



Las Americas Institute of Technology

Asignatura

Programación 3

Profesor

Kelyn Tejada

Asunto

Proyecto Final

Nombre

Neftalí Liriano

Matrícula

2022-0437

Fecha

18-08-2023

Índice

Introducción	3
Desarrollo	4
Conclusiones	22
Bibliografías	23

Introducción

En este trabajo propongo una estrategia que seguirá un plan de pruebas meticuloso con el fin de brindar a los usuarios una herramienta muy eficiente para la gestión de personajes de anime de sus series favoritas. Para satisfacer este plan se eligió la metodología Agile-Scrum como el marco de trabajo que guiará este proceso, permitiendo la adaptación continua y la entrega incremental de funcionalidades valiosas.

La calidad del producto es un aspecto crucial en cualquier proyecto de software. Para asegurar que el ManagerAnime cumpla con los estándares de calidad esperados, se ha elaborado un plan de pruebas exhaustivo. Este plan contempla más de 20 casos de prueba obligatorios, diseñados específicamente para validar las funcionalidades definidas en las historias de usuario. Estos casos de prueba abarcan escenarios que van desde el registro de personajes y la cantidad de personajes existente y que sean borrados correctamente del sistema en cuestión.

Desarrollo

Requisitos Iniciales:



ManagerAnime

Las tecnologías que usaremos en este proyecto son las siguientes:

- SQLite, para la base de datos del proyecto
- C#, como lenguaje de programación
- DOTNET, como framework
- HTML
- Materialize CSS
- Leaflet Javascript

Objetivo del proyecto:

El objetivo del proyecto es poder facilitar a la mayor cantidad de usuarios una herramienta que permita almacenar información de personajes de series de anime, toda la información posible de cualquier personaje y que tengan todas las informaciones ordenadas en una pagina web aceptable que les permita ver un mapa de la ubicación del personaje, siendo de ayuda para quienes quieren investigar acerca de un personaje y quieren tener su información de forma rápida e intuitiva.

Alcance del proyecto:

El proyecto tendrá las funciones básicas de cómo es el caso de:

- Agregar todos los datos para ser guardado en una base de datos sencilla
- Cuenta con la función de modificar los datos existentes de un personaje
- Tiene la función de eliminar la información de un personaje
- La función de exportar los reportes de un personaje en un formato ordenado
- Visualizar la ubicación de un personaje en un mapa
- . Cuenta con un listado para visualizar los personajes

El proyecto no cuenta con las siguientes funciones:

- No cuenta con un login para los usuarios
- No posee una base de datos en línea, los datos se guardan de forma local
- No cuenta con un buscador con filtro para encontrar a los personajes

Cronograma del Proyecto

Etapa del Proyecto	Duración Estimada
Definición de Requerimientos	1 semana
Diseño y Desarrollo	4 semanas
Pruebas	3 semanas
Ajustes y Correcciones	1 semana
Implementación y Despliegue	2 semanas
Evaluación y Cierre	1 semana
Total	12 semanas

Release

El sistema en su primer release podrá permitirle al usuario acceder a un menú con diferentes opciones como son el de registro de personajes, modificar personajes o eliminar personajes y provee de un listado de personajes filtrado por distintas opciones. También tendrá la función de exportar la información suministrada a un HTML o poder visualizar los datos del personaje en un mapa.

Para los requerimientos, el sistema deberá registrar los datos suministrados en una base de datos valiéndose de un lenguaje de programación, en cuanto a los reportes requerirá del uso de tecnologías como HTML para una vista ordenada en la web y finalmente el uso de una librería o framework de Javascript para el reporte en mapa.

Equipo de metodología Scrum:

Epicas:

1. Gestión de Personajes: Incluye la funcionalidad básica para agregar, editar y eliminar personajes.
2. Reportes y Análisis: Agrupa las historias relacionadas con la generación de informes y análisis de datos de los personajes.

