



Universidad Tecnológica Nacional  
Facultad Regional de Córdoba  
Ingeniería en Sistemas de Información  
Cátedra de Ingeniería de Software

Profesora: Ing. MELES, Silvia Judith.

Profesor: Ing. ROBLES, Joaquín Leonel.

Profesora: Ing. CRESPO, María Mickaela.

<p><b>PRÁCTICO 8</b> <b>SCRUM – PLANIFICACIÓN DE RELEASE Y DE SPRINT</b></p>
--

CURSO: 4K1

Grupo N°: 8

Carrizo, Lucas. Legajo: 72818.

Castellina, Bruno. Legajo: 72193.

Lerda Venchiarutti, Martín. Legajo: 71655

Lozada Bonelli, Juan. Legajo: 72665

*Fecha de Entrega: 15 de septiembre de 2020*

## Release Planning

### *Condiciones de contexto de la release planning*

Respecto a la duración de cada sprint, dado que no existe en el equipo un historial de trabajo y la duración estimada en la que se realizan las tareas, sumado a la incertidumbre generada por algunas tecnologías utilizadas para desarrollar el MVP, se tomó la decisión de planear realizar iteraciones de 2 semanas de duración (considerando únicamente 10 días hábiles).

En principio, dada la situación del equipo respecto a datos históricos y que no es factible realizar algunas iteraciones para observar la velocidad, para realizar la estimación de velocidad del equipo se optó por utilizar el método “Hacer un pronóstico” enunciado en el capítulo 16 del libro *Agile Estimating and Planning*.

Se siguieron los siguientes pasos:

- A. Estimación de las horas disponibles por individuo: 4 horas
- B. Estimación de las horas disponibles en una iteración: 200 hs (4 hs \* 5 integrantes \* 10 días por iteración)
- C. Stories que pueden ingresar en una iteración:
  - **Loguear Taxista** (24 hs)
    - Diseñar interfaz gráfica = 3 hs
    - Desarrollar frontend (código del lado del cliente) Javascript = 4 hs
    - Desarrollar backend para logueo de taxista = 4 hs
    - Realizar pruebas unitarias = 2 hs
    - Refactorizar código = 3 hs
    - Realizar pruebas de usuario = 2 hs
    - Realizar pruebas de regresión = 2 hs
    - Realizar “push” de código a repositorio = 1 hs
    - Generar documentación de la US = 3 hs

- **Pedir Taxi (57 hs)**

Acordar con el Product Owner el estilo de la interfaz = 3 hs

Desarrollar módulo de comunicación de backend con sistema de geoposicionamiento = 8 hs

Solicitar credenciales y desarrollar módulo de integración con Google Maps = 5 hs.

Diseñar la interfaz gráfica de la US = 6 hs

Desarrollar frontend (código del lado del cliente) Javascript = 4 hs

Desarrollar backend para la lógica del pedido del taxi = 4 hs

Realizar pruebas unitarias = 2 hs

Refactorizar código = 5 hs

Realizar pruebas de regresión = 4 hs

Realizar pruebas de usuario = 3 hs

Diseñar y ejecutar casos de prueba = 8 hs

Realizar push de código a repositorio = 1 hs

Generar documentación de la US = 4 hs.

- **Ver ubicación del pasajero (38 hs)**

Diseñar interfaz gráfica de la US = 6 hs

Desarrollar código frontend Javascript = 5 hs

Desarrollar backend de comunicación y sincronización de ubicación = 5 hs

Revisión de código = 2 hs

Realizar pruebas unitarias = 2 hs

Refactorizar código = 3 hs

Realizar pruebas de usuario = 4 hs

Diseñar y ejecutar casos de prueba de validación de la funcionalidad = 4 hs

Realizar pruebas de regresión = 3hs

Realizar push de código a repositorio = 1 hs

Generar documentación de la US = 3 hs.

- **Ocupar Taxi (21 hs)**

Diseñar interfaz gráfica = 2 hs

Desarrollar frontend (código del lado del cliente) Javascript = 4 hs

Desarrollar backend para cambio de estado del taxi = 4 hs

Revisión de código = 2 hs

Realizar pruebas unitarias = 2 hs

Refactorizar código = 3 hs

Realizar pruebas de usuario = 1 hs

Realizar pruebas de regresión = 1 hs

Realizar push de código a repositorio = 1 hs

Generar documentación de la US = 1 hs.

- **Notificar a taxista y a central pedido de taxi (21 hs)**

Desarrollar módulo de notificaciones push para notificar a dispositivo de taxista y central = 8 hs

Desarrollar código para integrar el módulo al backend = 3 hs

Revisión de código = 2 hs

Realizar pruebas unitarias = 1 hs

Refactorizar código = 2 hs

Realizar pruebas de usuario = 1 hs

Realizar pruebas de regresión = 2 hs

Realizar push de código a repositorio = 1 hs

Generar documentación de la US = 1 hs

Dando un total de horas estimadas de **161 horas**

En base a los Story Points de cada US:

- Loguear Taxista: 2 SP
- Pedir Taxi: 5 SP
- Ver ubicación del pasajero: 5 SP
- Ocupar Taxi: 2 SP

- Notificar a taxista y a central pedido de taxi: 3 SP

Siendo en total **17 Story Points**.

El restante de horas de la iteración (39 horas) no planificadas, se dejan sin asignar a ninguna tarea, con el fin de contemplar posibles imprevistos o dudas que puedan surgir durante el sprint, supliendo así también los tiempos de más requeridos por la inexperiencia del equipo.

