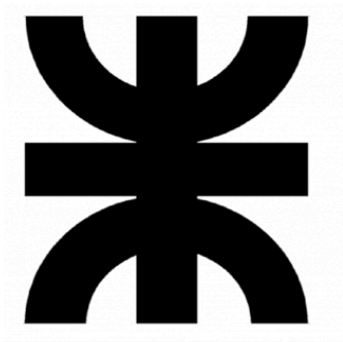


UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL CÓRDOBA



DEPARTAMENTO DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN

Ingeniería y Calidad de Software

TRABAJO PRÁCTICO N° 8

Planificación de Release y de Sprint

Docentes

- Ing. Judith Meles
- Ing. Mickaela Crespo
- Constanza Garnero

Curso 4K1

Integrantes Grupo N° 8:

- González, Anael Luz - 92566
- Egea, Francisco - 87207
- Franco, Ramiro - 89535
- Richard, Ian - 90254
- Moroni, Nicolás - 89818
- Varela, Mateo - 91048
- Spaccesi, Luca - 91792

Fecha de presentación: 10/10/24

Índice

Plan de Release del MVP.....	3
Objetivo.....	3
Distribución.....	3
Capacidad del equipo.....	4
Consideraciones de contexto.....	5
Equipo de trabajo.....	5
Horas lineales.....	5
Duración de los Sprints.....	6
Inicio del Sprint 1.....	6
Ceremonias de Scrum.....	6
Otras consideraciones.....	7
Minuta - Sprint Planning.....	8
Definición de hecho (DoD).....	9
Sprint Backlog.....	10

Plan de Release del MVP

A continuación se presenta el plan de release para la liberación de la primera versión del producto (MVP). El producto mínimo viable definido contempla la funcionalidad mínima y necesaria para validar la hipótesis planteada, permitiendo así obtener retroalimentación directa por parte del cliente.

Objetivo

Entregar una versión funcional del sistema (MVP) que permita a los pasajeros solicitar taxis cercanos y a los taxistas aceptar solicitudes y manejar el estado del taxi.

Distribución

El primer release estará dividido en 3 sprints de 2 semanas cada uno. En total abarcará 6 semanas. Cada sprint comprenderá un conjunto de User Stories de forma que la suma de los Story Points de cada sprint sean similares entre sí. A continuación presentamos las User Stories que incluye:

Plan de Release del MVP				SP Totales: 21
Sprint 1	Pedir taxi SP: 5	Notificar a Taxista SP: 2	SP Sprint: 7	
Sprint 2	Ver ubicación de pasajero SP: 5	Ocupar Taxi SP: 2	SP Sprint: 7	
Sprint 2	Loguear Taxista SP: 2	Buscar taxis cercanos SP: 3	Liberar taxi SP: 2	SP Sprint: 7

Sprint 1: Pedir taxi y Notificar taxista (SP: 7)

Sprint 2: Ver ubicación de pasajero y Ocupar taxi. (SP: 7)

Sprint 3: Loguear taxista, Ver taxis cercanos y Liberar taxi. (SP: 7)

Total SP: 21

Capacidad del equipo

PERSONAS	Días disponibles	Días para otras actividades de SCRUM	Horas esfuerzo por día	Horas de esfuerzo disponible
Gonzalez, Anael	10	2	2-4	0 (SCRUM MASTER)
Egea, Francisco	8	2	4-6	24-36
Franco, Ramiro	10	2	4-5	32-40
Richard, Ian	10	2	3-5	24-40
Moroni, Nicolás	10	2	2-3	16-24
Varela, Mateo	10	2	3-4	24-32
Spaccesi, Luca	8	2	4-5	24-30
			Total	144-202

Para la confección de esta tabla se consideró la **capacidad del equipo en un sprint**, es decir, 10 días hábiles (considerando 2 semanas de 5 días cada una).

Los días para otras actividades SCRUM consideran la *sprint planning*, *sprint review* y *sprint retrospective*.

