

Introducción

En este emocionante proyecto, exploraremos lo aprendido en el curso de Fundamentos A&P I mientras construimos una herramienta simple pero poderosa para la organización de tareas.

En el ajetreo y el bullicio de la vida diaria, a menudo es difícil realizar un seguimiento de todas las tareas pendientes. Ya sea para la gestión de proyectos, la planificación del trabajo o simplemente para mantener un registro de las cosas por hacer, una lista de tareas eficiente puede marcar la diferencia.

Equipo de Trabajo:

Laura Melisa Álvarez Úsuga.
Yusleidy Blandón Caicedo.
Julian Renteria Asprilla.

Recursos y Soporte:

Usamos **Trello** como Plataforma de gestión de proyectos he aquí el enlace para acceder a nuestro tablero: <https://trello.com/b/wC5ZOJSP/reto-final-mod2-c13>

Cronograma y Entregables:

Cronograma a alto nivel

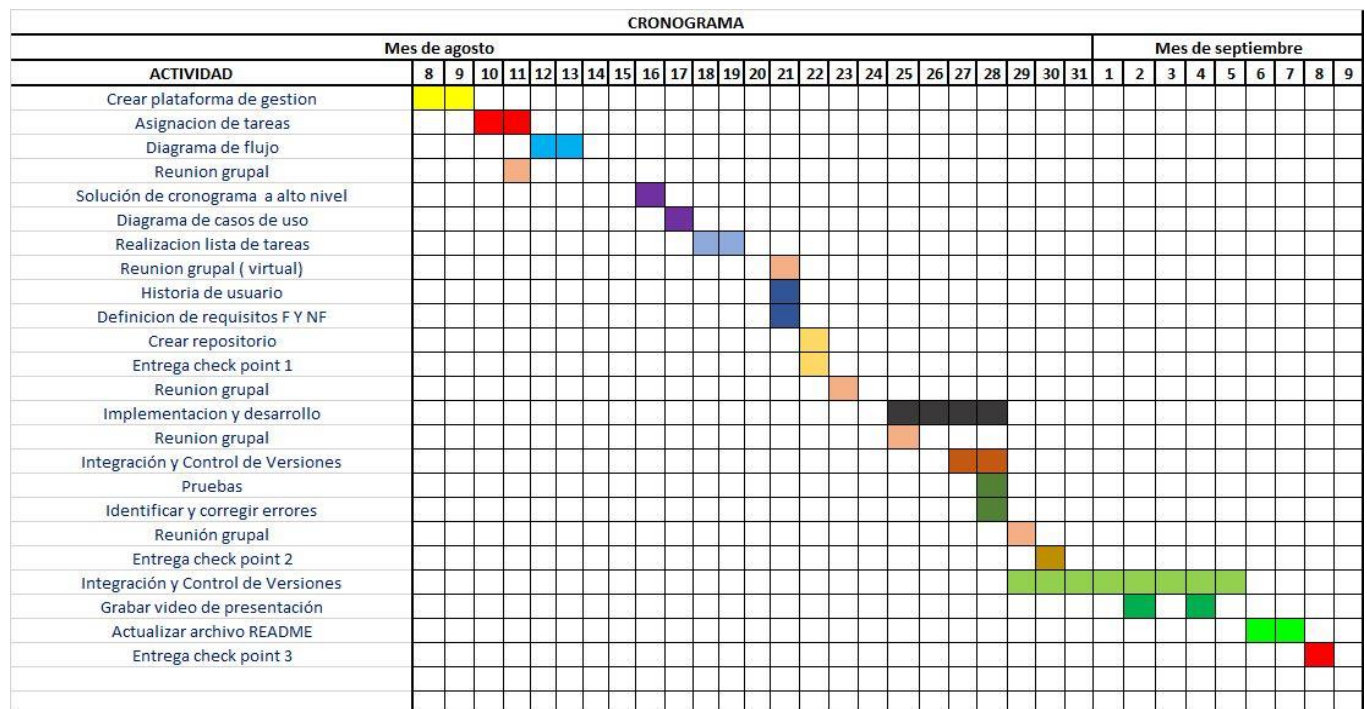
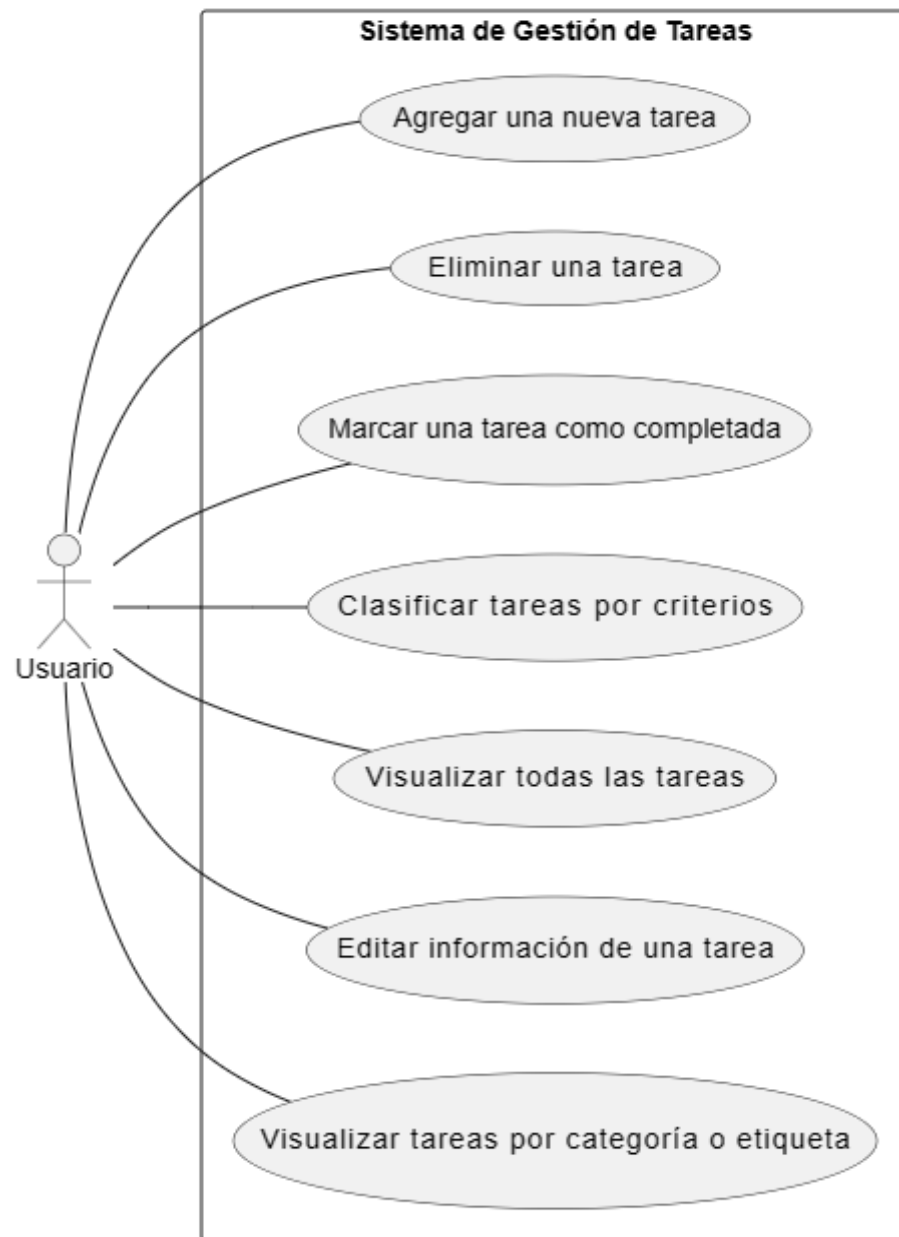
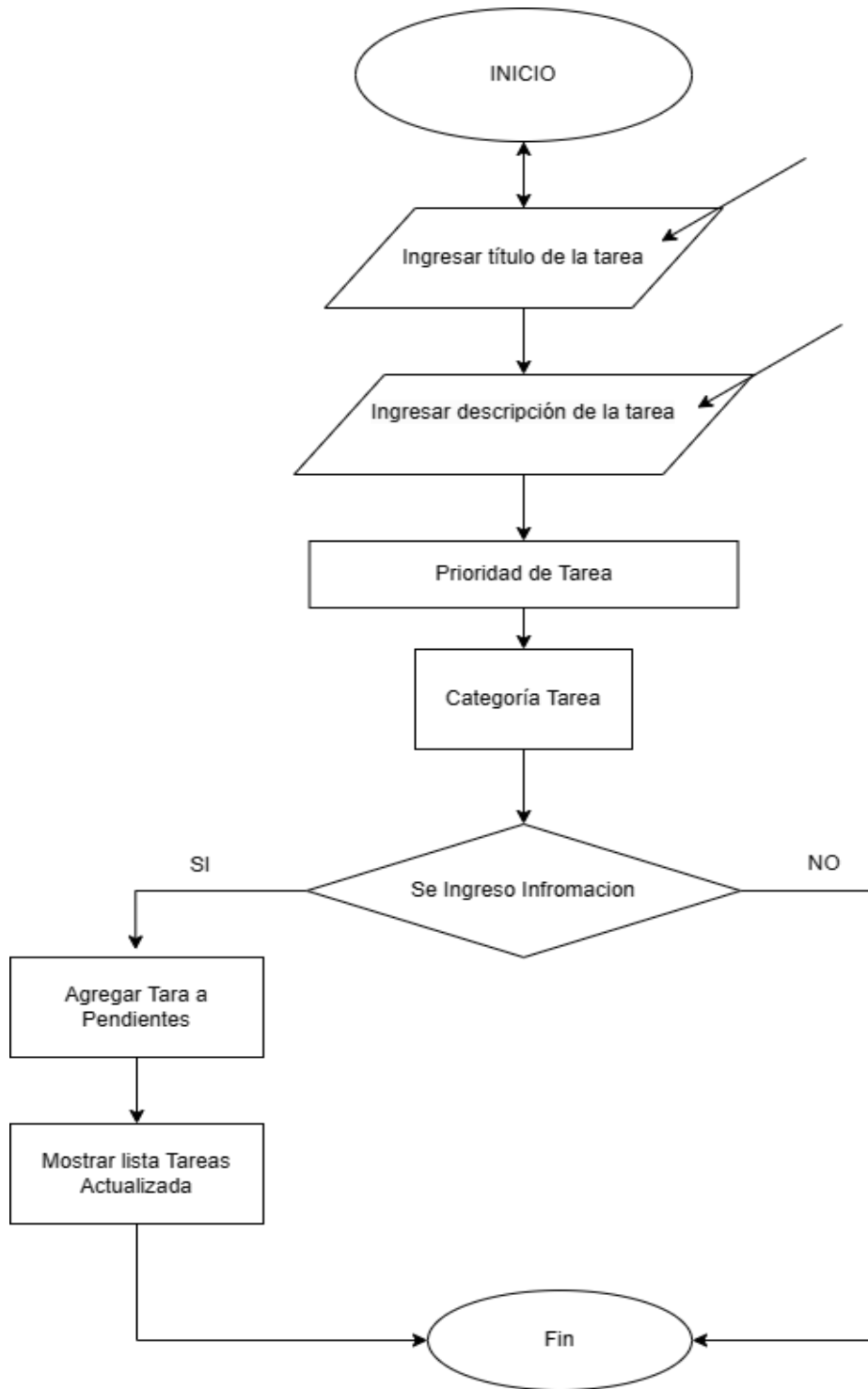


