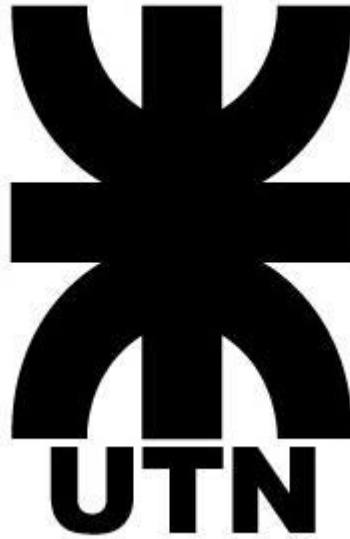


Universidad Tecnológica Nacional
Cátedra Ing. y Calidad de Software
Trabajo Práctico N°8



Tema: SCRUM - Planificación de Release y de Sprint

Fecha de entrega: 14/05/2024

Curso: 4K4

Docentes:

- Ing. Laura Covaro
- Ing. Mickaela Crespo
- Ing. Georgina González
- Constanza Garnero

Integrantes:

- Abadia, Martina (89633) - martiabadiaa@gmail.com
- Gimenez Scarponi, Federico (89340) - fedegimenez144@gmail.com
- Gutierrez Cuetti, Ulises Matheo (89874) - ulimatheo@gmail.com
- Lambertucci, Juan Pablo (90606) - jp_lambertucci@outlook.com
- Morales, Emiliano Javier (54112) - emiarthas@gmail.com
- Oliva Paez, Javier Nicolas (90664) - nicoolivapaez21@gmail.com
- Sanchez, Valentina (89442) - valensanchez.utn@gmail.com
- Vigliocco, Mariano (89434) - nvigliocco@gmail.com

ÍNDICE

MVP (Minimum Viable Product)	2
PLAN DE RELEASE	4
Backlog	4
SPRINT PLANNING	8
Ceremonias de SCRUM	9
Minuta Sprint Planning	10
Sprint Backlog	12
BIBLIOGRAFIA.....	14

MVP (Minimum Viable Product)

Objetivo:

- Permitir que los clientes (pasajeros) puedan trasladarse al lugar que desean solicitando un taxi cercano.
- Permitir a los taxistas ver la ubicación del pasajero que solicita un viaje para llevarlo.
- Permitir a los taxistas manejar el estado del taxi.

Justificación:

Se va a desarrollar una aplicación mobile, incluimos las siguientes user stories:

- **Loguear taxista:** debido a que el taxista debe estar logueado para poder recibir, visualizar los pedidos y poder seleccionar alguno.
- **Ocupar taxista:** debido a que es necesario que el taxista informe su estado de ocupado para no recibir más pedidos.
- **Liberar taxi:** debido a que es necesario que el taxista informe su estado de libre para recibir más pedidos.
- **Ver ubicación del pasajero:** debido a que es absolutamente necesario para que el taxista pueda ver en dónde está el pasajero y **poder** iniciar su viaje para buscarlo.
- **Buscar taxis cercanos:** posibilita a los pasajeros poder ver los taxis más cercanos para poder pedir un taxi a su conveniencia. Además nos permite resolver la obtención de datos de geoposicionamiento y tiempos.
- **Pedir taxi:** permite efectivamente pedir un taxi para el pasajero e iniciar la transacción
- **Notificar al taxista solicitud de taxi:** permite enviar una notificación que el pasajero ha solicitado un viaje, tanto para que el taxista inicie el viaje, además nos permite resolver el envío de mensaje de tipo push.

No se incluyen los aspectos de loguear pasajero ya que los mismos van a ser cargados por base de datos, ni tampoco ver mapas taxi debido a que es demasiado complejo para agregarlo en nuestro mvp y no nos aporta gran valor de negocio, tampoco se realizara la implementacion con la interfaz de facebook para loguear taxista, debido a que no pertenece al mvp por su complejidad y sera resuelto a futuro por la historia de usuario 'Loguear Pasajero'. No se incluirá todo lo relacionado a la central de taxis debido a que no compete a este mvp porque no nos sirve para validar la hipótesis propuesta

Criterio:

Las user stories incluidas permiten validar la idea de negocio en el mercado, focalizando en la funcionalidad que pone en contacto a pasajeros con taxistas. Si bien los taxistas deben poder iniciar sesión para vincularlos con los datos de su vehículo, el análogo para el pasajero no es requisito para poder realizar un pedido. La funcionalidad relacionada con la gestión de taxis por parte de la central no aporta el valor significativo para la validación del mercado por lo que se dejará para futuras iteraciones.