Historias de Usuario:

1. Como usuario, quiero poder agregar un nuevo personaje de anime con su nombre, apellido, foto, pronunciación, fecha de nacimiento y habilidad.
2. Como usuario, quiero poder editar los detalles de un personaje existente, incluyendo su nombre, foto, pronunciación y habilidad.
3. Como usuario, quiero poder eliminar un personaje de anime de la lista.
4. Como usuario, quiero ver una lista de todos los personajes de anime registrados con su información básica.
5. Como usuario, quiero poder ver los personajes ordenados por signo zodiacal.
6. Como usuario, quiero recibir un reporte de los personajes cuyos cumpleaños caen en un mes específico.
7. Como usuario, quiero ver un mapa con marcadores de la ubicación de los personajes en función de sus coordenadas de latitud y longitud.
8. Como usuario, quiero exportar los detalles de un personaje a un archivo HTML.
9. Como usuario, quiero ver un listado de personajes agrupados por serie de anime.
10. Como usuario, quiero un informe de personajes clasificados por estado (activo, inactivo, etc.).

Criterios de Aceptación:

- Los usuarios pueden agregar un nuevo personaje con todos los campos requeridos.
- La edición de un personaje actualiza correctamente su información.
- Los usuarios pueden eliminar un personaje de la lista.

- La lista de personajes muestra detalles básicos como nombre, apellido y serie de anime.
- Los personajes se ordenan adecuadamente por signo zodiacal y mes de cumpleaños.
- El mapa muestra marcadores en las ubicaciones especificadas por las coordenadas.
- Los detalles del personaje se exportan a un archivo HTML que se puede abrir en un navegador.
- El informe por serie de anime muestra los personajes agrupados por su respectiva serie.
- El informe por estado muestra los personajes agrupados por su estado.

Puntos por Historia:

1. 8 puntos
2. 5 puntos
3. 5 puntos
4. 8 puntos
5. 5 puntos
6. 5 puntos
7. 8 puntos
8. 8 puntos
9. 5 puntos
10. 5 puntos

Tareas:

Historia de Usuario 1: Agregar un nuevo personaje de anime:

- Crear un programa que permita al usuario ingresar los detalles del nuevo personaje, como nombre, apellido, foto, pronunciación, fecha de nacimiento, habilidad, serie de anime, sexo y estado.
- Validar los campos de entrada para asegurarse de que se proporcionen los datos requeridos y tengan el formato correcto.

- Guardar la información ingresada en la base de datos.
- Mostrar un mensaje de confirmación al usuario después de agregar exitosamente el nuevo personaje.
- Manejar posibles errores o excepciones durante el proceso de agregar el personaje.

Historia de Usuario 2: Editar los detalles de un personaje existente:

- Mostrar una lista de personajes existentes para que el usuario elija uno para editar.
- Desplegar un formulario con los detalles actuales del personaje elegido
- Permitir al usuario modificar los campos necesarios, como nombre, foto, pronunciación, habilidad, etc.
- Validar los cambios realizados por el usuario antes de guardarlos en la base de datos.
- Actualizar los datos del personaje en la base de datos con la información editada.
- Notificar al usuario cuando los cambios se hayan guardado exitosamente.
- Manejar posibles errores o excepciones durante el proceso de edición.

Historia de Usuario 3: Eliminar un personaje de anime:

- Mostrar una lista de personajes existentes para que el usuario seleccione uno para eliminar.
- Solicitar confirmación al usuario antes de eliminar el personaje.
- Eliminar el personaje seleccionado de la base de datos.
- Mostrar un mensaje de confirmación al usuario después de eliminar exitosamente el personaje.
- Manejar posibles errores o excepciones durante el proceso de eliminación.

Historia de Usuario 4: Ver una lista de personajes de anime:

- Recuperar la lista de personajes de la base de datos.
- Mostrar los detalles básicos de cada personaje, como nombre, apellido y serie de anime, en una interfaz de usuario.
- Permitir al usuario desplazarse por la lista para ver todos los personajes.
- Proporcionar un botón para regresar a la pantalla principal.

