



TP N° 8: SCRUM – Release and Sprint Planning – Planificación de Release y de Sprint

Cátedra: Ingeniería de Software

Docentes:

- Covaro, Laura Ines
- Massano, Maria Cecilia
- Crespo, María Mickaela

Alumnos:

- Diaz, Nahuel - 65144
- Santamarina, Manuel - 65503
- Andermatten, Alexis Ariel - 70287
- Cidale, Renato Raul - 65961
- Velasco, Melisa Paola - 58496
- Villanueva, Michael - 72171

Fecha de entrega: 16 /09/2020

Grupo: 2

Curso: 4K3

Plan de Release

A continuación se presenta el Product Backlog que contiene las Historias de Usuario ordenadas por prioridad acordadas por el Product Owner para el sistema Taxi Mobile:

Prioridad	Nombre US	Como	Quiero	Para	Estimación
1	Loguear taxista	taxista	loguearme	poder visualizar los pedidos de taxis.	2
2	Buscar taxis cercanos	pasajero	quiero ver cuáles con los taxis más cercanos a mi ubicación	pedir el taxi que más me convenga.	3
3	Pedir taxi	pasajero	poder pedir un taxi seleccionando el más conveniente de un mapa	asegurarme de que el taxi está cerca.	5
4	Notificar a taxista y a central pedido de taxi	pasajero	enviar una notificación al momento en que solicitó un viaje	que el taxista me busque y la central esté enterada del pedido.	3
5	Ver ubicación del pasajero	taxista	ver la ubicación del pasajero que ha solicitado un viaje	poder ir a buscarlo	5
6	Ocupar taxi	taxista	marcar que el taxi se encuentra ocupado	no recibir pedidos de servicio que no podrá atender.	2

Universidad Tecnológica Nacional - Facultad Regional Córdoba

7	Liberar taxi	taxista	marcar que el taxi se encuentra ocupado	no recibir pedidos de servicio que no podrá atender.	2
8	Registrar central de taxis	administrador de central	dar de alta la central	poder tomar viajes con Taxi-mobile	1
9	Loguear pasajero	pasajero	loguearme	poder visualizar los taxis más cercanos	2
10	Marcar taxi como fuera de servicio	taxista	quiero marcar que el taxi se encuentra fuera de servicio	no recibir pedidos de servicio que no podrá atender.	2
11	Ver mapa de taxis	administrador de la central	quiero ver la ubicación de todos los taxis de la central y si tienen viajes en curso	saber la disponibilidad actual	5
					Total 32

Para la planificación del release, el equipo tomará en cuenta:

- Las relaciones entre las historias de usuarios.
- Funcionalidades comunes entre las historias de usuarios agrupadas en cada sprint
- Esfuerzo estimado para realizarlas
- Duración de los Sprints definidas por el equipo sujeto a horas de disponibilidad (2 semanas)

Para agrupar las US en cada sprint se tomó en cuenta las funcionalidades consignadas al MVP. Ellas quedaron definidas como:

- Loguear taxista
- Buscar taxis cercanos
- Pedir taxi
- Notificar al taxista y central
- Ver ubicación del pasajero
- Ocupar taxi
- Liberar taxi

A continuación se realizarán las US que no forman parte del MVP que son:

- Registrar central de taxi
- Loguear pasajero
- Marcar taxi como fuera de servicio
- Ver mapa de taxis.

División de HU divididas en sprints

Sprint 1	
Nombre US	Complejidad
Loguear taxista	2
Buscar taxis cercanos	3
Pedir taxi	5
Notificar a taxista y a central pedido de taxi	3
TOTAL	13

Sprint 2	
Nombre US	Complejidad
Ver ubicación del pasajero	5
Ocupar taxi	2
Liberar taxi	2
Registrar central de taxi	1
TOTAL	10

Sprint 3	
Nombre US	Complejidad
Loguear pasajero	2
Marcar taxi como fuera de servicio	2
Ver mapa de taxi	5
TOTAL	9

El tiempo estimado para la entrega del MVP sería de un mes ya que las historias de usuario que forman parte del MVP suman un total de 23 puntos y la velocidad estimada del equipo es de 13 puntos por sprint.