Brindando como resultado:

- *Estimación de las horas disponibles: 4 horas*
- *Estimación de las horas disponibles en una iteración: 200 horas*
- *Velocidad del equipo estimada: 17 SP*

### *Plan de Release*

En base a esto, el plan de release del MVP de “*Taxi Mobile*” contempla el lanzamiento de la release en los siguientes sprints:

Release	Prioridad	Frase Verbal US	Estimación
1	1	Pedir Taxi	5
	2	Ocupar Taxi	2
	3	Liberar Taxi	2
	4	Loguear Taxista	2
	5	Ver ubicación del pasajero	5
2	6	Buscar taxis cercanos	3
	7	Notificar a taxista y a central pedido de taxi	3

Total de SP de la release: 22

## Sprint Planning

### *Minuta de planificación del Sprint Planning*

<b>Sprint N°: 1</b>
<b>Duración del Sprint (días): 10</b>
<b>Objetivo del Sprint:</b> Realizar la implementación para el PO de las funcionalidades de la aplicación relacionadas a solicitar un taxi a través de un mapa interactivo, permitirle al taxista visualizar la ubicación de su pasajero y establecer el estado actual de su vehículo en “Ocupado” o “Libre”. Las US incluidas son: pedir taxi, ocupar taxi, liberar taxi, loguear taxista y ver ubicación del pasajero..
<b>Equipo SCRUM:</b> <ul style="list-style-type: none"><li>• Carrizo, Lucas Ezequiel</li><li>• Castellina, Bruno Alfredo</li><li>• Lerda Venchiarutti, Martín</li><li>• Lozada Bonelli, Juan</li><li>• Paillet, Lucas Mario</li></ul>
<b>Capacidad del Equipo en Horas Ideales: 161 horas</b>

### *Sprint Backlog*

US	SP	Tarea	Horas de la tarea	Total hs US
Pedir taxi	5	1. Acordar con el Product Owner el estilo de la interfaz	3	57
		2. Desarrollar módulo de comunicación de backend con sistema de geoposicionamiento	8	

		3. Solicitar credenciales y desarrollar módulo de integración con Google Maps 4. Diseñar la interfaz gráfica de la US 5. Desarrollar frontend (código del lado del cliente) Javascript 6. Desarrollar backend para la lógica del pedido del taxi 7. Realizar pruebas unitarias 8. Refactorizar código 9. Realizar pruebas de regresión 10. Realizar pruebas de usuario 11. Diseñar y ejecutar casos de prueba 12. Realizar push de código a repositorio 13. Generar documentación de la US	5 6 4 4 2 5 4 3 8 1 4	
Ocupar taxi	2	1. Diseñar interfaz gráfica 2. Desarrollar frontend (código del lado del cliente) Javascript 3. Desarrollar backend para cambio de estado del taxi 4. Revisión de código 5. Realizar pruebas unitarias 6. Refactorizar código 7. Realizar pruebas de usuario 8. Realizar pruebas de regresión 9. Realizar push de código a repositorio 10. Generar documentación de la US	2 4 4 2 2 3 1 1 1 1	21
Liberar taxi	2	1. Diseñar interfaz gráfica 2. Desarrollar frontend (código del lado del cliente) Javascript 3. Desarrollar backend para cambio de estado del taxi 4. Revisión de código 5. Realizar pruebas unitarias 6. Refactorizar código 7. Realizar pruebas de usuario 8. Realizar pruebas de regresión 9. Realizar push de código a repositorio 10. Generar documentación de la US	2 4 4 2 2 3 1 1 1 1	21
Loguear taxista	2	1. Diseñar interfaz gráfica 2. Desarrollar frontend (código del lado del cliente) Javascript 3. Desarrollar backend para logueo de taxista 4. Realizar pruebas unitarias	3 4 4 2	24

		5. Refactorizar código 6. Realizar pruebas de usuario 7. Realizar pruebas de regresión 8. Realizar “push” de código a repositorio 9. Generar documentación de la US	3 2 2 1 3	
Ver ubicación del pasajero	5	1. Diseñar interfaz gráfica de la US 2. Desarrollar código frontend Javascript 3. Desarrollar backend de comunicación y sincronización de ubicación 4. Revisión de código 5. Realizar pruebas unitarias 6. Refactorizar código 7. Realizar pruebas de usuario 8. Diseñar y ejecutar casos de prueba de validación de la funcionalidad 9. Realizar pruebas de regresión 10. Realizar push de código a repositorio 11. Generar documentación de la US	6 5 5 2 2 3 4 4 3 1 3	38
<b>Total SP</b>	16		<b>Total horas</b>	161

#### *Condiciones de contexto de la sprint planning*

- Tal como se indicó para la release planning, son 4 las horas laborales diarias de cada uno de los integrantes del equipo.
- Si bien existen 200 horas como máximo destinadas a cada iteración, en este primer sprint se completaron los story point de velocidad estimada para el equipo en 161 horas.
- Esta diferencia de 39 horas no planificadas (divisibles aproximadamente en 7.8 horas por integrante) permiten establecer un margen de horas ante cualquier posible imprevisto que tenga el equipo durante el transcurso del sprint, ya sea producto de la propia inexperiencia en trabajar con Scrum o los inconvenientes que puedan surgir con algunas tecnologías que se deben implementar en la aplicación y en las cuales no tenemos total dominio.



### *Bibliografía*

- Cohn, Mike. *Agile Estimating and Planning*. Recuperado de [https://uv.frc.utn.edu.ar/pluginfile.php/373061/mod\\_resource/content/2/agile-estimating-and-planning-algunos%20capitulos.pdf](https://uv.frc.utn.edu.ar/pluginfile.php/373061/mod_resource/content/2/agile-estimating-and-planning-algunos%20capitulos.pdf)
- Ken Schwaber y Jeff Sutherland. *La guía de Scrum (julio 2013)*. Recuperado de <https://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-es.pdf>