Historia de Usuario 5: Ordenar personajes por signo zodiacal:

- Implementar la lógica para calcular el signo zodiacal en función de la fecha de nacimiento de cada personaje.
- Agrupar los personajes en categorías según su signo zodiacal.
- Mostrar una lista de personajes ordenados por signo zodiacal en la interfaz de usuario.
- Proporcionar la opción de seleccionar un signo zodiacal específico para ver los personajes relacionados.
- Permitir al usuario regresar a la lista completa.

Historia de Usuario 6: Generar un informe de cumpleaños:

- Solicitar al usuario que seleccione un mes específico para el informe.
- Filtrar la lista de personajes para incluir solo aquellos cuyos cumpleaños caen en el mes seleccionado.
- Mostrar una lista de personajes con detalles como nombre, apellido, fecha de nacimiento y edad.
- Proporcionar al usuario la opción de imprimir o exportar el informe en un formato legible.

Historia de Usuario 7: Ver un mapa con marcadores de personajes:

- Recuperar las coordenadas de latitud y longitud de cada personaje de la base de datos.
- Utilizar una biblioteca de mapas (por ejemplo, Leaflet) para mostrar un mapa interactivo en la interfaz de usuario.
- Colocar marcadores en las ubicaciones correspondientes a las coordenadas de cada personaje en el mapa.
- Mostrar información relevante sobre el personaje cuando se hace clic en un marcador.
- Permitir al usuario explorar el mapa y los marcadores.

Historia de Usuario 8: Exportar detalles de un personaje a un archivo HTML:

- Recopilar los detalles del personaje seleccionado de la base de datos.
- Crear una estructura HTML con estilos y formato para mostrar los detalles.
- Llenar la plantilla HTML con los detalles del personaje, como nombre, apellido, foto, serie de anime, etc.

- Generar un archivo HTML con los datos del personaje.
- Proporcionar al usuario el archivo HTML.

Historia de Usuario 9: Ver un listado de personajes agrupados por serie de anime:

- Recuperar la lista de series de anime y la información de los personajes relacionados de la base de datos.
- Mostrar una lista de series de anime en la interfaz de usuario.
- Permitir al usuario seleccionar una serie para ver los personajes asociados a ella.
- Mostrar los detalles básicos de los personajes de esa serie en una lista.

Historia de Usuario 10: Generar un informe de personajes por estado:

- Filtrar la lista de personajes en función de su estado (activo, inactivo, etc.).
- Mostrar una lista de personajes con detalles como nombre, apellido, serie de anime y estado.
- Proporcionar al usuario la opción de imprimir o exportar el informe en un formato legible.

Equipo de Trabajo (Roles y Habilidades):

- Scrum Master: Encargado de liderar las ceremonias Scrum y asegurar que el equipo siga la metodología correctamente.
- Desarrollador Frontend: Habilidad en desarrollo de interfaces de usuario utilizando tecnologías web como HTML, CSS y JavaScript.
- Desarrollador Backend: Experiencia en desarrollo de lógica de negocios y bases de datos utilizando lenguajes como C# y tecnologías como Entity Framework.
- Scrum Team: Habilidad en la optimización de las historias de usuario y la realización de sprint del proyecto.
- Tester/QA: Experiencia en pruebas de software para garantizar que la aplicación funcione correctamente.

Las Herramientas para usar en el proyecto son:

- Azure DevOps
- Github
- Slack
- Visual Studio Code

Fechas de Ceremonias Scrum:

- Sprint Planning: 2023-08-18
- Daily Standup: Todos los días a las 9:00 AM
- Sprint Review: 2023-09-10
- Sprint Retrospective: 2023-09-11

Plan de Pruebas:

1. Lista de Requerimientos Funcionales y No Funcionales

Requerimientos Funcionales:

1. Agregar un nuevo personaje de anime.
2. Editar los detalles de un personaje existente.
3. Eliminar un personaje de anime.
4. Ver una lista de personajes de anime.
5. Ordenar personajes por signo zodiacal.
6. Generar un informe de cumpleaños.
7. Ver un mapa con marcadores de personajes.
8. Exportar detalles de un personaje a un archivo HTML.
9. Ver un listado de personajes agrupados por serie de anime.
10. Generar un informe de personajes por estado.

