



Ingeniería y calidad de software

# PLANIFICACIÓN DE RELEASE Y SPRINT

## Integrantes:

Beluzzo Gino, Federico	79233
Carrasco, Francisco	87930
De Dios, Guillermo	86635
Escudero, Octavio	89866
Galiano, Milagros	89513
Marchiori, Nicolás	85335
Pizarro, Emilio	87471
Vega, Candela	87345

## Docentes:

María Cecilia Massano  
Georgina Gisela González  
Laura Inés Covaro

Grupo 2

4K3 - 2024

# Índice

Condiciones de Contexto_____	2
Capacidad del equipo en horas ideales_____	2
Objetivo_____	3
Listado de user story_____	3
Duración de los sprints_____	4
Product backlog_____	4
Planificación por sprint_____	5
Minuta para Sprint Planning_____	5
Duración del sprint en días_____	5
Objetivo del sprint_____	6
Equipo scrum_____	6
Sprint backlog_____	7
Tareas por sprint_____	7
Definición de done_____	8

## Condiciones de Contexto

Contamos con un equipo de ocho personas, 5 con seniority **junior** en sus áreas de especialización: desarrollo web (front y back), diseño UX/UI, análisis de datos y control de calidad y 3 con seniority **trainee**.

## Capacidad del equipo en horas ideales

Este es el cálculo de capacidad de nuestro equipo para un sprint ideal, considerando las horas ideales que cada miembro puede disponer para el proyecto por día teniendo en cuenta los horarios de cursado de cada uno.

Cada miembro del equipo tiene una disponibilidad de entre dos y tres horas diarias durante los días lunes a viernes, más 2 horas el día sábado, como se detalla en la siguiente tabla.

Nos referimos a capacidad del equipo, como el tiempo que cada integrante puede dedicarle al sprint (Horas por día: días de lunes a viernes, Total por semana: agrega horas del día Sábado)

Integrante del equipo	Horas por día	Total por semana	Total horas ideales por sprint
Beluzzo Gino, Federico	2 horas	2 horas x 5 días + 2 horas x 1 día = 12 horas	12 x 2 semanas = 24 horas
Carrasco, Francisco	2 horas	2 horas x 5 días + 2 horas x 1 día = 12 horas	12 x 2 semanas = 24 horas
De Dios, Guillermo	3 horas	3 horas x 5 días + 2 horas x 1 día = 17 horas	17 x 2 semanas = 34 horas
Escudero, Octavio	2 horas	2 horas x 5 días + 2 horas x 1 día = 12 horas	12 x 2 semanas = 24 horas
Galiano, Milagros	2 horas	2 horas x 5 días + 2 horas x 1 día = 12 horas	12 x 2 semanas = 24 horas
Marchiori, Nicolás	2 horas	2 horas x 5 días + 2 horas x 1 día = 12 horas	12 x 2 semanas = 24 horas
Pizarro, Emilio	3 horas	3 horas x 5 días + 2 horas x 1 día = 17 horas	17 x 2 semanas = 34 horas
Vega, Candela Belén	2 horas	2 horas x 5 días + 2 horas x 1 día = 12 horas	12 x 2 semanas = 24 horas
Total			212 horas

Para las ceremonias, se tuvieron en cuenta los siguientes puntos:

- **Daily scrum** con una duración de 15 minutos por día.
- El día 1 del sprint se realizará la **sprint planning** de 2 horas de duración, calculadas como una hora por semana de trabajo.
- El día 14 del sprint se realizará la **sprint review** de 2 horas aproximadas y por último, la **sprint retrospective** que tendrá una duración de 45 minutos por cada semana que dure el sprint, dando un total de 1 hora y media.

En la siguiente tabla se detalla con más precisión la información previamente especificada:

Ceremonia	Horas por día	Horas totales en el sprint
Daily	0,25	0,25 x 13 días = 3,25 horas
Planning	2	2
Review	2	2
Retrospective	1,5	1,5
Total		Aproximadamente 9 horas

En conclusión, para la realización del release contamos con una capacidad del equipo igual a 212 horas, en las que incluimos las 9 horas destinadas a las ceremonias. Dando un total de **203 horas**, sin contemplar ceremonias.