Diagrama Casos De Uso

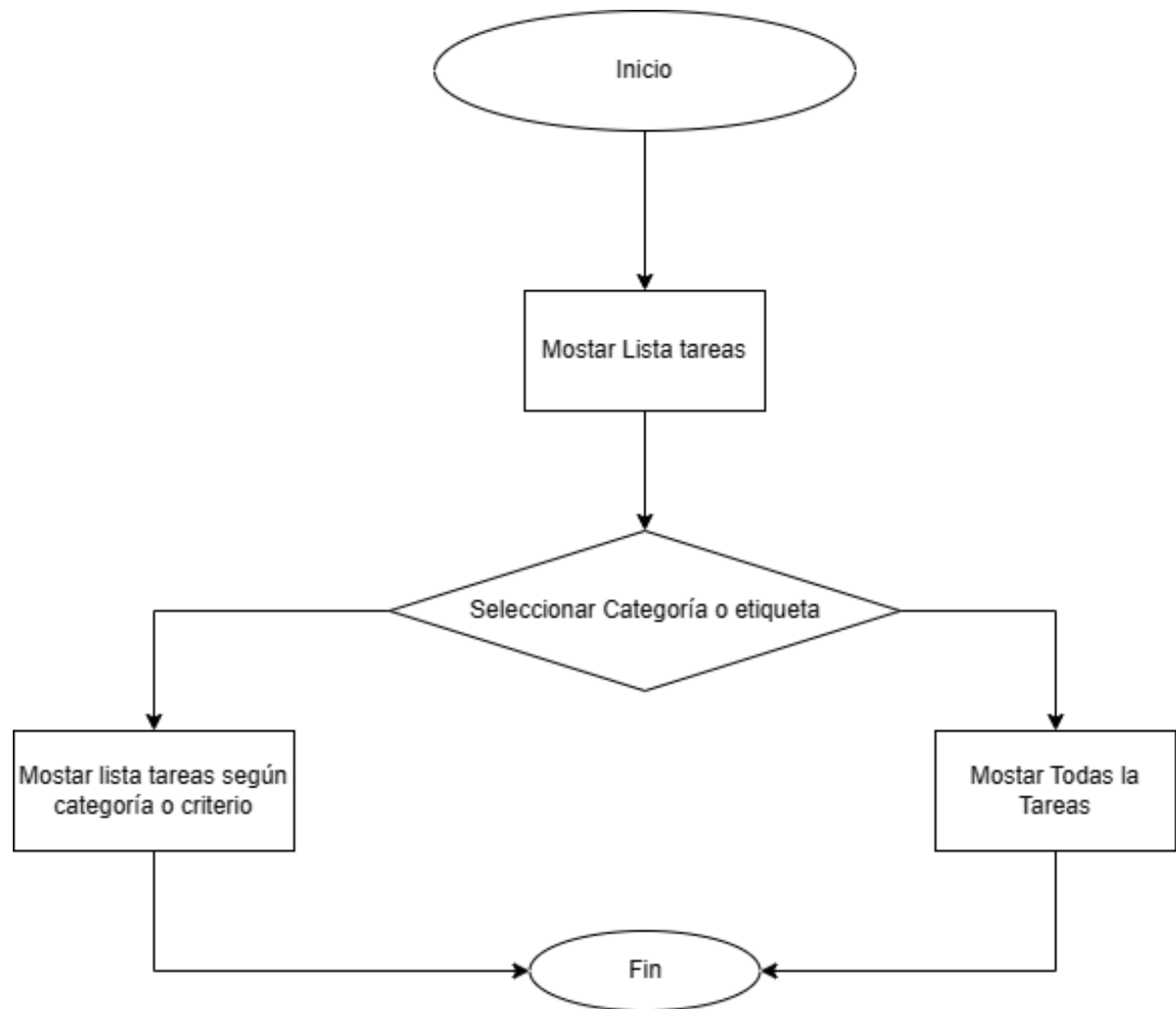


Diagramas De Flujo.