Requerimientos No Funcionales:

1. La aplicación debe ser fácil de usar y tener una interfaz intuitiva.
2. Los datos de los personajes deben almacenarse de manera segura en la base de datos.
3. La aplicación debe funcionar correctamente en los navegadores web más comunes.
4. Las respuestas de la aplicación deben ser rápidas y eficientes.
5. La aplicación debe ser escalable para manejar un aumento en el número de personajes y usuarios.

1.2 Casos de Prueba

1. Pruebas de Registro de Personajes:

1. Registro Exitoso:

- Descripción: Verificar que un nuevo personaje pueda ser registrado exitosamente en el sistema.
- Pasos:
 1. Acceder a la opción de "Agregar Personaje" en la interfaz de gestión.
 2. Completar todos los campos requeridos con datos válidos.
 3. Confirmar el registro.
 4. Verificar que el personaje se haya agregado correctamente en la lista de personajes.

2. Registro con Datos Faltantes:

- Descripción: Comprobar que el sistema detecta y maneja adecuadamente el registro de personajes con campos obligatorios faltantes.
- Pasos:
 1. Acceder a la opción de "Agregar Personaje".
 2. Dejar al menos un campo obligatorio en blanco.

3. Intentar confirmar el registro.
4. Verificar que se muestre un mensaje de error indicando los campos faltantes.

3. Registro con Fecha de Nacimiento Inválida:

- Descripción: Validar que el sistema rechaza el registro de personajes con fechas de nacimiento en el futuro.
- Pasos:
 1. Acceder a la opción de "Agregar Personaje".
 2. Ingresar una fecha de nacimiento futura.
 3. Intentar confirmar el registro.
 4. Verificar que se muestre un mensaje de error indicando que la fecha es inválida.

2. Pruebas de Edición de Personajes:

4. Edición Exitosa de Datos:

- Descripción: Asegurarse de que se puedan editar los datos de un personaje existente.
- Pasos:
 1. Acceder a la opción de "Editar Personaje".
 2. Seleccionar un personaje de la lista.
 3. Modificar al menos un campo de datos.
 4. Confirmar la edición.
 5. Verificar que los cambios se reflejen correctamente en la información del personaje.

5. Edición de Personaje Inexistente:

- Descripción: Comprobar que el sistema maneja adecuadamente la edición de un personaje que no existe.
- Pasos:
 1. Acceder a la opción de "Editar Personaje".
 2. Ingresar un ID de personaje inválido.
 3. Intentar confirmar la edición.

4. Verificar que se muestre un mensaje de error indicando que el personaje no existe.

3. Pruebas de Eliminación de Personajes:

6. Eliminación Exitosa de Personaje:

- Descripción: Confirmar que un personaje se puede eliminar correctamente del sistema.
- Pasos:
 1. Acceder a la opción de "Eliminar Personaje".
 2. Seleccionar un personaje de la lista.
 3. Confirmar la eliminación.
 4. Verificar que el personaje ya no aparece en la lista de personajes.

7. Eliminación de Personaje Inexistente:

- Descripción: Validar que el sistema maneja la eliminación de un personaje que no existe.
- Pasos:
 1. Acceder a la opción de "Eliminar Personaje".
 2. Ingresar un ID de personaje inválido.
 3. Intentar confirmar la eliminación.
 4. Verificar que se muestre un mensaje de error indicando que el personaje no existe.