Frases verbales de las **historias de usuario** incluidas en el MVP (junto con Story Points):

Taxista

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

Pasajero

- Buscar taxis cercanos - 3.
- Pedir taxi - 5.
- Notificar al taxista solicitud de taxi - 3.

PLAN DE RELEASE

Cantidad de Sprints: 3 Sprints.

Duración de Sprints: 2 semanas cada uno.

Duración del Proyecto: 1 mes y medio (6 semanas)

Sprint	U.S que entran en el Sprint (junto con Story Points)
1	<ul style="list-style-type: none"> • Loguear taxista - 2 • Buscar taxis cercanos - 3 • Pedir taxi - 5
2	<ul style="list-style-type: none"> • Ver ubicación del pasajero - 5 • Ocupar taxi - 2 • Liberar taxi - 2 • Notificar a taxista solicitud de taxi - 3
3	El 3er Sprint se utilizará para corregir errores, adaptarse a la retroalimentación y mejorar diseños UI/UX

Backlog

US	Tareas	Suma Esfuerzo Estimado
Pedir Taxi (5)	<ul style="list-style-type: none"> - Investigar como poder generar un mapa o conectarse a una API y mi propia ubicación (10hs) - Diseñar Arquitectura del SW (5hs) - Crear Esquema BD (5hs) - Codificar Backend, conectando con la BD y con la obtención de ubicación de geoposicionamiento y el mapa (22hs) - Codificar Lógica para los cálculos de tiempo estimado (4hs) - Diseñar UI/UX con el mapa para seleccionar y Pedir taxi (4hs) - Codificar UI/UX y conectar con los datos de backend (20hs) - Realizar pruebas unitarias (4hs) - Realizar pruebas de integración (4hs) - Realizar pruebas de aceptación (2hs) 	86 hs

	<ul style="list-style-type: none"> - Realizar Documentación (2hs) - Revisión de un integrante del equipo (Code review) (4hs) 	
Buscar Taxis Cercanos (3)	<ul style="list-style-type: none"> - Investigar cómo obtener datos de geoposicionamiento de los taxistas en tiempo real (8hs) - Realizar modificaciones en la Arquitectura del SW (2hs) - Realizar modificaciones necesarias en la BD (2hs) - Codificar Backend, conectando con la BD y con la obtención de ubicación de geoposicionamiento (15hs) - Codificar Lógica para los cálculos de las distancias con los 5 más cercanos con estado disponible (4hs) - Diseñar UI/UX (4hs) - Codificar UI/UX y conectar con los datos de Backend (15hs) - Realizar pruebas unitarias (4hs) - Realizar pruebas de integración (4hs) - Realizar pruebas de aceptación (2hs) - Realizar Documentación (2hs) - Revisión de un integrante del equipo (Code review) (4hs) 	66 hs
Logear Taxista (2)	<ul style="list-style-type: none"> - Realizar modificaciones necesarias en la Arquitectura del SW (2hs) - Realizar modificaciones necesarias en la BD (2hs) - Codificar Backend, Utilizando las prácticas de seguridad necesarias para conectar con BD y validar los datos de usuario y contraseña (10hs) - Diseñar UI/UX con iconos y campos a completar (2hs) - Codificar UI/UX (8hs) - Realizar pruebas unitarias (3hs) - Realizar pruebas de integración y seguridad (3hs) - Realizar pruebas de aceptación (1hs) - Realizar Documentación (1hs) - Revisión de un integrante del equipo (Code review) (4hs) 	36 hs
	<ul style="list-style-type: none"> - Investigar cómo recibir datos del sistema de geoposicionamiento online y cómo visualizarlo en el mapa 	44hs

