



UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL CÓRDOBA

Ingeniería en Sistemas de Información

Trabajo Práctico Nº8:

“Release y Sprint Planning”

Cátedra: Ingeniería de Software

Curso: 4K2

Docentes:

- Meles, Judith (Adjunto)
- Massano, Cecilia (JTP)
- Robles, Joaquin (Ayudante 1ra)

Número de grupo: 1

Integrantes:

- Barrionuevo, Gastón 72413
- Carranza, Exequiel 60848
- D' Agostino, Julieta 69880
- Floreano, Micaela 72276
- Pellis, Lucas 63025
- Screpnik, Julieta 74842

Release y Sprint Planning

Minuta Sprint Planning	
<p>Sprint N°: 1</p> <p>Duración del Sprint: 15 días</p> <p>Objetivo del Sprint: Ofrecer al cliente la funcionalidad de solicitar un servicio de taxi por medio de la búsqueda de un taxi libre más cercano a la ubicación en la que se encuentra en ese momento. También se ofrecerá la funcionalidad de generar una notificación al conductor del vehículo al momento de ser confirmada la solicitud de servicio por parte del cliente. Se contempla para ello las US de Loguear taxista, Buscar taxis cercanos, Pedir taxi, Notificar a taxista solicitud de taxi, ver ubicación del pasajero, ocupar taxi y liberar taxi</p> <p>Capacidad del equipo en horas ideales: 300 horas</p> <p>Equipo Scrum: Barrionuevo, Gastón Carranza, Exequiel D'Agostino, Julieta Floreano, Micaela Pellis, Lucas Screpnik, Julieta</p>	
DEFINICIÓN DE HECHO (DONE)	SPRINT BACKLOG
<ul style="list-style-type: none">• Diseño revisado y aprobado por el PO• Código completo<ul style="list-style-type: none">○ Código versionado en el repositorio en la rama correspondiente○ Código siguiendo el manual de buenas prácticas○ Refactorización de código○ Revisión de código• Probado<ul style="list-style-type: none">○ Pruebas unitarias hecha<ul style="list-style-type: none">■ Camino feliz■ Camino alternativo■ Cambios de estado■ Datos guardados correctamente○ Pruebas de integración	<ul style="list-style-type: none">• Buscar taxis cercanos• Pedir taxi• Notificar a taxista solicitud de taxi• Loguear taxista• Ocupar taxi• Liberar taxi• Ver ubicación del pasajero <p>La división de las User Stories en tareas con sus respectivas horas de trabajo se encuentra al final del documento.</p>

hecha ○ Prueba de Regresión hecha ● Prueba de Aceptación realizada ● Con defectos conocidos con prioridad y severidad bajas	
--	--

Consideraciones externas:

- La capacidad del equipo en horas ideales se estimó tomando en cuenta que es el primer proyecto del equipo trabajando juntos, además no posee experiencia profesional en cuanto a las tecnologías que solicita el usuario para la creación de la aplicación y es novato en el uso de metodología Scrum.
- Horas laborales diarias por integrante: 5h
- Partimos de la base que existe un sprint anterior (Sprint 0) en el cual se realizó ciertas actividades que nos permiten poder trabajar en el MVP:

Tareas Sprint 0:

- Diseño de prototipos
- Armar esquema de desarrollo
- Crear y configurar repositorio
- Establecer reglas de buenas prácticas
- Diagrama de clases
- Diseñar la BD
- Generar ambiente de testing, desarrollo y producción.
- Generar BD de testing desarrollo y producción.
- (SPIKE) Investigación API Google Maps
 1. Investigar cómo se integra esta tecnología en el proyecto.
 2. Crear prueba de concepto para poder integrar API de Maps.
- (SPIKE) Investigar y realizar prueba de concepto para utilizar geoposicionamiento.
- (SPIKE) Investigar integración con la API de Facebook
- Diagramar máquina de estados relacionando el estado al color correspondiente

