

Cátedra de Ingeniería de Software

Trabajo Práctico N° 8

SCRUM - Release and Sprint Planning - Planificación de Release y de Sprint

Curso: 4K4

Docente asignado:

- Robles Joaquin Leonel

Grupo N° 11:

- Casuscelli Franco	67021
- Díaz Luna Franco Emmanuel	77029
- Fraga Lucas Ezio	75909
- García Axel	70631
- Manzano Javier	73261
- Moya Joaquin	77625
- Odetti Agustín	77045
- Tavella Nahuel Nicolas	77688

Fecha de entrega: 04/05/2021

Condiciones de contexto:

Somos un equipo compuesto por 8 integrantes, cada uno capacitado en diferentes tecnologías para poder afrontar el desarrollo del producto. Cada integrante del equipo trabaja diferente cantidad de días y horas ya que todos son estudiantes universitarios y están disponibles en diferentes horarios, por lo tanto la capacidad total del equipo en un sprint es de 90 hs.

Un factor que puede influir en el desarrollo del producto es la infección de COVID-19 en uno de los integrantes del equipo debido a la pandemia mundial que estamos atravesando. Otro aspecto a tener en cuenta es que la conversación cara a cara se verá seriamente afectada.

El proyecto se realizará en 3 sprint con una duración de una semana cada uno, el equipo estará disponible 5 días en cada sprint ya que sábados y domingos son días no laborales.

Las tecnologías involucradas en el proyecto serán **React Native** para el desarrollo front-end de la app mobile. Para el backend se utilizara **.NET (C#)** y el motor de base de datos a utilizar será **SQL Server**. Utilizaremos algunas herramientas de comunicación, **Discord** para la comunicación del día a día en el equipo, **Trello** para gestionar el proyecto y un repositorio en **GitHub** como herramienta de control de versiones y colaboración entre el equipo.

A continuación se muestran los datos referidos a los días y horas de trabajo del grupo de desarrollo del proyecto:

Nombre	Días disponibles	Horas diarias disponibles	Horas diarias ideales	Capacidad por semana
Díaz Luna Franco E.	5	3	2	10
Fraga Lucas E.	5	4	3	15
García Axel	5	3	2	10
Manzano Javier	5	3	2	10
Moya Joaquin	5	3	2	10
Cassuscelli Franco	5	3	2	10
Tavella Nahuel	5	4	3	15
Odetti Agustin	5	3	2	10
Capacidad total del equipo por semana				90

Donde:

- Horas diarias ideales = Horas diarias disponibles - Horas de tiempo no productivo (alimentación / ocio / ir al baño)
- Capacidad por semana = Días disponibles * Horas diarias ideales

Plan de Release

El MVP se va a llevar a cabo en 3 iteraciones de 1 semana de duración cada una, ya que somos un equipo numeroso y por semana podemos hacer una gran cantidad de actividades.

User Stories que se van a incluir en cada sprint.

Primer Sprint

- Loguear taxista
- Ocupar taxi
- Liberar taxi

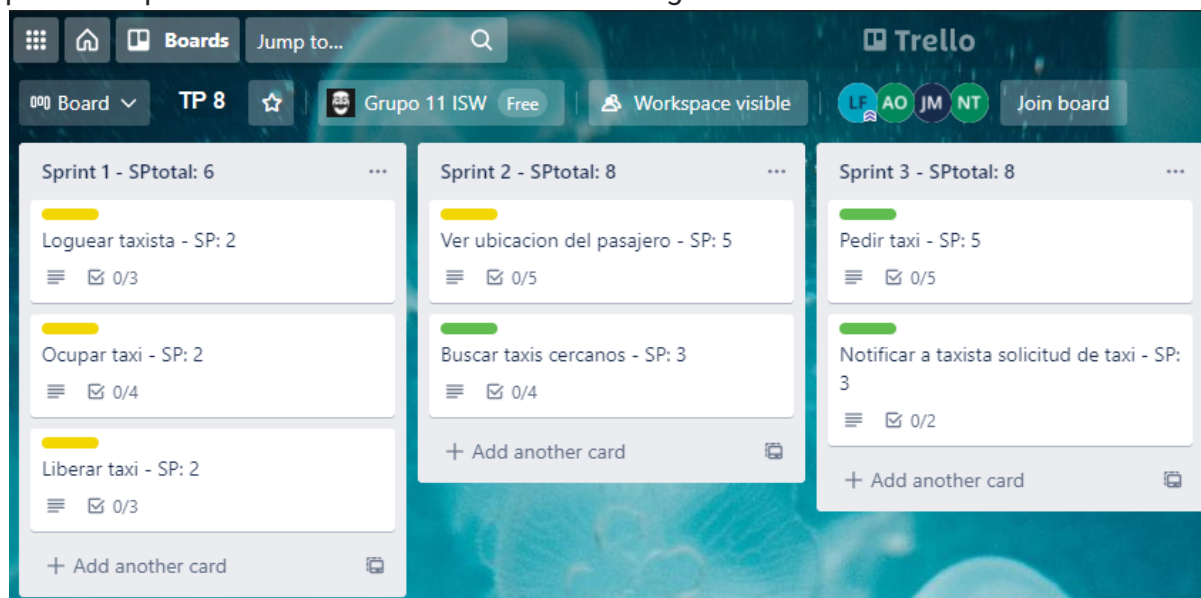
Segundo Sprint

- Ver ubicación del pasajero
- Buscar taxis cercanos

Tercer Sprint

- Pedir taxi
- Notificar a taxista solicitud de taxi

El criterio que utilizamos para seleccionar las US que contendrá cada Sprint es la prioridad que tiene cada US en el Product Backlog.



