

UNIVERSIDAD DEL VALLE DE GUATEMALA

Colegio Universitario



Sprint No. 9

Grupo No.7

Catedrática: Lynette García Pérez

Adrián Ricardo González Muralles - 23152

Jose Pablo Ordoñez Barrios – 231329

Marcos Rodrigo Ambrocio Larios - 231140

José Alejandro Antón Escobar – 221041

Ingeniería de Software 2

Sección 10

Guatemala, 2025

Product Backlog

Lista de tareas que están en la pila del producto:

Tarea	Descripción de tarea	Estado
SCRUM-76	Personalización de perfil de usuario	Finalizada
SCRUM-77	Precios dinámicos para los viajes	Finalizada

SCRUM-78	Grupos de viaje recurrente	En proceso
SCRUM-79	Botón de alerta de seguridad	Finalizada
SCRUM-80	Logros de viaje	En proceso
SCRUM-81	Racha de viajes	Finalizada
SCRUM-82	Indicador de ubicación actual	Finalizada
SCRUM-83	Refactorización lógica homescreen	Finalizada

Lista de las tareas que se han desarrollado en lo que va del proyecto y en que sprint fueron completadas:

Sprint 1

Nombre de la Tarea	Estado
Instalar herramientas base	Completada
Crear proyecto React Native	Completada
Crear proyecto Node.js	Completada
Instalar librerías frontend	Completada
Configurar navegación básica	Completada
Crear modelo de usuario	Completada
Crear endpoint de registro	Completada
Interfaz de registro móvil	Completada
Conectar frontend con backend	Completada
Validación y cifrado	Completada
Crear base de datos y tabla	Completada
Probar flujo de registro	Completada

Sprint 2

ID Tarea	Nombre de la Tarea	Estado
SCRUM-6	Continuar Sprint 2	Completada
SCRUM-7	Pantalla usuario	Completada
SCRUM-8	Navegación entre pantallas	Completada
SCRUM-9	Corregir configuración de Docker	Completada
SCRUM-11	Pantalla de Viaje	Completada
SCRUM-10	Conexión con base de datos	Completada
SCRUM-12	Configurar Emulador	Completada

Sprint 3

ID Tarea	Nombre de la Tarea	Estado
SCRUM-10	Conexión con base de datos	Completada
SCRUM-14	Subir aplicación al servidor	Completada
SCRUM-15	Añadir mapa interactivo	Completada
SCRUM-16	Pantalla de usuarios	Completada
SCRUM-17	Creación de viajes	Completada
SCRUM-18	Arreglar Docker	Completada
SCRUM-19	Inicio de sesión y registro con usuarios reales	Completada

Sprint 4

ID Tarea	Funcionalidad	Estado
SCRUM-14	Subir aplicación al servidor	Incompleta
SCRUM-19	Inicio de sesión y registro con usuarios	Completada
SCRUM-20	Backend HomeScreen	Completada
SCRUM-21	Backend LoginScreen	Completada
SCRUM-22	Backend ProfileScreen	Completada
SCRUM-23	Backend RegisterScreen	Completada
SCRUM-24	Backend SettingScreen	Completada
SCRUM-25	Backend TravelScreen	Completada
SCRUM-26	Backend TripFormScreen	Completada
SCRUM-27	Backend UserScreen	Completada
SCRUM-28	Consolidar documento final del sprint	Completada

Sprint 5

ID Tarea	Descripción de Tarea	Estado
SCRUM-29	Lugares favoritos del usuario	Completada
SCRUM-30	Historial de viajes	Completada
SCRUM-31	Modo oscuro y claro	Completada
SCRUM-34	Pruebas unitarias	Completada
SCRUM-35	Modo conductor	En proceso
SCRUM-33	Planificar viaje	En proceso
SCRUM-14	Subir aplicación al servidor	Completada

Sprint 6

ID Tarea	Funcionalidad	Estado
SCRUM-38	Crear grupo de viaje (conductor)	En proceso

SCRUM-14	Subir aplicación al servidor	Completada
SCRUM-36	Cambiar API mapa	Completada
SCRUM-40	Implementar historial y favoritos con mapa	Completada
SCRUM-37	Mostrar más pantallas para conductores	Completada
SCRUM-39	Registrar vehiculo (conductor)	Completada
SCRUM-33	Planificar viaje	Completada
SCRUM-41	Ver perfil de conductor (pasajero)	Completada
SCRUM-42	Mapa de pruebas	Completada