Consideraciones de contexto

Equipo de trabajo

El equipo de desarrollo está compuesto por programadores, testers, un DevOps y un Scrum Master, con roles bien definidos pero flexibles según las necesidades de cada sprint.

- **Programadores:** Se encargan del desarrollo de las funcionalidades del MVP y desarrollo de tests unitarios. En caso de ser necesario, pueden colaborar con otras áreas.
- **Testers:** Son responsables de validar la funcionalidad de las User Stories desarrolladas en cada sprint. Están dedicados a la ejecución de casos de prueba de sistema, pruebas de integración y pruebas de regresión. También pueden apoyar en tareas de documentación o monitoreo de las métricas de calidad.
- **Scrum Master:** Actúa como facilitador de los sprints, asegurando que el equipo mantenga el ritmo adecuado y que no existan bloqueos en el flujo de trabajo. Se asegura también de que las ceremonias de Scrum se realicen a tiempo y que el equipo mantenga la visión del MVP.
Dado que el Scrum Master no tiene un rol técnico, sus horas no deben incluirse en la estimación de tiempo para la implementación o pruebas de una User Story.
- **DevOps:** El ingeniero DevOps será encargado del despliegue y en un futuro de los despliegues automatizados.

Horas lineales

Las horas de compromiso de cada miembro del equipo pueden variar ligeramente dependiendo de las necesidades específicas de cada sprint y de la disponibilidad de cada persona. Pero, en general, la idea es mantener un ritmo constante para evitar sobrecargar al equipo y asegurar una buena productividad.

Las horas ideales de trabajo por día, incluyen las horas dedicadas a reuniones como Dailys, Planning, Retrospective y Reviews.

Para las horas ideales, se consideran días laborales (Lunes, Martes, Miércoles, Jueves y Viernes), sólo los hábiles según el calendario oficial de Argentina.

No están planificadas vacaciones por parte de durante el periodo de los sprints, lo que garantiza una mayor estabilidad en la disponibilidad del equipo. Esto reduce los riesgos de posibles retrasos en la entrega del MVP y facilita la planificación precisa de las tareas.

Duración de los Sprints

El ciclo de desarrollo se estructurará en tres sprints de 2 semanas cada uno. Esta duración se ajusta al equipo para el desarrollo, prueba y mejora de las funcionalidades clave del MVP. Al mismo tiempo, otorga espacio para recibir retroalimentación temprana y aplicar mejoras en los siguientes sprints.

Inicio del Sprint 1

El primer sprint está previsto para comenzar en la **segunda semana de octubre de 2024**, lo que marcará el inicio oficial del desarrollo del MVP. Desde ese momento, el equipo seguirá un calendario preestablecido para las ceremonias de Scrum y la entrega de cada sprint.

Ceremonias de Scrum

- **Daily:** Estas reuniones se realizarán después del horario de almuerzo (14 hs) ya que la mayoría de los integrantes cursan en la facultad temprano. En esta reunión el equipo informará sobre su progreso, posibles bloqueos y lo que planean hacer durante el día. La duración ideal será de 15 minutos.
- **Sprint Planning:** Al inicio de cada sprint se dedicarán 2hs a la planificación de las User Stories a desarrollar. Durante esta sesión, se reconsideran las User stories identificadas en el plan de release del MVP en función de la retroalimentación del cliente. Se reasignan, en caso de ser necesario, los Story Points. Además, se desglosan las tareas de cada una de las User stories planificadas para el sprint.
- **Sprint Reviews:** Al finalizar cada sprint (viernes a las 16:00 hs), se hará una demostración de lo desarrollado, donde se espera tener una funcionalidad completa y revisada que pueda ser mostrada tanto a los stakeholders como a los clientes. Tendrá una duración máxima de 1 hora.
- **Sprint Retrospectives:** Se realizarán al cierre de cada sprint (viernes a las 18:00 hs), justo después de la Sprint Review, para discutir las mejoras en el proceso, identificar

cuellos de botella y optimizar el flujo de trabajo para los siguientes sprints. Tendrá una duración máxima de 45 minutos.

