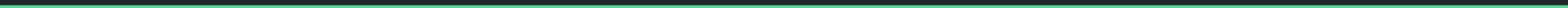


# GRUPO 1

# CIERRE SPRINT 2



# User Stories Planificadas

---

# Historias de usuario

US 3.1 - Crear publicación

8 SP	Cumplido
------	----------

US 3.2 - Eliminar publicación

8 SP	Cumplido
------	----------

US 2.3 - Configurar preferencias

5 SP	Cumplido
------	----------

US 2.4 - Buscador de destinos y actividades

5 SP	Cumplido
------	----------

US 4.2 - Guardar en favoritos

3 SP	Cumplido
------	----------

US 4.1 - Ver puntaje promedio

5 SP	Cumplido
------	----------

US 5.5 - Publicar reseña (Premium)

5 SP	Cumplido
------	----------

US 4.4 - Filtrar por categoría

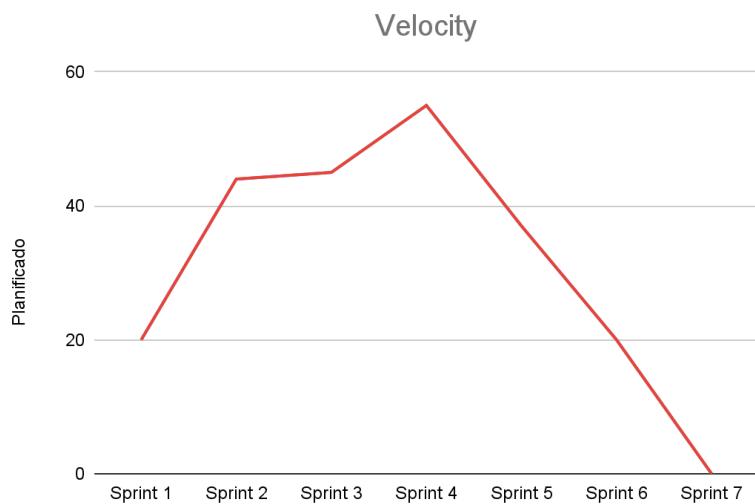
5 SP	Cumplido
------	----------

**Velocity**

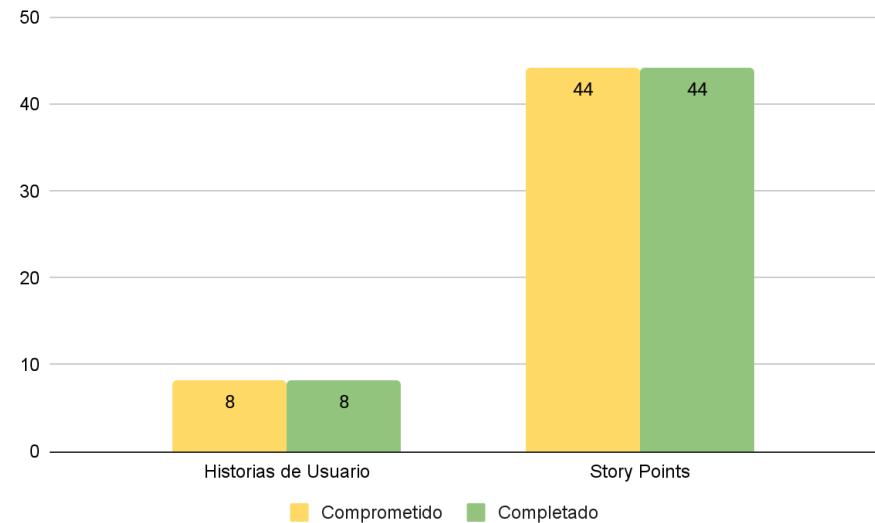
**44 SP**

**Total:**

# Métricas



Velocity actual: 44 SP



# Métricas



# Uso de IA

---

# Uso de IA

- Herramienta utilizada: Gemini y ChatGPT
- Versiones e iteraciones: Se le envió contexto con código ya hecho con el fin de generar código nuevo. Se utilizó también para resolución de errores.

# Uso de IA - Ejemplos

Estos son los archivos de main. Hay cambios en esos archivos puntualmente, analiza que esta en el archivo viejo y no esta en este y decime como integrar los cambios de la branch en estos archivos de main. Las dunctionalidades basicamente es que aparecen cosas con favoritos, y cosas sobre busquedas.

Espera que te mando algunos de la branch que me olvide de mandar

Perfecto, Imanol.  
Ya tengo los archivos de `main` y los de la branch. Te explico de forma precisa qué funcionalidades de la branch no están en el main y cómo integrarlas correctamente, tanto en backend como en frontend.

**RESUMEN GENERAL DE DIFERENCIAS**

Área	Branch	Main	Integración recomendada
Favoritos	Tiene tablas <code>favorites</code> , endpoints <code>/favorite</code> Y <code>/favorites</code> , y campo <code>is_favorite</code> EN <code>PublicationOut</code> .	No existe.	Crear modelo <code>Favorite</code> , endpoint <code>/favorites</code> y agregar campo <code>is_favorite</code> en el modelo <code>Publication</code> .
Solicitudes de eliminación	Tiene tabla <code>deletion_requests</code> , endpoints <code>/deletion-requests/...</code> , y campo <code>...</code>	No existe.	Copiar modelo, schema y endpoint <code>db_migrations.py</code> para crear la tabla <code>deletion_requests</code> .

