



UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL CÓRDOBA
Ingeniería en Sistemas de Información

Cátedra Ingeniería y Calidad de Software

Trabajo Práctico N° 8

SCRUM - Planificación de Release y de Sprint

Integrantes:

- Castro Wachs, Trinidad - 89828
- Colque, Sebastián - 90316
- Feretti, Lucía - 89553
- Fuxa, Joaquín - 89303
- Ibarra, Mauro - 90025

Docentes:

- Cecilia Massano
- Constanza Garneró
- Judith Meles

Índice

Plan de Release.....	2
Objetivo del Release.....	2
Consideraciones.....	2
Tareas de User Stories.....	3
Sprint planning.....	5
Minuta Sprint 1.....	5
Sprint Backlog.....	6
Definición de Hecho.....	7
Bibliografía.....	8

Plan de Release

Objetivo del Release

El objetivo del release es entregar una versión funcional del MVP que permita a los pasajeros solicitar taxis cercanos, y a los taxistas visualizar la ubicación del pasajero y gestionar el estado de sus vehículos.

Consideraciones

- Al ser el primer Release, el equipo no cuenta con velocidad que pueda usar de referencia para asignar la cantidad de story points que se podrían llegar a realizar por sprint. El equipo estima que un inicio cada integrante es capaz de comprometerse con 2-3 story point por sprint, dando un total de 10-15 story point.
- Los sprint tendrán una duración de dos semanas cada uno (10 días hábiles).
- El inicio del release será la tercera semana de octubre del corriente año, en el que se iniciará el primer sprint.
- El equipo consta de 5 personas, y la capacidad del equipo se estima considerando las siguientes actividades:
 - Reuniones Scrum:
 - Daily meetings : 15 minutos
 - Sprint Planning: 4 horas
 - Sprint Review: 2 horas
 - Retrospective: 1,5 horas
 - Dedicación: Cada miembro del equipo tiene una capacidad de hasta 15 horas por semana aproximadamente (3 horas por día en una semana laboral de 5 días por las actividades adicionales mencionadas).
 - Contexto: Se tiene en cuenta que Mauro, Joaquin y Sebastián trabajan 4 horas por día durante toda la semana; los cumpleaños de Trinidad el 21 de octubre, de Sebastián el 1 de noviembre; el examen que todos tenemos el 26 y 29 de octubre y el examen que tiene Lucia el 18 de octubre.

En total son 22 story points, y teniendo en cuenta una capacidad de 10-15 story points, necesitaremos dos sprints para llevar a cabo el release. Con una duración de sprint de dos semanas, el release duraría un mes.

Sprint 1 (14 de octubre al 28 de octubre)	Sprint 2 (29 de octubre al 11 de noviembre)
Pedir taxi (5 SP)	Ver ubicación del pasajero (5 SP)
Buscar taxis cercanos (3 SP)	Loguear taxista (2 SP)
Ocupar taxista (2 SP)	Notificar al taxista solicitud de taxis (3 SP)
Liberar taxi (2 SP)	

Tareas de User Stories

Loguear Taxista (2 SP)

- Diseñar e implementar la interfaz de inicio de sesión del taxista - 4 horas
- Desarrollar el backend para autenticación - 6 horas
- Integrar la API de Facebook para la autenticación en el inicio de sesión - 4 horas
- Implementar pruebas unitarias - 4 horas

Total: **18 horas**

Ocupar Taxi (2 SP)

- Diseñar la interfaz para marcar la ocupación del taxi - 3 horas
- Implementar la lógica de ocupación en el backend - 6 horas
- Actualizar estado del taxi en la base de datos - 3 horas
- Pruebas unitarias - 4 horas

Total: **16 horas**

Liberar Taxi (2 SP)

- Diseñar la interfaz de liberación del taxi - 3 horas
- Implementar la lógica para liberar el taxi en el backend. - 6 horas
- Actualizar estado del taxi en la base de datos - 3 horas
- Pruebas Unitarias - 4 horas

Total: **16 horas**

Ver Ubicación del Pasajero (5 SP)

- Integrar API de geolocalización para obtener los datos - 10 horas
- Diseñar mapa interactivo en la interfaz del taxista - 8 horas
- Pruebas de geolocalización y rendimiento - 12 horas
- Pruebas unitarias - 8 horas

Total: **38 horas**

Buscar Taxis Cercanos (3 SP)

- Integrar API de geolocalización para obtener datos - 10 horas
- Implementar algoritmo de búsqueda de taxis cercanos - 6 horas
- Calcular tiempo estimado de los taxis al destino - 4 horas
- Diseñar la interfaz para mostrar taxis cercanos en la interfaz del pasajero - 4 horas
- Pruebas unitarias - 6 horas