Agregar tarea



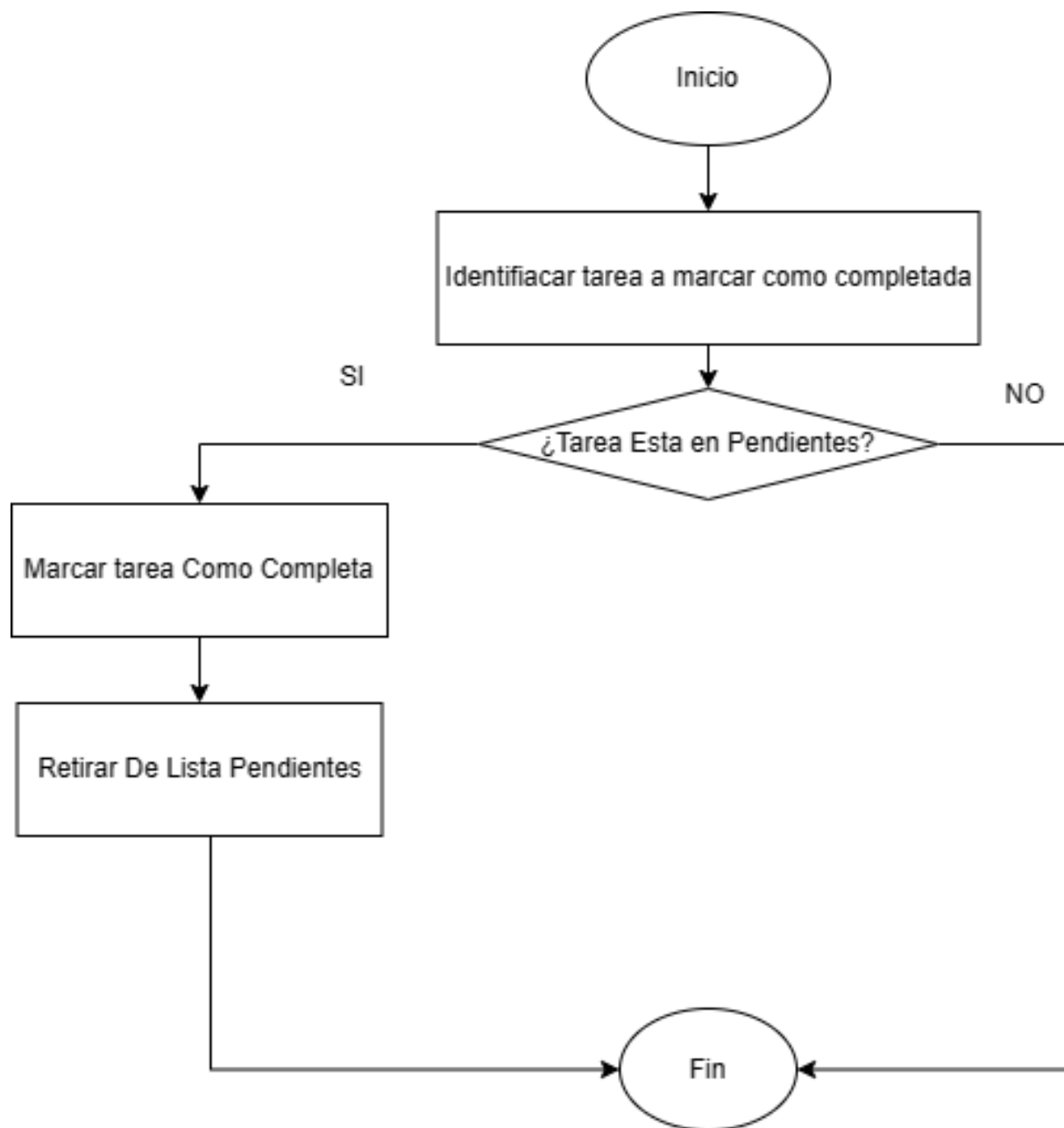
Visualizar tareas por categoría o etiqueta