Plan-Go-desarrollo\_preferences.zip  
Archivo zip

Este es el proyecto. Nuestra idea es poder desarrollar una funcionalidad que pueda incorporar favoritos dentro de lo que son las publicaciones. También queremos adicionar la visual desde el frontend para que lo pueda ver el usuario.  
Nuestra idea es relacionarlo con la base de datos agregando un campo dentro de los valores de una publicación que solo se habilite verificando que esta sea habilitada para un usuario premium,

Me dirías una propuesta para poder llevarlo a cabo

Pensado >

Perfecto, Imanol. No encontré referencias directas a una funcionalidad de `favoritos` dentro del proyecto que subiste, pero puedo proponerte una hoja de ruta concreta para implementarla integrando backend (FastAPI), base de datos (SQLAlchemy) y frontend.

**1. Objetivo general**

Agregar una funcionalidad que permita a los usuarios premium marcar publicaciones como "favoritas", visualizarlas en su perfil, y que esta información se almacene en la base de datos. Los usuarios *no premium* no deberían tener acceso a esta opción.

# Uso de IA - Justificación de Resultado Final

- Se utilizó la IA mediante el pasaje del contexto en el cual estaba el tp. Utilizando la IA se pudo desarrollar de manera más precisa sabiendo en qué partes del código se deberían realizar modificaciones.
- Además, la inteligencia artificial aportó buenas prácticas que favorecen la coherencia y consistencia de la base de datos, como el uso de relaciones adecuadas entre tablas. También contribuyó al diseño de APIs RESTful específicas y seguras, con endpoints claramente separados según la acción correspondiente. En el frontend, promovió un manejo de estado limpio en React, con una separación clara entre las vistas y la lógica de negocio.

# Retro

---

## **Sprint 2 - Evolución de Riesgos**

**1.** Si bien se alcanzó a realizar una investigación completa sobre la herramienta de IA ‘Gemini’ para su integración.

El tiempo que nos llevó generó retrasos en el desarrollo, por lo que se decidió ajustar el cronograma y posponer al siguiente sprint el feature relacionado.

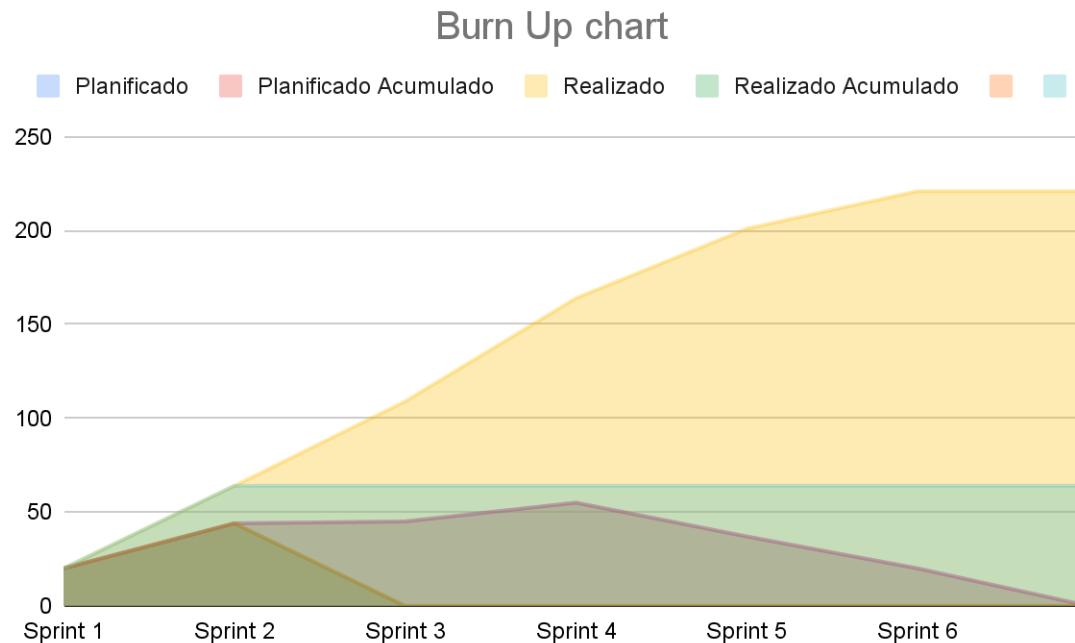
Esta reprogramación buscó reducir riesgos asociados a la falta de experiencia y asegurar una integración más sólida y planificada.

**2.** Hubo un problema en la comunicación y quedaron cambios sin mergear en una rama; generando un conflicto en el desarrollo.

## Sprint 2

- Luego de modificar el cronograma, se lograron cumplir todas las tareas planificadas.
- Incorporamos un nuevo artefacto de “gestión de cambios”, para un mejor seguimiento de posibles actualizaciones o cambios repentinos durante el desarrollo.
- Se tuvo una reunión grupal donde se buscó formas de mejorar la metodología de trabajo, para asegurar que no se vuelva a caer en inconsistencias durante el desarrollo.

# Avance



# **DEMO**

# Compromisos para Sprint 3

---

# Historias de usuario

US 4.4 - Gestión de suscripción Premium

8 SP

US 5.4 - Ordenar sugerencias por puntaje

5 SP

US 4.3 - Armar Itinerario (IA)

13 SP

US 5.7 - Ver reseñas en la ficha

8 SP

US 4.7 - Marcar publicaciones favoritas como  
pendientes o realizados

8 SP