Total: **30 horas**

Pedir Taxi (5 SP)

- Investigar e integrar API de geolocalización para obtener la ubicación actual de taxis y la del pasajero - 20 horas
- Diseñar la interfaz para la solicitud de taxi - 6 horas
- Implementar la funcionalidad para pedir un taxi y enviarlo al taxista - 6 horas
- Implementar mapa de visualización de taxis cercanos según mi localización - 4 horas
- Lógica para la selección de un taxi en el mapa - 4 horas
- Pruebas unitarias - 8 horas

Total: **48 horas**

Notificar al Taxista Solicitud de Taxi (3 SP)

- Investigar e implementar sistema de notificaciones push y toda su configuración - 14 horas
- Implementar envío de notificación a la central y al taxista - 4 horas
- Pruebas Unitarias - 4 horas

Total: **22 horas**

Duración Plan Release: Para realizar el MVP de manera completa se necesitará contar con una capacidad total de **188 horas** aproximadamente.

Sprint planning

Minuta Sprint 1

Sprint Número 1

Duración del Sprint en días hábiles: 10 días (desde 14 de octubre hasta 28 de octubre)

Objetivo del Sprint: Permitir a los usuarios “pasajeros” buscar taxis cercanos y solicitar uno, mientras que los usuarios “taxistas” podrán gestionar el estado de su vehículo marcándolo como “ocupado” cuando estén asignados a un pasajero, o como “liberado” cuando aún no tengan un pasajero confirmado.

Equipo Scrum:

- Trinidad - Scrum Master
- Lucia - Developer
- Mauro - Developer
- Sebastián - Developer Trainee
- Joaquín - QA Tester

Capacidad del Equipo en horas Ideales: 76-114 horas

Persona	Días Disponibles (sin tiempo personal)	Días para otras actividades Scrum	Horas por Dia	Horas Disponibles
Trinidad	9	2	2-3	14-21
Lucia	9	2	2-3	14-21
Mauro	10	2	2-3	16-24
Sebastián	10	2	2-3	16-24
Joaquín	10	2	2-3	16-24
TOTAL				76-114

Sprint Backlog

User Story	Actividades	Horas ideales
Pedir taxi (5 SP)	<ul style="list-style-type: none"> Investigar e integrar API de geolocalización para obtener la ubicación actual de taxis y la del pasajero - 20 horas Diseñar la interfaz para la solicitud de taxi - 6 horas Implementar la funcionalidad para pedir un taxi y enviarlo al taxista - 6 horas Implementar mapa de visualización de taxis cercanos según mi localización - 4 horas Lógica para la selección de un taxi en el mapa - 4 horas Pruebas unitarias - 8 horas 	48 hs
Buscar taxis cercanos (3 SP)	<ul style="list-style-type: none"> Integrar API de geolocalización para obtener datos - 10 horas Implementar algoritmo de búsqueda de taxis cercanos - 6 horas Calcular tiempo estimado de los taxis al destino - 4 horas Diseñar la interfaz para mostrar taxis cercanos en la interfaz del pasajero - 4 horas Pruebas unitarias - 6 horas 	30 hs
Ocupar taxista (2 SP)	<ul style="list-style-type: none"> Diseñar la interfaz para marcar la ocupación del taxi - 3 horas Implementar la lógica de ocupación en el backend - 6 horas Actualizar estado del taxi en la base de datos - 3 horas Pruebas unitarias - 4 horas 	16 hs
Liberar taxi (2 SP)	<ul style="list-style-type: none"> Diseñar la interfaz de liberación del taxi - 3 horas Implementar la lógica para liberar el taxi en el backend. - 6 horas Actualizar estado del taxi en la base de datos - 3 horas Pruebas Unitarias - 4 horas 	16 hs
12 Story Points		110 horas ideales

Definición de Hecho

- Código completo
 - El código fuente de la funcionalidad ha sido escrito.
 - El código tiene comentarios significativos.
 - El código se encuentra en el repositorio.
 - Se han aplicado buenas prácticas de escritura al código.
- Cumple con los criterios de aceptación en la user story.
- Se han ejecutado pruebas unitarias y han pasado exitosamente.
- Los bugs menores encontrados han sido registrados para ser abordados en el futuro (si son aceptables para la entrega).
- El código ha sido revisado y aprobado por al menos otro miembro del equipo.
- Unir la rama de la funcionalidad a la rama master.

Bibliografía

- Guía Scrum 2020 (Edición en Español para América Latina y el Sur de América)
- Presentaciones teóricas sobre SCRUM, planificación de release y sprint y métricas