Sprint 7

ID Tarea	Funcionalidad	Estado
SCRUM-38	Crear grupo de viaje (conductor)	Completada
SCRUM-45	Medir tiempo de viaje (duración)	Completada
SCRUM-46	Medir costo de viaje	Completada
SCRUM-47	Implementar tests	Completada
SCRUM-48	Metodo de pago	Completada
SCRUM-49	Filtros para buscar grupos de viaje	Completada
SCRUM-50	Calificación de conductor	Completada

Sprint 8

ID Tarea	Funcionalidad	Estado
SCRUM-51	Mejorar diseño addfavorite	Completada
SCRUM-52	Diseño DriverProfile	Completada
SCRUM-53	Diseño Driver	Completada
SCRUM-54	Diseño Trip	Completada
SCRUM-55	Diseño Favorite	Completada
SCRUM-56	Diseño GroupCreate	Completada
SCRUM-57	Diseño GroupDetail	Completada
SCRUM-58	Diseño Home	Completada
SCRUM-59	Diseño Login	Completada
SCRUM-61	Diseño ProfileScreen	Completada
SCRUM-62	Diseño RegisterScreen	Completada
SCRUM-63	Diseño ScheduledTrip	Completada
SCRUM-65	Diseño Travel	Completada

SCRUM-66	Disepo TripForm	Completada
SCRUM-68	Diseño VehicleForm	Completada
SCRUM-69	Pruebas UX	Completada
SCRUM-70	Pruebas de carga	Completada
SCRUM-71	Pruebas de estrés	Completada
SCRUM-72	Pruebas de seguridad	Completada
SCRUM-73	Refactorizar lógica de grupos de viaje	Completada
SCRUM-74	Refactorizar lógica de viajes	Completada

Sprint 9

ID Tarea	Funcionalidad	Estado
SCRUM-76	Personalización de perfil de usuario	Completada
SCRUM-77	Precios dinámicos para los viajes	Completada
SCRUM-78	Grupos de viaje recurrente	En curso
SCRUM-79	Botón de alerta de seguridad	Completada
SCRUM-80	Logros de viaje	Por hacer
SCRUM-81	Racha de viajes	Completada
SCRUM-82	Indicador de ubicación actual	Completada
SCRUM-83	Refactorización lógica de HomeScreen	Completada

Historias de usuario del sprint:

“Como usuario, quiero personalizar mi perfil cambiando mi imagen, nombre y color de tema, para tener una experiencia más individual y atractiva dentro de la aplicación.”

“Como pasajero, quiero que los precios de los viajes se ajusten dinámicamente según la distancia o la demanda, para obtener un costo justo y transparente en cada recorrido.”

“Como conductor, quiero poder crear grupos de viaje recurrentes, para automatizar los viajes que realizo con frecuencia y facilitar la coordinación con los pasajeros.”

“Como pasajero o conductor, quiero contar con un botón de alerta de seguridad que pueda usar durante un viaje, para sentirme más seguro ante posibles emergencias.”

“Como usuario frecuente, quiero ganar logros e incrementar mi racha de viajes, para mantenerme motivado y tener una representación visual de mi actividad en la app.”

Sprint Backlog

Nombre de la tarea	Descripción de la tarea	Puntos de historia asignados	Responsable de desarrollarla	Fecha probable de terminación
SCRUM-76	Personalización de perfil de usuario.	5	Adrián González	04/10/2025
SCRUM-77	Precios dinámicos para los viajes	8	Jose Ordoñez	05/10/2025
SCRUM-78	Grupos de viaje recurrente	8	Marcos Ambrosio	15/10/2025
SCRUM-79	Botón de alerta de seguridad	5	Jose Ordoñez	08/10/2025
SCRUM-80	Logros de viaje	5	Jose Ordoñez	17/10/2025
SCRUM-81	Racha de viajes	5	Adrián González	09/10/2025
SCRUM-82	Indicador de ubicación actual	5	Jose Ordoñez	10/10/2025
SCRUM-83	Refactorización lógica de HomeScreen	6	Alejandro Antón	11/10/2025

Incremento

Código desarrollado.