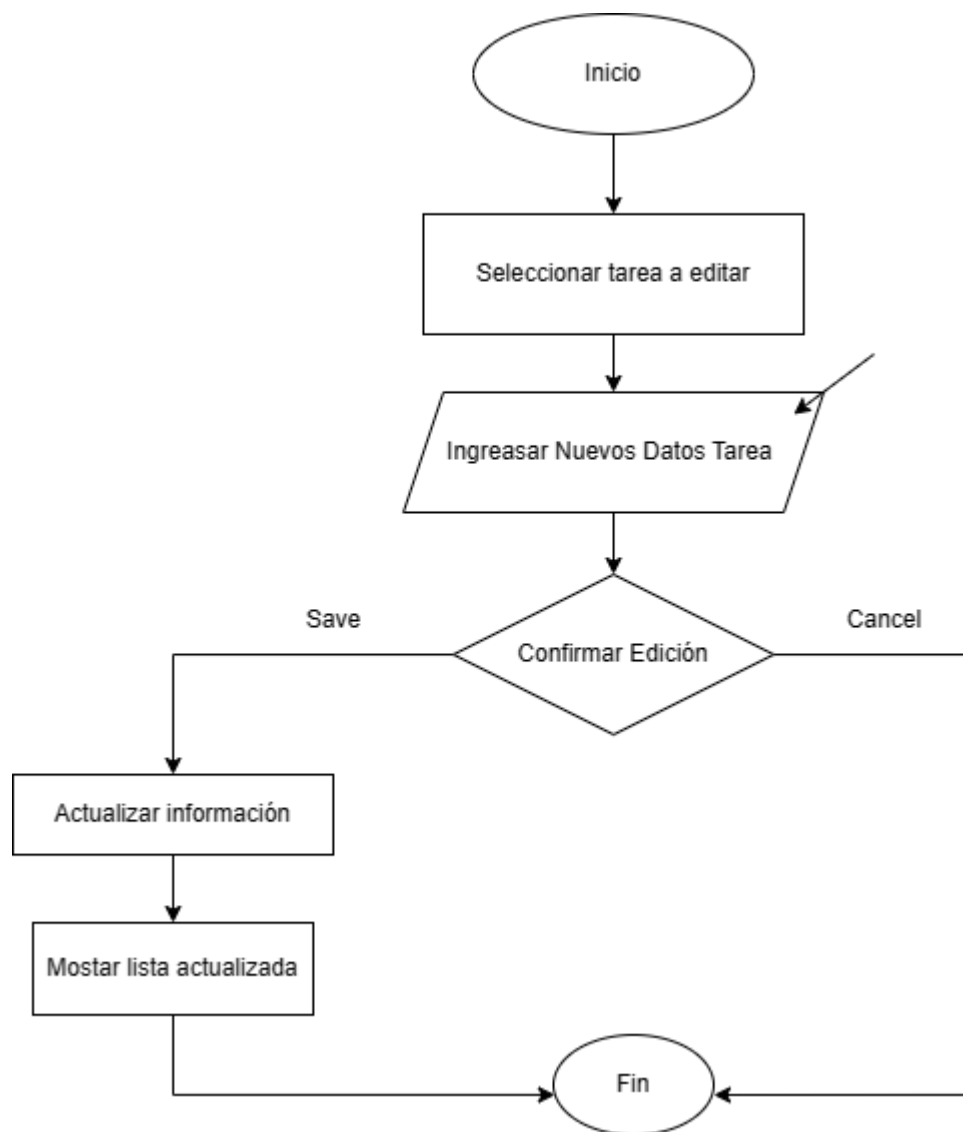
Eliminar Tarea



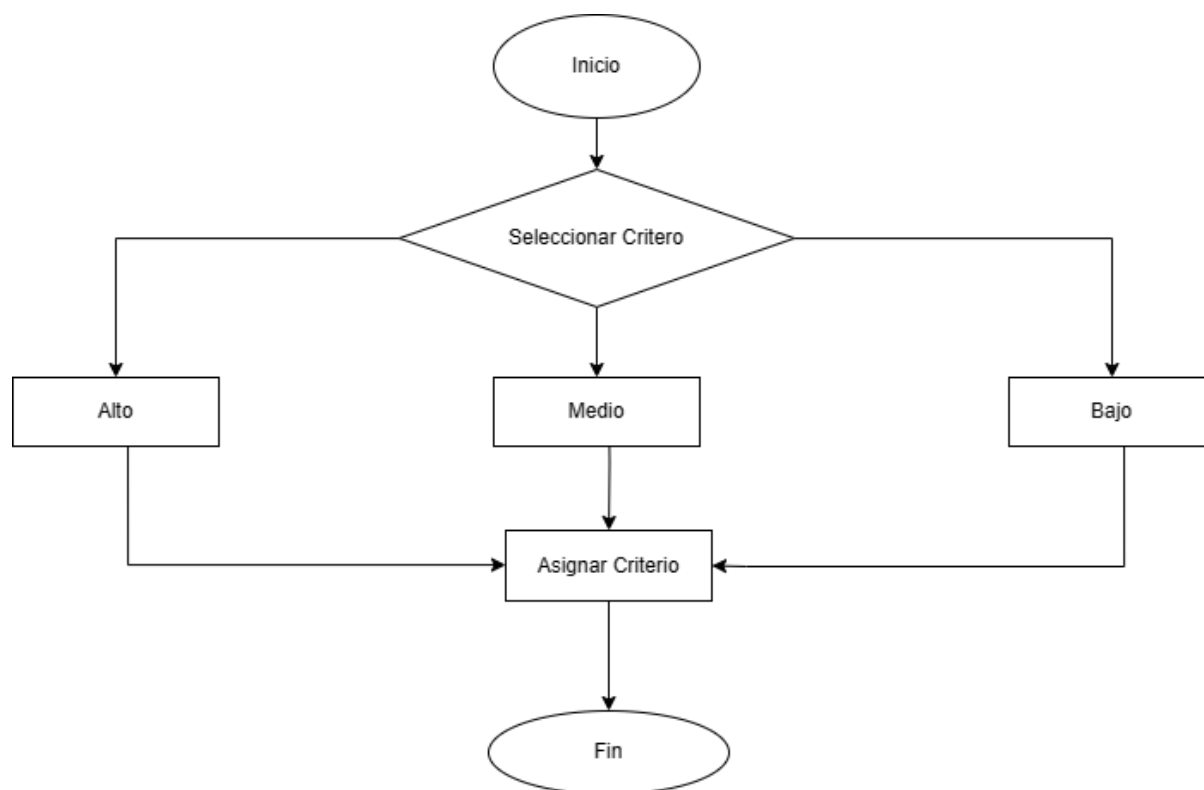
Marcar como Completada



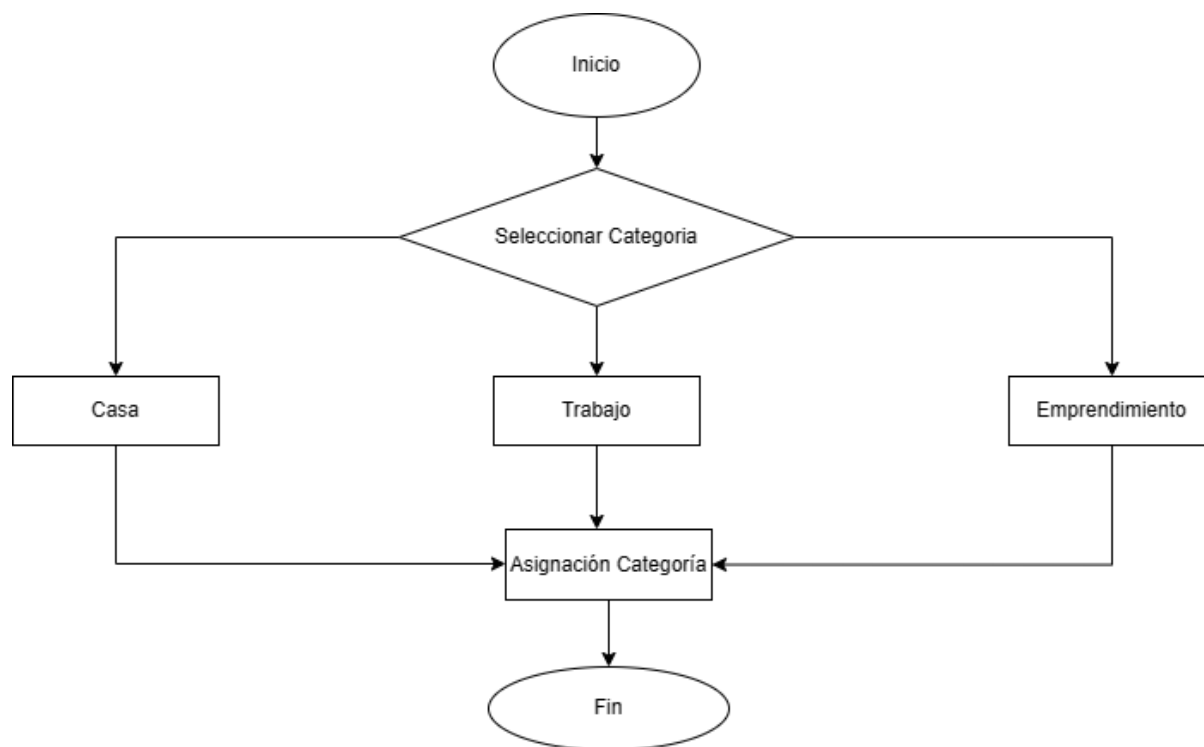
Editar información de tarea



Asignación Criterio



Asignación Categoría



Historias de usuarios:

Historia de Usuario					
Nombre de la Historia	Descripción	Prioridad	Estado	Categoría	Estimación de Tiempo
Agregar nueva tarea	Permitir a los usuarios crear tareas ingresando detalles como título, descripción, fecha de vencimiento y prioridad.	Alta	Realizado	Tareas	2 Horas
Marcar como completa	Habilitar la opción para que los usuarios indiquen que una tarea ha sido completada.	Media	Realizado	Funcionalidades	3 Horas
Editar tarea	Permitir a los usuarios modificar detalles de tareas existentes como descripción, o prioridad.	Alta	Realizado	Tareas	2 Horas
Eliminar tareas	Proporcionar la opción de eliminar tareas que ya no son necesarias o relevantes.	Media	Realizado	Funcionalidades	1.5 Horas
Visualizar todas las tareas	Mostrar una lista completa de todas las tareas para facilitar su gestión y revisión.	Media	Realizado	Interfaz	2 Horas