## Planificación primer release

### Objetivo

Permitir la conexión entre pasajeros y taxistas mediante una plataforma que optimice el proceso de solicitud y gestión de taxis.

### Listado de user story

Taxista

- Loguear taxista
- Ocupar taxi
- Liberar taxi
- Ver ubicación del pasajero

#### Pasajero

- Buscar taxis cercanos
- Pedir taxi
- Notificar a taxista solicitud de taxi

## Duración de los sprints

El equipo definió que los sprints tendrán una duración total de 14 días, es decir sprints de 2 semanas. Para desarrollar las user stories incluidas en el primer release, realizaremos 5 sprints, más adelante detallaremos cada uno. Nos da un total de 10 semanas para finalizar la implementación del MVP.

## Product backlog

A continuación, se listan las user story a implementar priorizadas y con su respectiva estimación:

- Pedir taxi (5)
- Loguear taxista (2)
- Notificar a taxista solicitud de taxi (3)
- Buscar taxis cercanos (3)
- Ver ubicación del pasajero (5)
- Ocupar taxi (2)
- Liberar taxi (2)

Para alcanzar el objetivo del MVP, consideramos fundamental priorizar la funcionalidad pedir taxi, ya que nos permite conectar a los pasajeros con taxistas al contar con la posibilidad de solicitar taxis en cualquier lugar, lo que constituye la esencia del servicio que la aplicación ofrece.

Posteriormente se plantean implementar las users referidas al logueo del taxista y la notificación a taxista de solicitud de taxi ya que es necesario garantizar que los taxistas estén disponibles y puedan recibir las solicitudes de viaje generadas por los pasajeros. Además, se incluyen las users buscar taxis cercanos y la ubicación del pasajero. Todas estas users aseguran que la plataforma funcione como una vía de comunicación efectiva entre ambas partes.

Finalmente se agregan funcionalidades que cierran el ciclo de disponibilidad de los vehículos, ocupar y liberar taxi, las cuales no son de alta prioridad pero sí son fundamentales para lograr un completo funcionamiento del sistema, permitiendo lograr la conformidad en el uso del software.

## Planificación por sprint

Según la capacidad y la velocidad del equipo, se estima que el equipo de desarrollo podrá implementar alrededor de **5** puntos de historia por sprint.

Sprint	User story	Story points
1	<ul style="list-style-type: none"><li>Pedir taxi</li></ul>	5
2	<ul style="list-style-type: none"><li>Loguear taxista</li><li>Notificar a taxista solicitud de taxi</li></ul>	5
3	<ul style="list-style-type: none"><li>Buscar taxis cercanos</li><li>Ocupar taxi</li></ul>	5
4	<ul style="list-style-type: none"><li>Ver ubicación del pasajero</li></ul>	5
5	<ul style="list-style-type: none"><li>Liberar taxi</li></ul>	2

## Planificación del sprint

### Minuta para Sprint Planning

Sprint nro: 1

#### Duración del sprint en días

Se definieron sprint de 14 días. El primer sprint comenzará el 08/10, cómo tendrá una duración de 2 semanas, finalizará el 22/10. Los días sábados trabajaremos 2 horas por integrante del equipo de desarrollo, los días domingo y feriados no se destinarán horas al desarrollo del MVP.

En este periodo de tiempo, debemos contemplar un feriado del 11/10. El horario en el que se desarrollaran las actividades será de 9 am a 17 pm, detallando:

- De 10 a 10:15 hs se realizará la daily todos los días
- El día 08/10 se realizará la planning de 9 a 11 hs
- El día 22/10 se realizará la review de 11 a 13 hs, la retrospective de 15 a 16:30hs