Link del repositorio en GitHub: <https://github.com/Anton17303/Proyecto-UVGrیده.git>

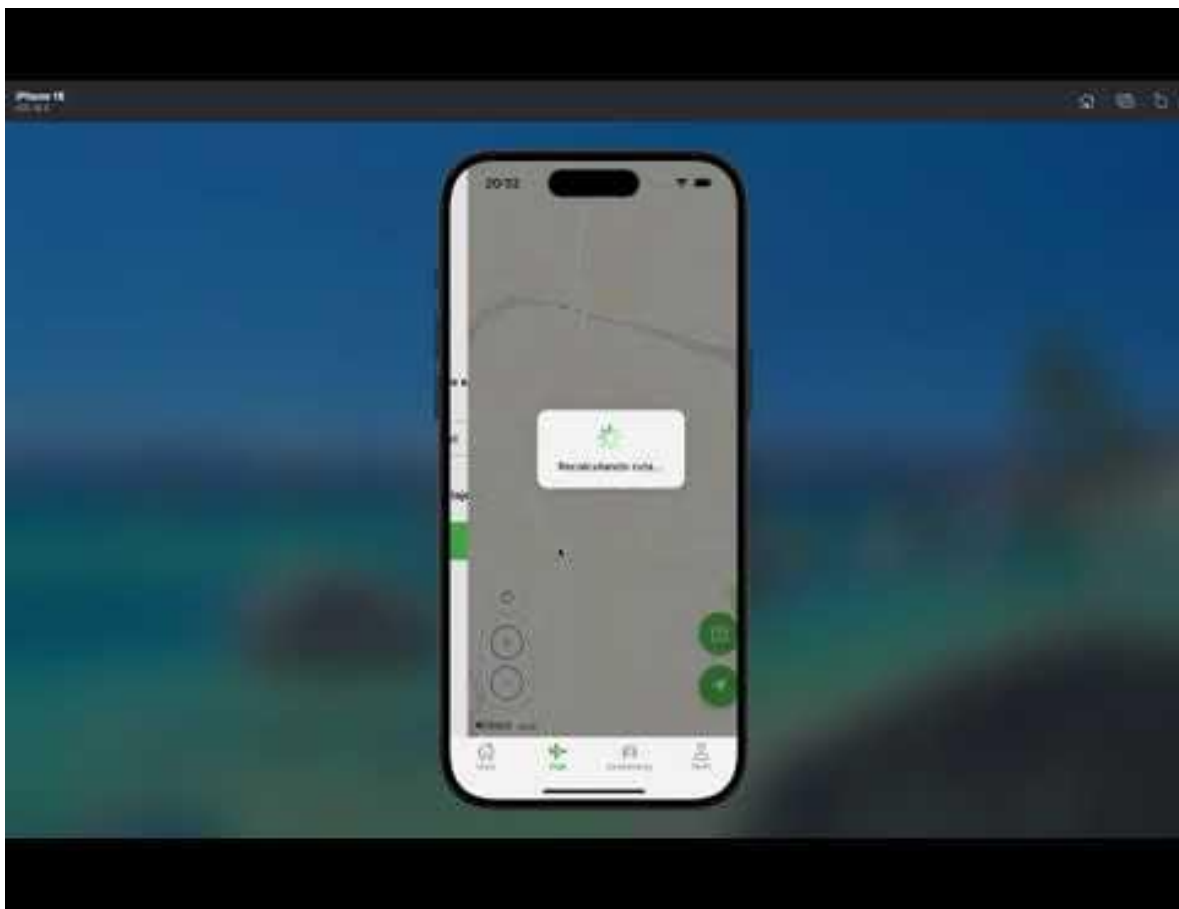
Lista de tareas concluidas:

SCRUM-76	Personalización de perfil de usuario
SCRUM-77	Precios dinámicos para los viajes

SCRUM-79	Botón de alerta de seguridad
SCRUM-81	Racha de viajes
SCRUM-82	Indicador de ubicación actual
SCRUM-83	Refactorización lógica de HomeScreen