Categorizar tareas	Etiquetar las tareas con categorías para una organización más clara y permitir a los usuarios filtrar por temas.	Alta	Realizado	Mejoras	1.5 Horas

Cómo lo Hicimos:

- **Identificamos Funcionalidades Clave:** Nos sentamos juntos y pensamos en las acciones que los usuarios querían realizar en el sistema, como agregar tareas, marcarlas como completadas, organizarlas, etc.
- **Desglosamos en Historias de Usuario:** A partir de esa lista, dividimos cada funcionalidad en historias de usuario individuales. Cada historia representaba una acción específica que un usuario quería hacer.
- **Descripciones y Detalles:** Escribimos descripciones breves para cada historia y agregamos detalles como prioridad y estimación de tiempo. Esto ayudó a comprender mejor cada historia.
- **Asignamos Responsabilidades:** Decidimos quién sería responsable de cada historia. Algunas historias las compartimos entre nosotros según nuestras habilidades y conocimientos.
- **Revisión Conjunta:** Pasamos tiempo revisando todas las historias juntos. Esto aseguró que cada historia reflejara las necesidades reales del sistema.

Por Qué Fue Importante:

- **Entender las Necesidades del Usuario:** Crear historias de usuario nos ayudó a comprender lo que los usuarios quieren del sistema. Esto aseguró que el sistema sea realmente útil.
- **Evitar Malentendidos:** Al tener descripciones claras, evitamos confusiones sobre qué se debe construir y cómo debe funcionar.
- **Priorización Informada:** Las historias nos permitieron priorizar lo que los usuarios realmente necesitan. No perdimos tiempo en características menos importantes.
- **Medir el Progreso:** Cada historia completada marcó un progreso tangible. Fue gratificante ver cuánto avanzamos.
- **Adaptarse Rápidamente:** Si las necesidades cambiaban, podíamos ajustar o agregar historias según lo que aprendíamos.
- **Trabajo en Equipo:** Colaborar en la creación de historias unió nuestras habilidades y nos aseguró que todos estuviéramos en la misma página.
- **Cumplimiento de Expectativas:** Las historias aseguraron que el sistema final cumpla con lo que queríamos lograr.