Duración del release:

La duración total del primer release será de 3 semanas, el cual incluirá el MVP.

Planificación de Sprint

Minuta de Sprint Planning																	
<p>Sprint Nro: 1</p> <p>Duración del Sprint en días: 7 días</p> <p>Objetivo del Sprint: Obtener un primer incremento del producto mediante el desarrollo de las funcionalidades Loguear taxista, Ocupar taxi y Liberar taxi. Implica crear tablas iniciales de base de datos, estructura de desarrollo de código y estructura de repositorio donde se comenzará a llevar a cabo el proyecto.</p> <p>Equipo Scrum:</p> <ul style="list-style-type: none"> • Cassuscelli Franco • Diaz Luna Franco Emmanuel • Fraga Lucas Ezio • García Axel • Manzano Javier • Moya Joaquin • Odetti Agustin • Tavella Nahuel <p>Capacidad del equipo en horas ideales: 90</p>																	
Definición de Hecho (DONE)	Sprint Backlog																
<table border="1"> <tbody> <tr> <td><input type="checkbox"/></td> <td>Criterios de aceptación realizados de las US</td> </tr> <tr> <td><input type="checkbox"/></td> <td>Cumplimiento de pruebas de aceptación de las US</td> </tr> <tr> <td><input type="checkbox"/></td> <td>Aplicación de buenas prácticas y estilos de código</td> </tr> <tr> <td><input type="checkbox"/></td> <td>Pruebas en dispositivos/navegadores pasada</td> </tr> <tr> <td><input type="checkbox"/></td> <td>Pruebas de integración pasada</td> </tr> <tr> <td><input type="checkbox"/></td> <td>Pruebas de rendimiento pasada</td> </tr> <tr> <td><input type="checkbox"/></td> <td>Código y documentación agregados y actualizados en repositorio</td> </tr> <tr> <td><input type="checkbox"/></td> <td>Cero defectos conocidos</td> </tr> </tbody> </table>	<input type="checkbox"/>	Criterios de aceptación realizados de las US	<input type="checkbox"/>	Cumplimiento de pruebas de aceptación de las US	<input type="checkbox"/>	Aplicación de buenas prácticas y estilos de código	<input type="checkbox"/>	Pruebas en dispositivos/navegadores pasada	<input type="checkbox"/>	Pruebas de integración pasada	<input type="checkbox"/>	Pruebas de rendimiento pasada	<input type="checkbox"/>	Código y documentación agregados y actualizados en repositorio	<input type="checkbox"/>	Cero defectos conocidos	<p>Adjunta la imagen abajo</p>
<input type="checkbox"/>	Criterios de aceptación realizados de las US																
<input type="checkbox"/>	Cumplimiento de pruebas de aceptación de las US																
<input type="checkbox"/>	Aplicación de buenas prácticas y estilos de código																
<input type="checkbox"/>	Pruebas en dispositivos/navegadores pasada																
<input type="checkbox"/>	Pruebas de integración pasada																
<input type="checkbox"/>	Pruebas de rendimiento pasada																
<input type="checkbox"/>	Código y documentación agregados y actualizados en repositorio																
<input type="checkbox"/>	Cero defectos conocidos																

Tareas de User Stories

Sprint 2 y 3

Ver ubicacion del pasajero - SP: 5
in list [Sprint 2 - SPtotal: 8](#)

LABELS

Taxista +

Description

Edit

Como taxista quiero ver la ubicación del pasajero que ha solicitado un viaje para poder ir a buscarlo

☒
Criterios de Aceptación
Delete

0%

☐ El celular debe contar con el sistema de posicionamiento online.
☐ El pasajero se debe visualizar gráficamente en un mapa.
☐ Probar visualizar ubicación del pasajero para un pedido de viaje cuando no está activo el sistema de ubicación del celular. (falla)

Add an item

☒
Pruebas de usuario
Delete

0%

☐ Probar visualizar ubicación del pasajero para un pedido de viaje pendiente. (pasa)
☐ Probar visualizar ubicación del pasajero para un pedido de viaje cuando no se ha solicitado ninguno. (falla)

Add an item

☒
Análisis
Delete

0%

☐ Creación de tabla Pasajero [1h]
☐ Realizar documentación de la funcionalidad [5h]
☐ Crear procedimientos almacenados para la tabla Pasajero (insert-update-delete-select) [3h]

Add an item

☒
Desarrollo
Delete

0%

☐ Consumir API de Google Maps [3h]
☐ Maquetar funcionalidad [3h]
☐ Dar estilos a la funcionalidad [2h]

Add an item

☒
Testing
Delete

0%

☐ Realizar tests unitarios [2h]
☐ Realizar test funcional [2h]
☐ Realizar prueba de integracion [2h]
☐ Documentar pruebas [2h]

Add an item

Buscar taxis cercanos - SP: 3
in list [Sprint 2 - SPtotal: 8](#)

LABELS

Pasajero +

Description

Edit

Como pasajero quiero ver cuáles con los taxis más cercanos a mi ubicación para pedir el taxi que más me convenga.