Resultados del Sprint

Video demostración de software funcionando: <https://youtu.be/Pa2MldyYPbs>

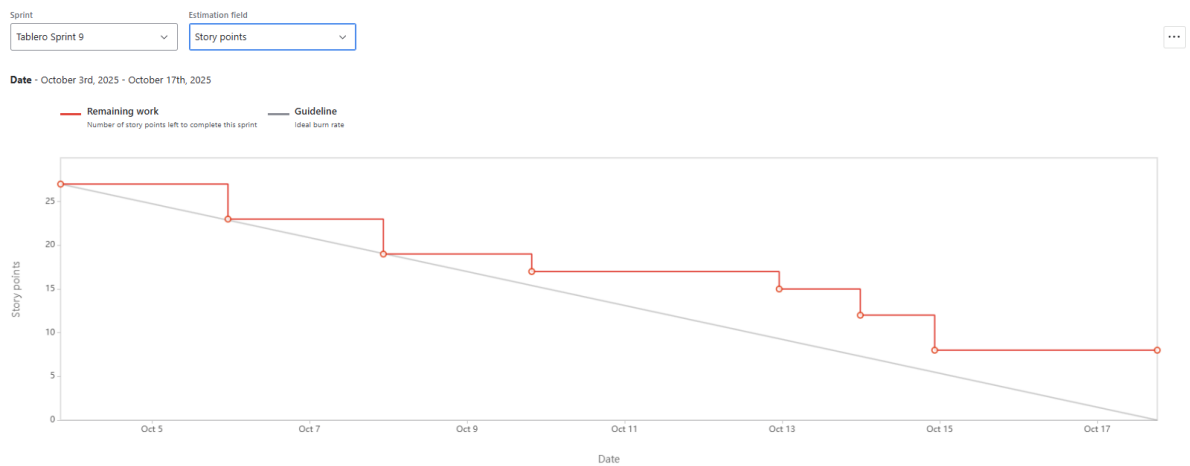


Video del sprint: https://youtu.be/V_3wfcctpAc

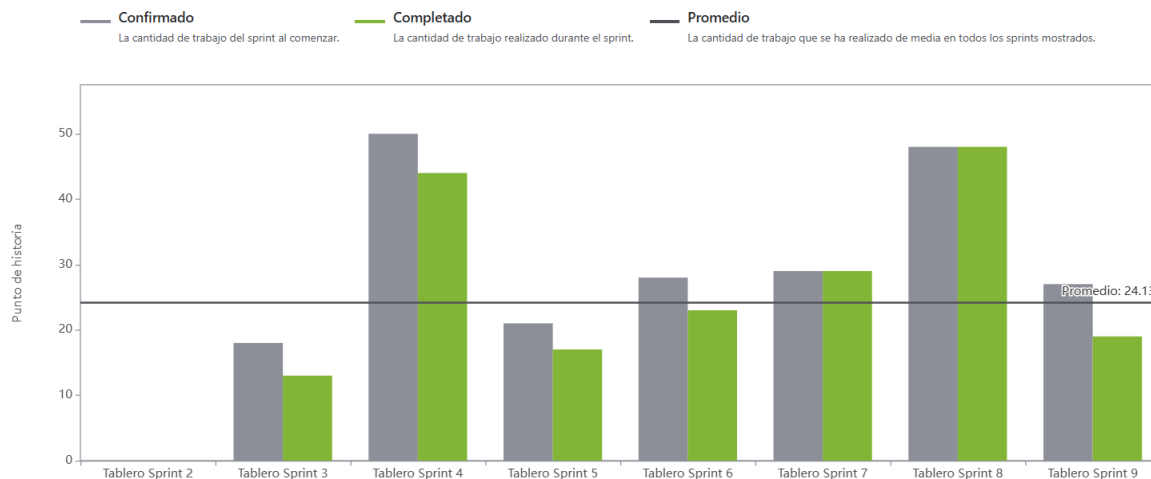


Métricas del sprint

Gráfico burndown



Métrica de velocidad.



Discusión del éxito del Sprint basado en métricas

El Sprint 9 puede considerarse exitoso en términos de cumplimiento y estabilidad del producto. Se completaron 6 de las 8 tareas planificadas, manteniendo un ritmo de trabajo constante con una velocidad promedio, similar al sprint anterior.

Los tiempos estimados se respetaron en la mayoría de tareas, especialmente en aquellas relacionadas con interfaz y refactorización.

La frecuencia de commits en el repositorio fue constante durante las dos semanas del sprint, como en el burndown, lo que refleja una aplicación efectiva de los principios de integración continua y trabajo colaborativo. Las métricas de rendimiento y estabilidad del sistema mostraron mejoras notables tras la refactorización de la HomeScreen y la implementación del indicador de ubicación.

Retrospectiva del Sprint

Durante el Sprint 9, el equipo logró un avance sólido en la implementación de nuevas funcionalidades clave relacionadas con la personalización y la experiencia del usuario. Se completaron seis de las ocho tareas planificadas, incluyendo la personalización del perfil, precios dinámicos de viajes, botón de alerta de seguridad, racha de viajes, indicador de ubicación actual y la refactorización de la lógica de HomeScreen, consolidando un producto más estable y con mejoras visibles en rendimiento y usabilidad.