4. Pruebas de Reportes:

8. Generación de Listado de Personajes:

- Descripción: Verificar que se pueda generar un listado de todos los personajes registrados.
- Pasos:
 1. Acceder a la opción de "Reporte de Personajes".
 2. Verificar que se muestre una lista completa con información de cada personaje.

9. Generación de Reporte por Signo Zodiacal:

- Descripción: Asegurarse de que el sistema genere correctamente un reporte de personajes por signo zodiacal.

- Pasos:
 1. Acceder a la opción de "Reporte por Signo Zodiacal".
 2. Verificar que se muestre la información de los personajes agrupada por signo zodiacal.

5. Pruebas de Mapa de Personajes:

10. Visualización de Mapa de Personajes:

- Descripción: Comprobar que el sistema muestre correctamente la ubicación de los personajes en un mapa.
- Pasos:
 1. Acceder a la opción de "Mapa de Personajes".
 2. Verificar que los personajes se muestren en sus ubicaciones correspondientes.

11. Visualización de Detalles del Personaje en el Mapa:

- Descripción: Asegurarse de que se puedan ver los detalles de un personaje al hacer clic en su marcador en el mapa.
- Pasos:
 1. Acceder al "Mapa de Personajes".
 2. Hacer clic en el marcador de un personaje.
 3. Verificar que se muestre una ventana emergente con los detalles del personaje.

6. Pruebas de Autenticación:

12. Inicio de Sesión Exitoso:

- Descripción: Verificar que los usuarios puedan iniciar sesión correctamente en el sistema.
- Pasos:
 1. Acceder a la pantalla de inicio de sesión.
 2. Ingresar un nombre de usuario y contraseña válidos.
 3. Confirmar el inicio de sesión.
 4. Verificar que se redirige al usuario a la página principal.

13. Inicio de Sesión con Credenciales Incorrectas:

- Descripción: Comprobar que el sistema maneja adecuadamente el inicio de sesión con credenciales incorrectas.
- Pasos:
 1. Acceder a la pantalla de inicio de sesión.
 2. Ingresar un nombre de usuario y contraseña incorrectos.
 3. Intentar iniciar sesión.
 4. Verificar que se muestre un mensaje de error indicando que las credenciales son incorrectas.

7. Pruebas de Permisos y Roles:

14. Asignación de Rol Exitosa:

- Descripción: Asegurarse de que un administrador pueda asignar un rol a un usuario.
- Pasos:
 1. Acceder a la opción de "Gestión de Roles y Permisos".
 2. Seleccionar un usuario.
 3. Asignar un rol específico al usuario.
 4. Verificar que el rol se asignó correctamente al usuario.

15. Acceso a Funcionalidad Restringida por Rol:

- Descripción: Verificar que los usuarios solo puedan acceder a las funcionalidades permitidas por su rol.
- Pasos:
 1. Iniciar sesión con diferentes roles de usuario (administrador, usuario regular, etc.).
 2. Intentar acceder a funcionalidades restringidas por rol.
 3. Verificar que las funcionalidades se muestran o bloquean según el rol del usuario.

8. Pruebas de Seguridad:

16. Inyección de Código SQL:

- Descripción: Intentar realizar una inyección de código SQL en los campos de entrada para verificar la seguridad del sistema.
- Pasos:
 1. Acceder a diferentes campos de entrada (nombres, contraseñas, etc.).

2. Ingresar código SQL malicioso.
3. Intentar enviar los datos.
4. Verificar que el sistema detecta y bloquea la inyección de código SQL.

17. Ataque de Fuerza Bruta en Contraseñas:

- Descripción: Realizar intentos repetidos de inicio de sesión con contraseñas incorrectas para evaluar la protección contra ataques de fuerza bruta.
- Pasos:
 1. Acceder a la pantalla de inicio de sesión.
 2. Ingresar contraseñas incorrectas varias veces.
 3. Verificar que el sistema bloquea temporalmente el acceso después de un número definido de intentos fallidos.