Integrante del equipo	Horas por día	Días disponibles	Total por sprint
Beluzzo Gino, Federico	2 horas	9 Días hábiles 2 Sábados	$9 \text{ Días} * 2 \text{ Hs} + 2 \text{ Días} * 2 \text{ Hs} = 22 \text{ Hs}$
Carrasco, Francisco	2 horas	9 Días hábiles 2 Sábados	$9 \text{ Días} * 2 \text{ Hs} + 2 \text{ Días} * 2 \text{ Hs} = 22 \text{ Hs}$
De Dios, Guillermo	3 horas	9 Días hábiles 2 Sábados	$9 \text{ Días} * 3 \text{ Hs} + 2 \text{ Días} * 2 \text{ Hs} = 29 \text{ Hs}$
Escudero, Octavio	2 horas	9 Días hábiles 2 Sábados	$9 \text{ Días} * 2 \text{ Hs} + 2 \text{ Días} * 2 \text{ Hs} = 22 \text{ Hs}$
Galiano, Milagros	2 horas	9 Días hábiles 2 Sábados	$9 \text{ Días} * 2 \text{ Hs} + 2 \text{ Días} * 2 \text{ Hs} = 22 \text{ Hs}$
Marchiori, Nicolás	2 horas	9 Días hábiles 2 Sábados	$9 \text{ Días} * 2 \text{ Hs} + 2 \text{ Días} * 2 \text{ Hs} = 22 \text{ Hs}$
Pizarro, Emilio	3 horas	9 Días hábiles 2 Sábados	$9 \text{ Días} * 3 \text{ Hs} + 2 \text{ Días} * 2 \text{ Hs} = 29 \text{ Hs}$
Vega, Candela Belén	2 horas	9 Días hábiles 2 Sábados	$9 \text{ Días} * 2 \text{ Hs} + 2 \text{ Días} * 2 \text{ Hs} = 22 \text{ Hs}$
<b>Total</b>			<b>190 horas</b>

Para la realización del sprint contamos con una capacidad del equipo igual a 190 horas, en las que incluimos las 9 horas destinadas a las ceremonias. Dando un total de **181 horas**, sin contemplar ceremonias.

## Objetivo del sprint

El objetivo del sprint es implementar la funcionalidad de **pedir taxi**, enfocándose exclusivamente en que los usuarios puedan realizar solicitudes de taxis desde la aplicación. Este sprint se centrará en desarrollar la interfaz de solicitud, gestionando los eventos necesarios para registrar y procesar cada pedido de taxi.

## Equipo scrum

- Carrasco, Francisco
- De Dios, Guillermo
- Beluzzo Gino, Federico
- Escudero, Octavio

- Galiano, Milagros
- Marchiori, Nicolás
- Pizarro, Emilio
- Vega, Candela Belén

## Sprint backlog

- Pedir taxi

## Tareas por sprint

User Story	Tareas	Duración (horas)
Pedir taxi	Diseñar la interfaz de usuario para solicitar taxi	10
	Crear base de datos y tablas	10
	Programar la interfaz de usuario para solicitar taxi	12
	Investigar obtención de datos de geoposicionamiento	8
	Investigar visualización del mapa	8
	Integrar la API de mapas en la interfaz	12
	Validar la solicitud de taxi (GPS activado y ubicación válida)	10
	Integrar geolocalización para obtener la ubicación actual del pasajero	10
	Conectar frontend con backend (Enviar solicitud)	8
	Programar pruebas unitarias	8
	Implementar visualización del mapa	12
	Implementar visualización taxi disponible	10
	Realizar pruebas de componentes	10
	Realizar pruebas de integración	12
	Corrección de errores detectados	10
	Crear documentación de usuario final	8



	Realizar pruebas de aceptación	10
<b>Total de horas</b>		<b>168</b>

## Definición de done

Definimos que la US está terminada y lista para mostrar al Product Owner para su aprobación, cuando cumple con los siguientes criterios:

- Diseño de interfaz revisado y aprobado
- Código:
  - Código completo y actualizado
  - Código adecuado al documento de estilos
  - Código refactorizado
  - Código comentado
  - Código en el repositorio con control de versiones
- Documentación de usuario final actualizada y completa
- Código testeado:
  - Pruebas unitarias aprobadas y ejecutadas correctamente
  - Pruebas de aceptación ejecutadas y aprobadas
  - Pruebas de usabilidad ejecutadas y aprobadas
  - Pruebas de aceptación exitosas
- Corrección de errores realizada
- Gestión de excepciones realizada
- Cumplimiento de normas de privacidad y seguridad

## Bibliografía

- Ken Schwaber & Jeff Sutherland (2020). *La Guía Scrum*. Recuperado de <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>
-