
2. Criterios de Aceptación y Rechazo: Se establecieron criterios específicos para la aceptación y el rechazo de las pruebas, lo que permite una evaluación precisa del desempeño del sistema en cada escenario de prueba.
- 3.Herramientas de Pruebas: La elección de Selenium WebDriver, WebDriverManager y Python se justifica por su eficacia en la automatización de pruebas de aplicaciones web, simplificando la gestión de controladores y proporcionando un entorno de pruebas robusto.
4. Cronograma de Trabajo: El cronograma detallado permite una planificación efectiva del desarrollo, ejecución y revisión de las pruebas, asegurando una gestión del tiempo eficiente y cumplimiento de los objetivos establecidos.
5. Plantillas y Plan de Automatización: Se desarrollaron plantillas para los casos de prueba y la documentación del equipo de pruebas, facilitando la organización y estandarización de las pruebas. El plan de automatización proporciona una estrategia clara para el desarrollo y ejecución de las pruebas.
6. Ejecución y Documentación: La ejecución de pruebas debe seguir el plan establecido, con una revisión meticulosa de los resultados y documentación precisa para garantizar que todas las funcionalidades se verifiquen correctamente y cualquier problema se aborde de manera oportuna.