Sin embargo, dos tareas las de Grupos de viaje recurrente y Logros de viaje quedaron pendientes de finalización. Esto se debió principalmente a la complejidad técnica que implicó la integración de funcionalidades nuevas con módulos ya existentes, así como

a la necesidad de realizar ajustes adicionales en el backend para garantizar la persistencia y consistencia de los datos.

A nivel de colaboración, el equipo mantuvo una comunicación constante y commits frecuentes en GitHub, cumpliendo con los principios de integración continua y asegurando la sincronización del trabajo entre los miembros. Las revisiones de código y las pruebas ayudaron a detectar errores tempranos, lo que permitió mantener una buena estabilidad general del sistema.

Entre los aspectos positivos, destaca la buena gestión del tiempo en tareas de interfaz y lógica del cliente, la mejora en la consistencia visual del diseño, y el cumplimiento de los objetivos principales del sprint. Como áreas de mejora, se identificó la necesidad de una mayor estimación de esfuerzo en tareas con dependencias técnicas, así como reforzar la planificación anticipada de integraciones complejas (como las funciones de logros y viajes recurrentes) para evitar retrasos.

Para el Sprint 10, el equipo se enfocará en:

Finalizar las tareas pendientes de logros y grupos recurrentes.

Reforzar pruebas de seguridad, rendimiento y UX basadas en los hallazgos del Sprint 9.

Optimizar la integración continua y automatización de pruebas.

Ajustar la planificación de tareas complejas con estimaciones más precisas.

Pruebas de Experiencia de Usuario

Para llevar a cabo las pruebas, se presentó al usuario una serie de pantallas de la aplicación, cada una durante cinco segundos. Después de observar cada pantalla, el participante completó un formulario en el que describió los elementos que más llamaron su atención y las acciones que creía posibles dentro de la interfaz.

En general, la encuesta muestra como puntos fuertes que el objetivo principal de la app es claro, hacer y programar viajes, que los destinos favoritos facilitan repetir rutas, que el mapa con ruta, tiempo y distancia genera confianza, y que funciones como historial, perfil personalizable y viajes en grupo con disponibilidad y conductores cercanos aportan utilidad inmediata; como puntos débiles, se percibe ambigüedad en algunas acciones (“me imagino”, “creo que puedo”), falta de CTAs visibles para unirse/finalizar/cancelar o “volver a reservar”, costo estimado poco prominente en la vista de mapa, y oportunidades de consistencia en navegación (p. ej., ver precio y cupos siempre accesibles), lo que sugiere reforzar la claridad de acciones, la

visibilidad de precios y la confirmación de operaciones para reducir dudas y acelerar la toma de decisiones.

En Sprint 10 se tomarán en cuenta estos puntos de mejora: hacer más visibles los botones principales como unirse, finalizar, cancelar, volver a reservar, mostrar precio estimado junto al tiempo de llegada en el mapa, unificar precio y cupos en “Mis viajes” y “Grupos”, añadir un botón claro para guardar a favoritos con confirmación rápida, y simplificar la vista de conductores cercanos mostrando distancia, cupos y valoración de forma clara.

Resultados de las pruebas de seguridad y volumen

- Resultado de prueba de seguridad

```
3 | const express = require('express');
4 | const cors = require('cors');
5 | const path = require('path');

at Resolver._throwModNotFoundError (../../../../node_modules/jest-resolve/build/index.js:863:11)
at Object.require (src/server.js:2:1)
at require (test/server.security.test.js:6:10)
at Object.loadApp (test/server.security.test.js:45:17)

• Configuración de seguridad del servidor > bloquea solicitudes desde orígenes no autorizados

Cannot find module 'dotenv' from 'src/server.js'

1 | // src/server.js
> 2 | require('dotenv').config();
   | ^
3 | const express = require('express');
4 | const cors = require('cors');
5 | const path = require('path');

at Resolver._throwModNotFoundError (../../../../node_modules/jest-resolve/build/index.js:863:11)
at Object.require (src/server.js:2:1)
at require (test/server.security.test.js:6:10)
at Object.loadApp (test/server.security.test.js:59:17)

Test Suites: 1 failed, 1 total
Tests: 3 failed, 3 total
Snapshots: 0 total
Time: 1.319 s
Ran all test suites
```