Otras consideraciones

Se dispone de 1 teléfono móvil con iOS y 2 con Android para realizar las pruebas de las US.

La User Story que implica la notificación del taxista no considera la integración con la central de taxis ni su desarrollo. Por ello, se considera que su estimación de Story Points baja a 2.

Debido a que no hay información en el MVP que indique la necesidad de crear la base de datos, consideramos según nuestro criterio que “mockear” taxistas será suficiente para probar la hipótesis planteada con el MVP.

Todos los miembros del equipo de trabajo están estudiando en la UTN Facultad Regional Córdoba y cursan a la mañana. Sus horarios laborales empiezan luego de cursar, es decir, a partir de las 13 hs. y sus jornadas laborales no pueden ser tan extensas debido a que tienen que estudiar para diversas asignaturas que demandan mucho tiempo.

Minuta - Sprint Planning

Minuta para el sprint 1

Sprint Nro. 1

Duración del Sprint en días: *10 días (5 días hábiles en 2 semanas)*

Objetivo del Sprint: *El sistema debe permitir que el pasajero pida un taxi y enviar una notificación push al taxista.*

Equipo Scrum:

- González, Anael Luz - *Scrum Master*
- Egea, Francisco - *Desarrollador*
- Franco, Ramiro - *Desarrollador*
- Richard, Ian - *Desarrollador*
- Moroni, Nicolás - *DevOps*
- Varela, Mateo - *Tester*
- Spaccesi, Luca - *Tester*

Capacidad del Equipo en Horas Ideales: 144-202 horas

Definición de hecho (DoD)

Definición de hecho
Código <ul style="list-style-type: none"><input type="checkbox"/> Código documentado siguiendo buenas prácticas.<input type="checkbox"/> Código revisado por otros desarrolladores (code review)<input type="checkbox"/> Código con formato estandarizado según ESLint<input type="checkbox"/> Código alojado en el repositorio, en la rama (branch) de desarrollo correspondiente<input type="checkbox"/> Código organizado aplicando separación de responsabilidades (separation of concerns)<input type="checkbox"/> Código refactorizado para mejorar su legibilidad y eficiencia
Interfaz gráfica <ul style="list-style-type: none"><input type="checkbox"/> Interfaz gráfica diseñada y documentada (prototipo)<input type="checkbox"/> Interfaz gráfica revisada, probada y validada
Pruebas <ul style="list-style-type: none"><input type="checkbox"/> Código probado exhaustivamente mediante pruebas unitarias<input type="checkbox"/> Casos de prueba funcionales y no funcionales diseñados, ejecutados y documentados<input type="checkbox"/> Pruebas de integración realizadas para asegurar que los módulos interactúan correctamente<input type="checkbox"/> Pruebas de regresión ejecutadas para asegurarse de que los cambios no introducen errores en el sistema existente.<input type="checkbox"/> Entorno de pruebas correctamente configurado, actualizado y equivalente al entorno de producción
Despliegue <ul style="list-style-type: none"><input type="checkbox"/> Sistema desplegado en el entorno de staging para revisión final<input type="checkbox"/> Plan de rollback en caso de errores en producción documentado y listo para ejecutar
Documentación <ul style="list-style-type: none"><input type="checkbox"/> Manual de usuario actualizado con instrucciones claras, ejemplos y capturas de pantalla<input type="checkbox"/> Manual de usuario adaptado al contexto y lenguaje específico del cliente, evitando jerga técnica
Cero defectos conocidos
La US debe pasar todos los criterios de aceptación