USER STORIES PASAJERO

- Buscar taxis cercanos → 50h
 1. Diseñar el prototipo - Buscar taxi cercanos. - 5h
 2. Desarrollo de interfaz - Crear pantalla para buscar taxi según filtros. - 8h
 3. Desarrollar la lógica de negocio - Buscar taxis cercanos según filtros. - 10h
 4. Crear íconos de taxi en el mapa - Diseño. - 1h
 5. Crear casos de prueba. - 5h
 6. Realizar pruebas unitarias. - 10h
 7. Realizar pruebas de integración - 3h
 8. Realizar prueba de Regresión - 7h
 9. Realizar prueba de aceptación - 1h
- Pedir taxi → 46h
 1. Diseñar el prototipo para pedir taxi. - 5h
 2. Generar mapa - Usar API Google Maps para posicionar los taxis cercanos. - 15h
 3. Desarrollar lógica de negocio - Pedir taxi cercanos. 3h
 4. Crear casos de prueba - 5h
 5. Realizar pruebas unitarias. - 8h
 6. Realizar pruebas de integración - 2h
 7. Realizar prueba de Regresión - 7h
 8. Realizar prueba de aceptación - 1h
- Notificar a taxista solicitud de taxi → 44h
 1. Diseñar el prototipo de notificación. - 3h
 1. Lógica para establecer servicio de notificación push - 16h
 2. Diseñar mensajes de notificación. - 2h
 3. Crear casos de prueba - 5h
 4. Realizar pruebas unitarias. - 8h
 - 5 Realizar pruebas de integración - 2h
 6. Realizar prueba de Regresión - 7h
 7. Realizar prueba de aceptación - 1h

USER STORIES TAXISTA

- Loguear taxista → 40h
 1. Diseñar el prototipo para loguear taxista - 4h
 1. Establecer mecanismo de manejo de sesión. - 6h
 2. Utilizar solución de logueo con facebook. - 3h
 3. Diseño de interfaz para logear taxista. - 2h
 4. Lógica de negocio para loguear a un taxista. - 8h
 5. Crear casos de prueba - 5h
 6. Realizar pruebas unitarias. - 5h
 7. Realizar pruebas de integración - 2h
 8. Realizar prueba de Regresión - 7h
 9. Realizar prueba de aceptación - 1h
- Ocupar taxi → 36h
 1. Diseñar el prototipo - 3h
 1. Desarrollo de interfaz taxi libre o con estado pendiente - 8h
 2. Lógica de negocio, actualizando su estado. - 6h
 3. Crear casos de prueba - 4h
 4. Realizar pruebas unitarias. - 5h
 5. Realizar pruebas de integración - 2h
 6. Realizar prueba de Regresión - 7h
 7. Realizar prueba de aceptación - 1h
- Liberar taxi → 36h
 1. Diseñar el prototipo para liberar taxi. - 3h
 1. Desarrollo de interfaz liberar taxi con pedido en curso. - 8h
 2. Lógica de negocio, actualizando su estado. - 6h
 3. Crear casos de prueba - 4h
 4. Realizar pruebas unitarias. - 5h
 5. Realizar pruebas de integración - 2h
 6. Realizar prueba de Regresión - 7h
 7. Realizar prueba de aceptación - 1h
- Ver ubicación del pasajero → 48h
 1. Diseñar el prototipo para pedir taxi. - 5h
 1. Desarrollo de interfaz utilizando API Google Maps- 12h
 2. Lógica de negocio, obtener datos de ubicación a partir del sistema de geoposicionamiento y marcar la posición del pasajero en el mapa. - 8h
 3. Crear casos de prueba - 5h
 4. Realizar pruebas unitarias. - 8h
 5. Realizar pruebas de integración - 2h
 6. Realizar prueba de Regresión - 7h
 7. Realizar prueba de aceptación - 1h

Release Planning

Consideraciones externas:

- Consideramos que el MVP debe estar en un 1 sprint, lo que demostramos en nuestro release planning.
- Consideramos sprints de 15 días.
- Consideramos que el equipo puede realizar 22 SP por sprint.
- Consideramos la priorización de US que se requieren para realizar el MVP.

US

taxista

- Loguear taxista (2)
- Ocupar taxi (2)
- Liberar taxi (2)
- Ver ubicación del pasajero (5)

Pasajero

- Buscar taxis cercanos (3)
- Pedir taxi (5)
- Notificar a taxista solicitud de taxi (3)

SP totales=22

User Story Map:

<i>Sprint 1 - Release 1</i>					
<i>Prioridad</i>	<i>US que entran</i>	<i>Riesgo</i>	<i>Tiempo estimado</i>	<i>Story Points</i>	<i>Responsable</i>
1	Loguear taxista	Bajo	40 hs	2	Carranza/ Barrionuevo
2	Buscar taxis cercanos	Medio	50 hs	3	Screpnik/D'Ago stino
3	Pedir taxi	Alto	46 hs	5	Screpnik/D'Ago stino/Carranza/ Pellis
4	Notificar a taxista solicitud de taxi	Medio	44 hs	3	Barrionuevo/Flo reano
5	Ver ubicación del pasajero	Alto	48 hs	5	Pellis/Floreano/ Screpnik/D'Ago stino
6	Ocupar taxi	Medio	36 hs	2	Carranza/Pellis
7	Liberar taxi	Medio	36 hs	2	Barrionuevo/ Floreano

Bibliografía consultada:

- Cohn, Mike — Agile Estimating and Planning — Editorial Prentice Hall 2006 — Capítulo 12 y 14.
- Guía de Scrum.