Sprint Planning

Minuta

- **Sprint Nro. 1**
- **Duración del Sprint en días:** 10
- **Objetivo del Sprint:** Se plantea como objetivo para este primer Sprint lograr que el rol del taxista pueda loguearse en el sistema, y que el rol pasajero pueda realizar la búsqueda de taxis cercano, pedir el taxi que desee y una vez realizado, enviar la notificación del pedido tanto al taxista como a la central correspondiente.
- **Equipo Scrum:**
 - Diaz, Nahuel
 - Santamarina, Manuel
 - Andermatten, Alexis Ariel
 - Cidale, Renato Raul
 - Velasco, Melisa Paola
 - Villanueva, Michael
- **Capacidad del Equipo en Horas Ideales:** 204

Definition of Done del equipo del proyecto

Definición de Hecho para el Equipo
<ul style="list-style-type: none"><input type="checkbox"/> Diseño revisado<input type="checkbox"/> Código completo<ul style="list-style-type: none"><input type="checkbox"/> Código refactorizado<input type="checkbox"/> Código con formato estándar<input type="checkbox"/> Código Integrado<input type="checkbox"/> Código Inspeccionado<input type="checkbox"/> Código Documentado<input type="checkbox"/> Código con pruebas unitarias<input type="checkbox"/> Probado<ul style="list-style-type: none"><input type="checkbox"/> Prueba de unidad hecha<input type="checkbox"/> Prueba de integración hecha<input type="checkbox"/> Prueba de Regresión hecha<input type="checkbox"/> Plataforma probada<input type="checkbox"/> Prueba de Aceptación realizada<input type="checkbox"/> Cero defectos conocidos

Disponibilidad por integrante para el primer sprint (10 días - 2 semanas)

Integrantes	Días disponible	Horas por día	Total de horas
Diaz, Nahuel	9	5	32
Santamarina, Manuel	9	5	32
Andermatten, Alexis Ariel	9	4	36
Cidale, Renato Raul	9	4	36
Velasco, Melisa Paola	9	4	36
Villanueva, Michael	9	4	32
Total de Horas para el Sprint			234

Condiciones de contexto: se tuvo en cuenta que hay un feriado en medio. Cada uno tiene 4 horas disponibles por día a excepción de dos integrantes del equipo, los cuales tienen 5 horas disponibles, dentro de los días donde se trabajará para el primer sprint, obteniendo un total de 234 horas ideales.

Sprint Backlog

Enunciado del ítem de Product Backlog	Estimación de US	Tarea	Horas ideales estimadas
Como taxista quiero loguearme para poder visualizar los pedidos de taxis.	2	Diseñar UI	5
		Analizar API de Facebook	4
		Implementar código backend	6
		Implementar código frontend	6
		Implementar base de datos	6
		Escribir pruebas unitarias	5
		Refactorizar	4
		Integrar código	2
		Crear Pruebas	5
		Ejecutar Pruebas	3
		Documentar código	3
		Total	49
Como pasajero quiero ver cuáles con los taxis más cercanos a mi ubicación para pedir el taxi que más me convenga.	3	Diseñar UI	5
		Analizar cómo resolver el geoposicionamiento y tiempos	5
		Implementar el geoposicionamiento y tiempos	12
		Implementar resto del backend	7
		Implementar resto del frontend	6
		Implementar base de datos	5
		Escribir pruebas unitarias	5
		Integrar código	5

		Crear Pruebas	8
		Ejecutar Pruebas	4
		Documentar código	3
		Total	65
Como pasajero quiero poder pedir un taxi seleccionando el más conveniente de un mapa para asegurarme de que el taxi está cerca.	5	Diseñar UI	6
		Implementar frontend de la vista del mapa	7
		Implementar backend de la vista del mapa	5
		Implementar resto del backend	6
		Implementar resto del frontend	7
		Implementar base de datos	6
		Escribir pruebas unitarias	6
		Refactorizar	5
		Integrar código	3
		Crear pruebas	8
		Ejecutar pruebas	3
		Documentar código	2
		Total	64
Como pasajero quiero enviar una notificación al momento en que solicitó un viaje para que el taxista me busque y la central esté enterada del pedido.	3	Diseñar formato de notificación	3
		Analizar notificaciones push	4
		Implementar notificaciones push	5
		Implementar backend	6
		Implementar frontend	6
		Implementar base de datos	2
		Escribir pruebas unitarias	3

		Refactorizar	4
		Integrar código	2
		Crear pruebas	5
		Ejecutar pruebas	3
		Documentar código	1
		Total	44
	13 Story Points	Horas Ideales	222

Velocidad del equipo estimada: 13 p

Velocidad del equipo estimada en horas: 4 hs disponibles por día * 6 integrantes del equipo * 10 días del sprint = 240 hs

Las 12 horas sobrantes serán usadas en caso de inconvenientes o retrasos que puedan llegar a suceder

Aclaración

Para el sprint backlog se utilizó el formato de hoja de cálculo ya que no utilizamos ninguna herramienta de gestión de sprint. También se podría haber utilizado alguna y como resultado se hubiese obtenido un sprint backlog con el siguiente formato:

Story	Tasks	
As a coach, I can assign swimmers to events for a meet	Determine rules about who can swim in which events 6	Specify acceptance tests to show how this should work 8
	Design user interface 16	Code user interface 8
	. . .	
As a swimmer, I can update my demographics	Specify acceptance tests 5	Change view-only demographics page to allow edits 6

Bibliografía

Cohn, Mike — Agile Estimating and Planning — Editorial Prentice Hall 2006 — Capítulos 13, 14, 15 y 16.