Sprint Backlog

User stories	Tareas (junto con horas ideales)	Horas ideales
Pedir Taxi (SP: 5)	<div>Implementar lógica Mapa 7</div> <div>Codificar UI Mapa 4</div> <div>Investigar Servicios de Mapa 16</div> <div>Diseñar UI Mapa 2</div> <div>Crear tests automatizados 6</div> <div>Diseñar Casos de Prueba 8</div> <div>Mockear taxistas 2</div> <div>Realizar pruebas de regresión 4</div> <div>Codificar UI Solicitud 4</div> <div>Diseñar UI Solicitud 3</div> <div>Codificar lógica de selección de taxi 12</div> <div>Integrar geolocalización 8</div> <div>Crear Manual de Usuario 2</div> <div>Ejecutar Casos de Prueba 7</div> <div>Realizar pruebas de integración 4</div>	89
Notificar a taxista solicitud de taxi (SP: 2)	<div>Implementar Servicio de Notificaciones Push 5</div> <div>Agregar guía a manual de usuario 2</div> <div>Crear tests automatizados 6</div> <div>Diseñar Casos de Prueba 6</div> <div>Ejecutar Casos de Prueba 5</div> <div>Realizar pruebas de regresión 4</div> <div>Configurar Notificaciones Push 5</div> <div>Investigar Envío de Mensajes Push 10</div> <div>Preparación de ambiente y pruebas 10</div> <div>Realizar pruebas de integración 4</div>	57
TOTAL SP: 7		TOTAL HORAS: 146

El backlog del Sprint 1 incluye las siguientes User Stories desglosadas en tareas junto con sus respectivas consideraciones.

User Story: Pedir Taxi

Como pasajero quiero poder pedir un taxi seleccionando el más conveniente de un mapa para asegurarme de que el taxi está cerca.

- Implementar lógica de Mapa - 7 horas
 - Integración de la solución investigada para el servicio de mapa.
- Integrar geolocalización - 8 horas
 - Contempla integrar la geolocalización del teléfono con el sistema
- Codificar UI Mapa - 4 horas
- Investigar Servicios de Mapa - 16 horas
 - Considera buscar posibles APIs y librerías externas
- Diseñar UI Mapa - 2 horas
- Crear tests automatizados - 6 horas
- Codificar UI Solicitud - 4 horas
 - Contempla el desarrollo de componentes respectivos a la selección de un taxi
- Diseñar UI Solicitud - 3 horas
- Codificar lógica de selección de taxi - 12 horas
 - Contempla validaciones correspondientes

- *Crear Manual de Usuario - 2 horas*
- *Diseñar Casos de Prueba - 8 horas*
 - *Casos de prueba de sistema*
- *Ejecutar Casos de Prueba - 7 horas*
- *Mockear taxistas - 2 horas*
 - *No hay información en el MVP que indique la necesidad de crear la base de datos, consideramos según nuestro criterio que “mockear” taxistas será suficiente para probar con la hipótesis del MVP*
- *Realizar pruebas de regresión - 4 horas*
- *Realizar pruebas de integración - 4 horas*

Total: 89 horas lineales

User Story: Notificar a taxista solicitud de taxi

Como pasajero quiero enviar una notificación al momento en que solicito un viaje para que el taxista me busque.

- *Implementar servicio de notificaciones Push - 5 horas*
 - *Integración del servicio mediante el desarrollo de un módulo reutilizable*
- *Agregar guía a manual de usuario - 2 horas*
- *Desarrollar y configurar Notificaciones Push - 5 horas*
 - *Contempla la estructura de la notificación, su contenido, y el seteo del destinatario correspondiente.*
- *Crear tests automatizados - 6 horas*
- *Diseñar casos de prueba - 6 horas*
 - *Casos de prueba de sistema*
- *Ejecutar casos de prueba - 5 horas*
- *Preparación de ambiente y pruebas - 10 horas*
 - *Despliegue y configuración del entorno staging para pruebas*
- *Investigar envío de notificaciones push - 10 horas*
- *Realizar pruebas de regresión - 4 horas*
- *Realizar pruebas de integración - 4 horas*

Total: 57 horas lineales

Consideración: El MVP no incluye la implementación e integración con la central de taxis.