Ver ubicación del pasajero (5)	(8hs) - Codificar Backend, conectando con la BD y con la obtención de ubicación de geoposicionamiento y el mapa (10hs) - Realizar modificaciones necesarias en la Arquitectura del SW (2 hs) - Realizar modificaciones necesarias en la BD (2hs) - Diseñar UI/UX (2 hs) - Codificar UI/UX y conectar con los datos de backend (15hs) - Realizar pruebas unitarias (4hs) - Realizar pruebas de integración (4hs) - Realizar pruebas de aceptación (2hs) - Realizar Documentación (2hs) - Revisión de un integrante del equipo (Code review) (4hs)	
Ocupar Taxi (2)	-Diseñar interfaz de ocupación de taxi y creación de iconos (5hs) - Realizar modificaciones necesarias en la Arquitectura del SW (2hs) - Realizar modificaciones necesarias en la BD (2hs) - Codificar cambio de estado de entidad (2hs) - Diseñar UI/UX con iconos y campos a completar (2hs) - Codificar UI/UX (8hs) - Realizar pruebas unitarias (3hs) - Realizar pruebas de integración y seguridad (3hs) - Realizar pruebas de aceptación (1hs) - Realizar Documentación (1hs) - Revisión de un integrante del equipo (Code review) (4hs)	34hs
Liberar Taxi (2)	-Diseñar interfaz de liberación de taxi y creación de iconos (5hs) - Realizar modificaciones necesarias en la Arquitectura del SW (2hs) - Realizar modificaciones necesarias en la BD (2hs) - Codificar cambio de estado de entidad (2hs) - Diseñar UI/UX con iconos y campos a completar (2hs)	34 hs

	<ul style="list-style-type: none"> - Codificar UI/UX (8hs) - Realizar pruebas unitarias (3hs) - Realizar pruebas de integración y seguridad (3hs) - Realizar pruebas de aceptación (1hs) - Realizar Documentación (1hs) - Revisión de un integrante del equipo (4hs) 	
Notificar a taxista solicitud de taxi (3)	<ul style="list-style-type: none"> - Investigar cómo enviar notificaciones push (5hs) - Realizar modificaciones necesarias en la Arquitectura del SW (2hs) - Realizar modificaciones necesarias en la BD (2hs) - Codificar notificación (2hs) - Diseñar UI/UX (2hs) - Codificar UI/UX(6hs) - Realizar pruebas unitarias (3hs) - Realizar pruebas de integración y seguridad (3hs) - Realizar pruebas de aceptación (1hs) - Realizar Documentación (1hs) - Revisión de un integrante del equipo (Code review) (4hs) 	31hs

SPRINT PLANNING

Miembros del equipo:

1. Abadía, Martina.
2. Gimenez Scarponi, Federico.
3. Gutierrez Cuetti, Ulises Matheo.
4. Lambertucci, Juan Pablo.
5. Morales, Emiliano.
6. Oliva Paez, Javier Nicolas.
7. Sanchez, Valentina.
8. Vigliocco, Mariano.

Contexto del equipo:

Se deben considerar los feriados existentes en el período:

25 de mayo de 2024 - Día de la Revolución de Mayo: como es día sábado, no afecta al trabajo del equipo en el plan de release.

17 de junio de 2024 - Paso a la Inmortalidad del Gral. Don Martín Miguel de Güemes: día no laborable para el equipo. Afecta al trabajo del equipo.

20 de junio de 2024 - Paso a la inmortalidad del Gral. Manuel Belgrano: día laboral para el equipo. No hay cambios en horarios.

21 de junio de 2024 - Feriado con fines turísticos: día no laborable para el equipo. Afecta al trabajo del equipo.

Capacidad del equipo:

Nos basamos en los principios de la filosofía ágil para el trabajo continuo y de calidad. Somos un equipo auto-organizado que busca la excelencia técnica a través de:

- Comunicación directa y efectiva.
- Responsabilidad.
- Trabajo en equipo de calidad.
- Ambiente de confianza.