Identificador:	R001	Nombre:	Agregar tareas
Tipo:		¿Crítico?: SI/NO	Si
Necesario			
Prioridad de desarrollo:		Alta	
Entrada:		Salida:	
Agregar tareas		Se crea la tarea	
Descripción:			
Los usuarios pueden agregar nuevas tareas a la lista.			
Manejo de situaciones Anormales:			
El usuario omite presionar enter y no se crea con éxito la tarea			
Criterios de Aceptación			
Se crea con éxito la tarea			

Identificador:	R002	Nombre:	Edición de tareas
Tipo:		¿Crítico?: SI/NO	NO
Deseable			
Prioridad de desarrollo:	Medio		
Entrada:		Salida:	
Edición de tareas		Se muestran los cambios sobre una tarea anteriormente definida	
Descripción:			
<p>El contenido de las tareas existentes puede ser editado por los usuarios para hacer algún cambio.</p>			
Manejo de situaciones Anormales:			
<p>Si el usuario no realiza una edición o si la realiza y no la guarda quedará en su estado original.</p>			
Criterios de Aceptación			
<p>La tarea es editada con éxito</p>			

Identificador:	R004	Nombre:	Marcar tareas como completas.
Tipo:		¿Crítico?: SI/NO	SI
Necesario			
Prioridad de desarrollo:	Alta		
Entrada:		Salida:	
Marcar tareas como completas.		Confirmación de tarea realizada con un checkbox.	
Descripción:			
El usuario puede indicar que la tarea está completa a través de una selección en el checkbox.			
Manejo de situaciones Anormales:			
Usuario omite el checkbox.			
Criterios de Aceptación			
Tarea se muestra con el indicador checkbox como completada.			

Lista de tareas priorizadas

Definición y Planificación:

- **Establecer objetivos y alcance del proyecto:** Establecer claramente lo que deseas lograr con el sistema y qué funcionalidades incluirá
- **Identificar roles y responsabilidades del equipo:** Definir quiénes serán los miembros del equipo y qué roles asumirán en el proyecto.
- **Crear repositorio en GitHub.**
- **Configurar plataforma de gestión de tareas:** En este proyecto usaremos la plataforma de Trello

Diseño y Documentación:

- **Crear diagramas de flujo para funcionalidades clave:** Visualizar cómo las acciones de los usuarios se conectarán y fluirán dentro del sistema.
- **Diseñar la interfaz de usuario en función del frontend proporcionado:** Utilizar el diseño proporcionado para construir la apariencia visual del sistema.
- **Crear diagramas de casos de uso:** Representar escenarios donde los usuarios interactúan con el sistema y cómo se comportará en respuesta.
- **Escribir historias de usuario:** Describir las tareas específicas que los usuarios quieren realizar y cómo el sistema debe responder.
- **Documentar requisitos funcionales y no funcionales:** Enumerar lo que el sistema debe hacer (funcional) y cómo debe hacerlo (no funcional).

Implementación:

- **Configurar almacenamiento en local (localStorage / sessionStorage):** Preparar la estructura para almacenar la información de las tareas en el navegador.
- **Desarrollar función para agregar nueva tarea:** Desarrollar el código que permita a los usuarios crear y guardar nuevas tareas en el sistema.
- **Implementar función para eliminar tarea:** Crear la capacidad para que los usuarios eliminen tareas que ya no necesitan.
- **Implementar función para marcar tarea como completada:** Permitir a los usuarios hacer un seguimiento de las tareas completadas.

- **Desarrollar función para editar información de una tarea:** Permitir que los usuarios modifiquen detalles de las tareas existentes.
- **Implementar función para visualizar todas las tareas:** Mostrar una lista completa de todas las tareas en el sistema.

Integración y Control de Versiones:

- **Inicializar repositorio Git en proyecto:** Configurar el control de versiones para el proyecto en GitHub.
- **Realizar commits y utilizar ramas para desarrollo:** Hacer registros de los cambios en el código en el repositorio a medida que avanzas.
- **Fusionar ramas de desarrollo con la rama principal:** Combinar las ramas de desarrollo con la rama principal (master) cuando las funcionalidades estén completas.
- **Utilizar ramas para desarrollar funciones de manera aislada:** Crear ramas separadas para trabajar en diferentes partes del proyecto sin afectar la rama principal.

Pruebas:

- **Realizar pruebas unitarias de funciones:** Probar individualmente cada función que se ha implementado para asegurar de que funcionan correctamente.
- **Probar integración y flujo del sistema:** Verificar que las funciones trabajen bien juntas en el sistema.
- **Identificar y corregir errores:** Encontrar y solucionar cualquier error o comportamiento inesperado en el sistema.
- **Realizar pruebas de extremo a extremo:** Probar todo el flujo del sistema para asegurarte de que las tareas se completen según lo planeado.

Documentación y Entrega:

- **Actualizar archivo README.md con instrucciones:** Proporcionar una guía para que otros puedan ejecutar y comprender el proyecto.
- **Detallar ciclo de vida del software aplicado:** Explicar cómo se aplicó las etapas del ciclo de vida del software en tu proyecto.
- **Preparar el proyecto para entrega:** Revisar que todo esté completo y funcione bien en el trabajo