☒
Criterios de Aceptación
Delete

0%

☐ Se muestra la ubicación del taxi y el tiempo estimado para llegar al lugar en el que se encuentra el pasajero.
☐ Se mostrarán hasta 5 taxis cercanos.

Add an item

☒
Pruebas de usuario
Delete

0%

☐ Probar visualizar taxis cuando existen al menos 5 taxis libres. (pasa)
☐ Probar visualizar taxis cuando no existe ningún taxi libre (falla)

Add an item

☒
Análisis
Delete

0%

☐ Crear procedimientos almacenados para la tabla Pasajero (insert-update-delete-select) [3h]
☐ Realizar documentación de la funcionalidad [5h]

Add an item

☒
Desarrollo
Delete

0%

☐ Crear estructura de funcionalidad [3h]
☐ Realizar conexión a la BD [2h]
☐ Configurar conexión API Google Maps [6h]
☐ Filtrar por taxis liberados cercanos y seleccionar 5 [2h]
☐ Dar estilo de funcionalidad [3h]

Add an item

☒
Testing
Delete

0%

☐ Realizar tests unitarios [2h]
☐ Realizar test funcional [2h]
☐ Realizar prueba de integracion [2h]
☐ Documentar pruebas [2h]

Add an item

Pedir taxi - SP: 5
in list [Sprint 3 - SPtotal: 8](#)

LABELS

Pasajero +

Description Edit

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

☒ **Criterios de Aceptación** Delete

0%

☐ El celular debe contar con el sistema de geoposicionamiento online.

☐ El pasajero se debe visualizar gráficamente en un mapa

Add an item

☒ **Pruebas de usuario** Delete

0%

☐ Probar seleccionar el taxi de entre un conjunto de taxis visualizados (pasa)

☐ Probar seleccionar un taxi cuando no existen taxis disponibles (realizando una selección en cualquier parte de la pantalla). (falla)

☐ Probar visualizar los taxis cuando no está activo el sistema de posicionamiento del celular(falla)

Add an item

☒ **Análisis** Delete

0%

☐ Realizar documentación de la funcionalidad [3h]

Add an item

☒ **Desarrollo** Delete

0%

☐ Diseñar interfaz gráfica[7hs]

☐ Diseñar iconos[3]

☐ Integrar API de Google Maps para poder seleccionar al taxista[10h]

☐ Obtener geoposicionamiento de los dispositivos móviles[10h]

☐ Maquetar alerta que indica cuando un taxi fue pedido con exito [2h]

☐ Dar estilo a la alerta [2h]

Add an item

☒ **Testing** Delete

0%

☐ Realizar tests unitarios [2h]

☐ Realizar test funcional [2h]

☐ Realizar prueba de integracion [2h]

☐ Documentar pruebas [2h]

Add an item

Notificar a taxista solicitud de taxi - SP: 3
in list [Sprint 3 - SPtotal: 8](#)

LABELS

Pasajero +

Description Edit

Como pasajero quiero enviar una notificación al momento en que solicito un viaje para que el taxista me busque y la central esté enterada del pedido.

☒ **Criterios de Aceptación** Delete

0%

Add an item

☒ **Pruebas de usuario** Delete

0%

☐ Probar notificar al taxista y la central cuando ambos tienen conexión de internet. (pasa) ⌚ 🔍 ...

☐ Probar notificar al taxista y la central cuando ninguno tiene conexión de internet. (falla)

Add an item

☒ **Análisis** Delete

0%

☐ Realizar documentación de la funcionalidad [3h]

Add an item

☒ **Desarrollo** Delete

0%

☐ Maquetar notificacion [2h]

☐ Dar estilos a la notificacion [2h]

☐ Realizar conexión con servidor push [3h]

☐ Realizar validaciones [2h]

Add an item

☒ **Testing** Delete

0%

☐ Realizar tests unitarios [2h]

☐ Realizar test funcional [2h] ⌚ 🔍 ...

☐ Realizar prueba de integracion [2h]

☐ Documentar pruebas [2h]

Add an item

División de las User Stories en tareas

El equipo decidió elegir 3 categorías para las tareas a realizar, derivadas de las User Stories, que van a ser parte del primer sprint, las mismas son: Análisis, Desarrollo y Testing. La definición de las tareas presentadas y su clasificación en las mencionadas categorías se deben a la experiencia de trabajo del equipo, siendo esta la forma en la que se han obtenido mejores resultados, como también una mejor organización y performance.