Persona	Días disponibles	Horas por día	Horas de esfuerzo disponible
Juan	10	5-6	56
Nico	10	2-3	28
Fede	10	3-4	36
Marti	10	3-4	34
Nano	10	3-4	36
Uli	10	3-4	36
Emi	10	3-4	36
Valen	10	3-4	38
TOTAL			300

Ceremonias de SCRUM

- **Sprint Planning:** normalmente, en un sprint de duración de un mes, el Sprint Planning tiene una duración de 8 horas. Como nuestros Sprint tendrán una duración de dos semanas, podemos asumir que el Sprint Planning debería durar como máximo **4 horas** de cada miembro por cada Sprint.
- **Daily meetings:** las “daily” por lo general tienen una duración de 15 minutos por día. Además, se realizan a lo largo de todo el Sprint. En un Sprint de dos semanas tenemos 10 días hábiles (sin tener en cuenta excepciones como feriados). Por ende, calculando, $15 \text{ minutos} * 10 \text{ días hábiles} = 150 \text{ minutos}$. Estos 150 minutos equivalen a **2 horas y media** de cada miembro dedicadas a las daily meetings por Sprint.
- **Sprint Review:** por lo general, se dedican 4 horas reloj al Sprint Review en un Sprint de un mes. Como nuestros Sprint tienen una duración de 2 semanas, asumimos que el Sprint Review tendrá una duración de **2 horas** de cada miembro por Sprint.
- **Sprint Retrospective:** se utilizan 3 horas dedicadas para un Sprint Retrospective en un Sprint de un mes. Como nuestros Sprint tienen una duración de dos semanas, se asume que se necesitarán **1,5 horas** de cada miembro para realizar el Sprint Retrospective en nuestro Sprint Planning.

Minuta Sprint Planning

Sprint Nro: 1

Duración del Sprint en días: 14

Objetivo del Sprint:

Desarrollar e implementar las funcionalidades clave del MVP que permitan a los pasajeros solicitar taxis cercanos. Incluimos el logueo del taxista como base para poder trabajar con este rol en el próximo Sprint y establecer así la base funcional del sistema. Este Sprint también servirá para medir la velocidad del equipo y ajustar la planificación para los siguientes Sprints en función de los resultados obtenidos.

Equipo Scrum:

- **Abadía, Martina:** Diseño de interfaces UI/UX, Desarrollo Front-End.
- **Gimenez Scarponi, Federico:** Arquitectura de Base de Datos.
- **Gutierrez Cuetti, Ulises Matheo:** Desarrollo Back-End.
- **Lambertucci, Juan Pablo:** Desarrollo y tester Back-End.
- **Morales, Emiliano:** Desarrollo y tester Back-End.
- **Oliva Paez, Javier Nicolás:** Desarrollo Front-End.
- **Sanchez, Valentina:** Arquitectura de Software (**Scrum Master**).
- **Vigliocco, Mariano:** Desarrollo Front-End.

Capacidad del Equipo en Horas Ideales: 220

Definición de Hecho para el Equipo:

- ☐ **Diseño revisado**
 - ☐ Se ha revisado la interfaz y se verificó que cumple con los criterios de paletas de colores, formas de iconos, etc.
 - ☐ Se ha revisado la interfaz y se verificó que es intuitiva y fácil de usar, con la curva de aprendizaje lo más suave posible para cumplir con la simplicidad requerida por el pasajero.
 - ☐ Se ha revisado la interfaz y el diseño cumple con la menor cantidad de interacciones por parte del taxista.