9. Pruebas de Rendimiento:

18. Carga de Personajes en la Lista:

- Descripción: Evaluar el tiempo de respuesta y la velocidad de carga al mostrar la lista completa de personajes.
- Pasos:
 1. Acceder a la opción de "Lista de Personajes".
 2. Verificar que la lista se carga de manera rápida y sin retrasos significativos.

19. Generación de Reporte Complejo:

- Descripción: Evaluar el tiempo necesario para generar un reporte complejo, como el reporte de personajes por signo zodiacal.
- Pasos:
 1. Acceder a la opción de "Reporte por Signo Zodiacal".
 2. Verificar que el reporte se genera en un tiempo razonable.

10. Pruebas de Integración:

20. Integración con Sistema de Autenticación Externo:

- Descripción: Verificar que el sistema se integre correctamente con un sistema de autenticación externo (por ejemplo, autenticación de Google).
- Pasos:
 1. Iniciar sesión utilizando las credenciales del sistema de autenticación externo.

2. Verificar que el sistema permita el acceso y cree el perfil correspondiente en la base de datos.

2. Criterios de Aceptación de las Pruebas

Para cada Historia de Usuario, los criterios de aceptación son:

- Los datos ingresados en la interfaz de usuario se guardan correctamente en la base de datos.
- Los cambios realizados en la edición de personajes se reflejan con precisión en la base de datos.
- Los personajes eliminados se eliminan definitivamente de la base de datos.
- Las listas de personajes, informes y mapas se generan y muestran correctamente en la interfaz.
- Los archivos HTML exportados contienen la información correcta del personaje.
- Los criterios de filtrado y agrupación se aplican correctamente en los informes.

3. Criterios de Rechazo en las Pruebas

- Cualquier dato ingresado no se guarda en la base de datos.
- Los cambios realizados en la edición de personajes no se actualizan en la base de datos.
- Los personajes eliminados siguen apareciendo en las listas.
- Los informes y mapas no se generan correctamente o muestran información incorrecta.
- Los archivos HTML exportados están vacíos o contienen información incorrecta.

4. Herramientas de Pruebas

- **Pruebas Manuales:** Para las pruebas manuales, utilizaremos herramientas como Microsoft Excel para registrar los casos de prueba y documentar los resultados.
- **Pruebas de Automatización:** Utilizaremos Moq para la automatización de pruebas de interfaz de usuario y XUnit como marco de prueba

5. Estimación del Tiempo de Ejecución de Pruebas

El tiempo estimado para la ejecución de pruebas varía según la complejidad de las historias de usuario y la cantidad de casos de prueba. Se estima que ambas pruebas tomarán aproximadamente 12 semanas y se estarán realizando a la par de la otra para mayor eficiencia.

Cronograma de Trabajo del Plan de Pruebas:

Semana	Actividad
Semana 1	Definición de casos de prueba para HU 1-3
	Creación de scripts de prueba automatizados
Semana 2	Ejecución de pruebas manuales para HU 1-3
	Ejecución de pruebas automatizadas para HU 1-3
Semana 3	Definición de casos de prueba para HU 4-6
	Ejecución de pruebas manuales para HU 4-6
Semana 4	Ejecución de pruebas automatizadas para HU 4-6
	Evaluación y documentación de resultados
Semana 5	Definición de casos de prueba para HU 7-9
	Ejecución de pruebas manuales para HU 7-9
Semana 6	Ejecución de pruebas automatizadas para HU 7-9
	Evaluación y documentación de resultados
Semana 7	Definición de casos de prueba para HU 10
	Ejecución de pruebas manuales para HU 10
Semana 8	Ejecución de pruebas automatizadas para HU 10
	Evaluación y documentación de resultados
Semana 9	Revisión y ajustes de casos de prueba
	Generación de informes finales de pruebas
Semana 10	Preparación de la demostración del plan de automatización
	Ejecución y demostración del plan de automatización
Semana 11	Evaluación y revisión de la efectividad del plan de pruebas
Semana 12	Finalización del informe de pruebas
	Entrega del informe de pruebas y cierre

6. Plantillas para Casos de Prueba

- Plantilla para Pruebas Manuales:
 - Nombre del caso de prueba.
 - Descripción del caso de prueba.
 - Pasos para seguir.
- Plantilla para Pruebas de Automatización (Moq y XUnit):
 - Escenario de prueba.
 - Script de automatización.
 - Datos de entrada.