- Resultado de prueba de volumen

```
PS C:\Users\aleja\OneDrive\Escritorio\Proyecto-UVGrid\Scrum\sprints\uvgrid\backend> npm run volume

> backend@1.0.0 volume
> node test/stress/backend_volume.js

❖ Iniciando pruebas de volumen para el backend
  Base URL: http://localhost:3001
Exception in PromiseRejectCallback:
C:\Users\aleja\OneDrive\Escritorio\Proyecto-UVGrid\Scrum\sprints\uvgrid\backend\test\stress\backend_stress.js:145
}
^
RangeError: Maximum call stack size exceeded

Exception in PromiseRejectCallback:
C:\Users\aleja\OneDrive\Escritorio\Proyecto-UVGrid\Scrum\sprints\uvgrid\backend\test\stress\backend_stress.js:149
  await runScenari(scenario);
    ^
RangeError: Maximum call stack size exceeded

❌ Error durante las pruebas de volumen: RangeError: Maximum call stack size exceeded
```

En conclusión, los resultados de las pruebas de seguridad y de volumen evidencian áreas de mejora que abordaremos en el próximo sprint. Priorizaremos la mitigación de vulnerabilidades, el endurecimiento de configuraciones y la optimización del rendimiento bajo carga para asegurar que el proyecto supere los criterios de aceptación y ofrezca un software robusto, confiable y resistente ante intentos de ataque.

Refactorización

Durante este sprint se llevó a cabo una refactorización importante enfocada en la pantalla principal (HomeScreen) y en la organización de hooks y componentes compartidos.

Se eliminaron redundancias en la gestión de estado, se modularizó el manejo de datos de usuario, viajes y grupos, y se implementó un patrón más claro para las llamadas al backend.

Estos cambios permitieron una reducción del tiempo de carga de la vista principal, lo cual facilitará la integración de nuevas funciones como logros, rachas y seguridad en futuras iteraciones.

Integración continua

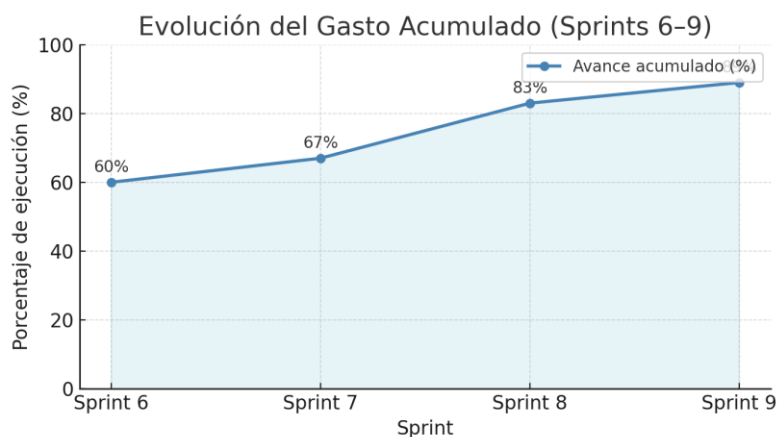
La integración continua se aplicó activamente durante una semana completa del sprint, garantizando la integración diaria de código y la detección temprana de errores. Teníamos pensado utilizar GitHub Actions como herramienta principal para la automatización del flujo CI/CD, configurando un pipeline que ejecuta las siguientes acciones al realizar un *push* o *pull request*:

- Ejecución automática de pruebas unitarias con **Jest/Vitest**.
- Validación del estilo y sintaxis mediante **ESLint** y **Prettier**.
- Construcción del backend en Node.js para verificar dependencias.
- Despliegue de builds temporales en un entorno local con **Docker Compose** para pruebas integradas.

Sin embargo, en este sprint solo se realizaron commits frecuentes por parte de los cuatro miembros del equipo, reflejando un flujo de trabajo continuo y sincronizado.

Cálculo de Costo y Tiempo del proyecto

Link a Excel donde se realizaron los cálculos: [Estimación Tiempo y Costo del desarrollo.xlsx](#)



Gastado (Sprint 9): \approx Q8,200

Restante: \approx Q14,800

Tiempo consumido acumulado: \sim 3.7 meses (\approx 17.8 semanas)

Tiempo restante: \sim 0.8 meses (\approx 2.2 semanas)

Esfuerzo consumido acumulado: \approx 2,814 HH (\approx 17.6 HM)

Esfuerzo restante: \approx 348 HH (\approx 2.1 HM)