- ☐ Se aplicaron las sugerencias de mejora.
- ☐ **Código Completo**
 - ☐ Código refactorizado: Se ha revisado el código y se refactoriza de acuerdo con las necesidades de las User Stories.
 - ☐ Código con formato estándar: Código llevado a formato estándar de acuerdo con el documento de estilos de código del equipo.
 - ☐ Código comentado: Se realizaron comentarios en cada clase/método que requiera una explicación para su entendimiento.
 - ☐ Código en el repositorio: Se aplica la práctica de integración continua para cada User Story.
 - ☐ Código inspeccionado: El código se inspeccionó por más de un miembro del equipo y cumple con todos los lineamientos de código explicitados en el documento de estilos de código
- ☐ **Documentación de Usuario Actualizada:**

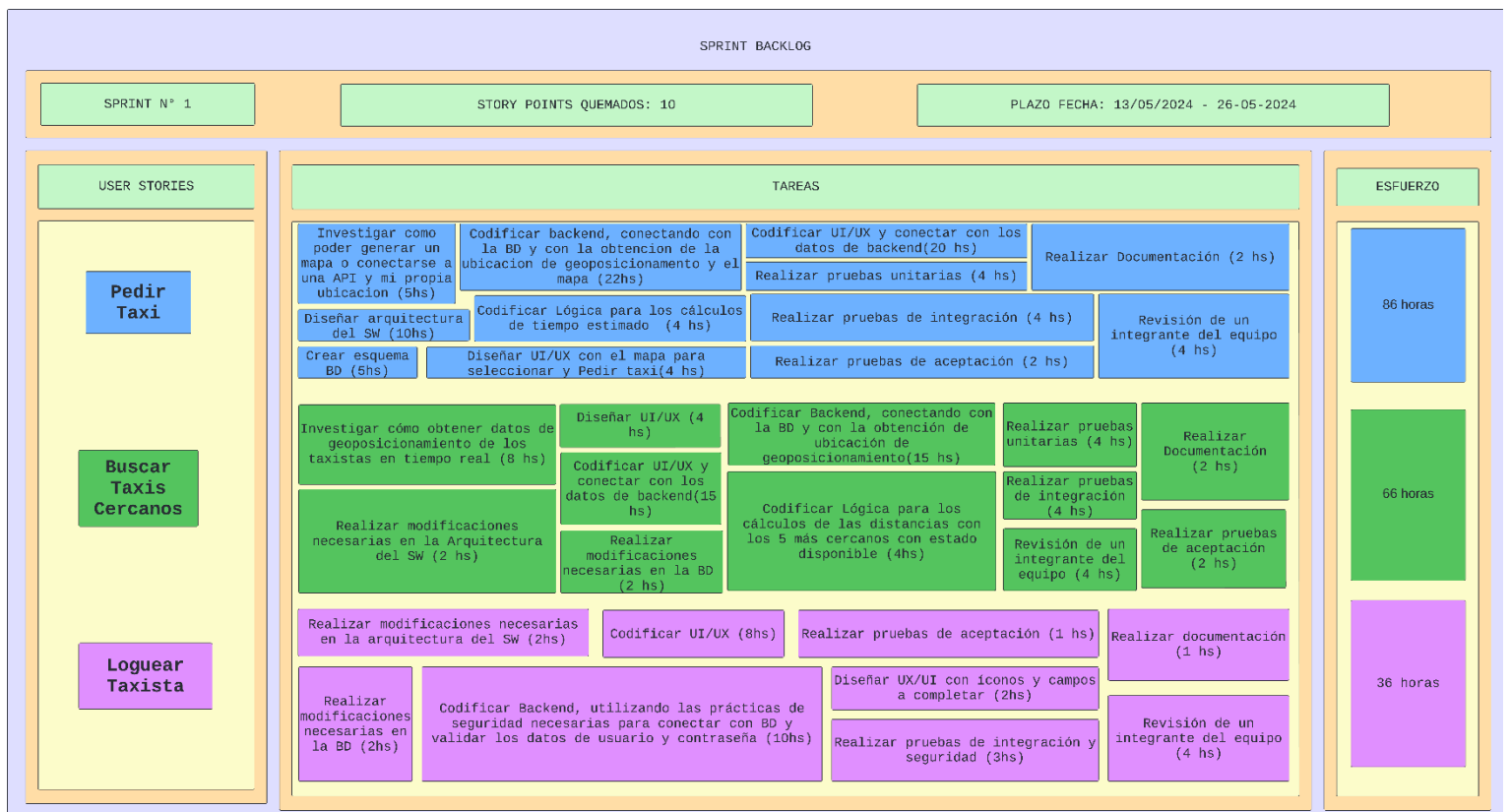
La documentación de usuario se encuentra actualizada con toda la información asociada a la usabilidad del software, seguridad, recomendaciones, etc.
- ☐ **Documentación técnica actualizada:**