- Verificación de resultados.

7. Equipos de Pruebas y Responsabilidades

- Líder de Pruebas: Coordinación y supervisión de las actividades de pruebas.
- Analista de Pruebas: Diseño y documentación de casos de prueba.
- Ingeniero de Automatización: Desarrollo y mantenimiento de scripts de prueba automatizados.
- Equipo de Desarrollo: Corrección de problemas identificados durante las pruebas.

8. Plan de Automatización de Pruebas

- Identificar los escenarios de prueba adecuados para la automatización.
- Desarrollar scripts de prueba automatizados utilizando Moq y XUnit.
- Ejecutar y depurar los scripts de prueba automatizados.
- Demostrar los resultados de las pruebas automatizadas.

Conclusiones

En conclusión, el desarrollo de un plan de pruebas exhaustivo y detallado es esencial para garantizar la calidad y la fiabilidad de cualquier tipo de sistema. A lo largo de este trabajo, se ha abordado la importancia de la planificación y ejecución de pruebas para validar que el sistema cumpla con los requerimientos funcionales y no funcionales, así como para identificar y corregir posibles errores y vulnerabilidades antes de su implementación.

El proceso de creación de un plan de pruebas ha involucrado varios pasos clave:

1. **Definición de Historias de Usuario (HU):** Establecer los requisitos funcionales y no funcionales a través de las HU es el punto de partida. Estas historias de usuario proporcionan la base para desarrollar casos de prueba significativos y realistas que reflejen las necesidades del usuario.
2. **Identificación de Casos de Prueba:** Los casos de prueba son escenarios específicos que evalúan el comportamiento del sistema en función de las HU. En este plan, se han definido 20 casos de prueba que cubren diversas áreas del sistema, desde el registro de personajes hasta la seguridad y el rendimiento.
3. **Criterios de Aceptación y Rechazo:** Establecer criterios claros de aceptación y rechazo para cada caso de prueba es esencial para determinar si el sistema funciona según lo esperado. Los criterios de aceptación definen los resultados deseados, mientras que los criterios de rechazo identifican los escenarios problemáticos.
4. **Herramientas de Pruebas:** La elección de las herramientas de pruebas es crucial para llevar a cabo las pruebas de manera eficiente y efectiva. La elección debe basarse en las necesidades del proyecto y la experiencia del equipo. Se recomienda justificar la elección de las herramientas utilizadas.
5. **Estimación de Tiempo y Cronograma:** Elaborar un cronograma de trabajo detallado para la ejecución de las pruebas es fundamental para garantizar una planificación adecuada de los recursos y el tiempo. La estimación del tiempo debe ser realista y considerar el alcance y la complejidad de las pruebas.
6. **Plantillas y Documentación:** El uso de plantillas estandarizadas para la documentación de casos de prueba, equipos de pruebas y otros aspectos facilita la comunicación y la comprensión en todo el equipo. La documentación adecuada también es esencial para mantener un registro claro de los resultados de las pruebas.
7. **Automatización de Pruebas:** La automatización de pruebas puede aumentar la eficiencia y la cobertura de las pruebas. Sin embargo, es importante seleccionar cuidadosamente qué casos de prueba deben automatizarse y cuáles deben realizarse manualmente.

Bibliografías

<https://github.com/Neta2603/Manageranime/tree/main>

[Proyecto Final Prog 3](#)

<https://dev.azure.com/20220437/Manageranime>