La documentación técnica está actualizada para reflejar todos los nuevos cambios introducidos que es de interés documentar (documentación de arquitectura, documentación de arquitectura de base de datos).
- ☐ **Pruebas**
 - ☐ Pruebas de unidad aceptadas:
 - ☐ Verificar que se envíen notificaciones al taxista y este las reciba.
 - ☐ Verificar que el taxista pueda rechazar o aceptar solicitudes correctamente.
 - ☐ Verificar que el servicio de búsqueda de taxis cercanos devuelva resultados cercanos a la ubicación del pasajero.
 - ☐ Verificar que el inicio de sesión se realice correctamente.
 - ☐ Pruebas de regresión aceptadas:
 - ☐ Verificar que las credenciales antiguas de los taxistas sigan siendo válidas luego de cambios en la aplicación.
 - ☐ Comprobar que el proceso de pedido de taxi siga siendo rápido luego de cambios en la interfaz de usuario.
- ☐ **Cero defectos conocidos**
- ☐ **Pruebas de Aceptación realizadas**

Sprint Backlog

US	Tareas	Suma Esfuerzo Estimado
Pedir Taxi (5)	<ul style="list-style-type: none"> - Investigar como poder generar un mapa o conectarse a una API y mi propia ubicación (10hs) - Diseñar Arquitectura del SW (5hs) - Crear Esquema BD (5hs) - Codificar Backend, conectando con la BD y con la obtención de ubicación de geoposicionamiento y el mapa (22hs) - Codificar Lógica para los cálculos de tiempo estimado (4hs) - Diseñar UI/UX con el mapa para seleccionar y Pedir taxi (4hs) - Codificar UI/UX y conectar con los datos de Backend (20hs) - Realizar pruebas unitarias (4hs) - Realizar pruebas de integración (4hs) - Realizar pruebas de aceptación (2hs) - Realizar Documentación (2hs) - Revisión de un integrante del equipo (Code review) (4hs) 	86hs
Buscar Taxis Cercanos (3)	<ul style="list-style-type: none"> - Investigar cómo obtener datos de geoposicionamiento de los taxistas en tiempo real (8hs) - Realizar modificaciones en la Arquitectura del SW (2hs) - Realizar modificaciones necesarias en la BD (2hs) - Codificar Backend, conectando con la BD y con la obtención de ubicación de geoposicionamiento (15hs) - Codificar Lógica para los cálculos de las distancias con los 5 más cercanos con estado disponible (4hs) - Diseñar UI/UX (4hs) - Codificar UI/UX y conectar con los datos de Backend (15hs) - Realizar pruebas unitarias (4hs) - Realizar pruebas de integración (4hs) - Realizar pruebas de aceptación (2hs) - Realizar Documentación (2hs) 	66hs

	- Revisión de un integrante del equipo (Code review) (4hs)	
Loguear Taxista (2)	<ul style="list-style-type: none"> - Realizar modificaciones necesarias en la Arquitectura del SW (2 hs) - Realizar modificaciones necesarias en la BD (2hs) - Codificar Backend, Utilizando las prácticas de seguridad necesarias para conectar con BD y validar los datos de usuario y contraseña (10hs) - Diseñar UI/UX con iconos y campos a completar (2hs) - Codificar UI/UX (8hs) - Realizar pruebas unitarias (3hs) - Realizar pruebas de integración y seguridad (3hs) - Realizar pruebas de aceptación (1hs) - Realizar Documentación (1hs) - Revisión de un integrante del equipo (Code review) (4hs) 	36hs



BIBLIOGRAFIA

- Cohn, Mike. (2006). "Agile Estimation and Planning." Editorial Prentice Hall.
- Schwaber, Ken, & Sutherland, Jeff. "La Guía de Scrum: La Guía Definitiva de Scrum: Las Reglas del Juego."