Order Tracker

Plan de Proyecto

## Introducción

Este es el plan de proyecto de Order Tracker, una aplicación mobile con un sistema de ventas y una aplicación web con un sistema de administraciones, desarrollado en el marco de la materia 75.47 - Taller de Desarrollo de Proyectos II en la Facultad de Ingeniería.

## Consideraciones generales y metodología

Se decidió utilizar una metodología basada en SCRUM, con iteraciones de dos semanas, reuniones semanales con el equipo y los product owners, utilizando herramientas gratuitas como *TargetProcess* para todo lo que es planificación y gestión de sprints, google docs para documentación asociada al proyecto, y otras herramientas descriptas en detalle en las secciones pertinentes en Aspectos de proyecto.

## Aspectos de proyecto

En esta sección se documentan en detalle el manejo de diversos aspectos de la gestión del proyecto, como ser la planificación, administración de alcance, etc.

### Planificación

El proyecto se planifica en sprints de dos semanas. En la reunión de planeamiento semanal se pasan las stories del product backlog, registradas en el board de *TargetProcess* descrito en Requisitos y especificación funcional a otro board de *TargetProcess* de Current Sprint, que contiene las stories a trabajar durante la sprint, dividido en listas que definen los diferentes estados en los que se encuentra cada story.

Los listas de estados del board son los siguientes:

***Backlog*:** Contiene las user stories asignadas al sprint actual que todavía no han sido tomadas por nadie. Las stories en este estado transicionan al estado de In progress sólo cuando alguien empieza a trabajar en ella.

***In progress*:** Contiene las user stories en las que se está trabajando. Cada user story tiene al menos un integrante designado trabajando en ella en este estado. La transición al estado de Done se da cuando la story está completamente desarrollada y probada.

***Done*:** Contiene las stories que ya están listas para mostrar a los product owners. Transicionan a Accepted cuando los product owners verifican que se implementó lo que se necesitaba durante la demo.

Una vez que se aceptan las stories de una sprint, la lista de *Accepted* se archiva, cambiándole el nombre a *Sprint yyyy-mm-dd*.

### Alcance

El alcance con compromiso se define al principio de cada sprint en conjunto con los product owners.

El product backlog y requerimientos está definido en un board de *TargetProcess*.

### Estimaciones

Se van a realizar estimación es de esfuerzo para poder evaluar y controlar el desarrollo sobre las user stories.

*Planning Poker* es el método a utilizar para llegar a una estimación donde todo el equipo aporte su visión al respecto.

La unidad de estimación serán horas hombre.

### Costos

Los costos se manejan sólo en horas de trabajo. Debido a la modalidad de la materia, se decidió calcular un costo fijo de 10 horas semanales por integrante del grupo, lo que da un costo total de 500 horas de trabajo.

Para controlar los costos, se cargan las horas utilizadas para cualquier tarea realizada en el proyecto a través de *targetprocess.* Esto nos permite controlar semanalmente en las reuniones con los products owners las horas utilizadas frente a las horas disponibles y estimadas de trabajo semanal.

### Equipo y roles

|  |  |
| --- | --- |
| Clientes:   * Madeira, Maria Mercedes * Sapia, Belén | Desarrolladores y testers   * Arias, Damian * Laghi, Guido * Lucadei, Pablo * Piano, Sergio * Seminara, Dario |

### Interesados

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Nombre | Rol | Influencia | Forma de contacto | Frecuencia |
| 1 | Maria Mercedes Madeira | Cliente | Alta | Email, Reunión | 2 semanas |
| 2 | Belén Sapia | Cliente | Alta | Email, Reunión | 2 semanas |
| 3 | Vendedores internos | Vendedor de productos | Baja | - | - |
| 4 | Administrador | Administra el sistema | Media | - | - |
| 5 | Clientes finales | Comprador de productos | Muy baja | - | - |

### Calendarización

Contamos con 5 sprints de 2 semanas cada uno. Dentro de las mismas tenemos en consideración lo que se define en las reuniones de planificación.

Entregas con el cliente definidas para los días:

04/04/2016 - 18/04/2016 - 02/05/2016 - 16/05/2016 - 30/05/2016

### Asignación de tareas

Cada desarrollador tomará del backlog de cada sprint, tareas que estén disponibles, evaluando prioridades o dependencias que deban ser consideradas en su momento.

### Configuración y versionado

Para el control de versiones se utilizará un único repositorio GIT alojado en github.com.

Se hará una versión del tipo “release” por cada sprint finalizado.

### Tecnología

* Se usará “MySQL” como motor de base de datos.
  + Versión 5.7
* Se usará el framework “Grails” para desarrollo del backend.
  + Lenguaje Java
  + Versión 3.0.4
* Se usará “Android Studio” para el frontend mobile, respetando los lineamientos de “Material design” para diseño y estética del producto final.
  + Lenguaje Java
* Se usará “JQuery” y “Materialize” en el frontend web para el administrador.
  + Lenguaje HTML, CSS, JavaScript

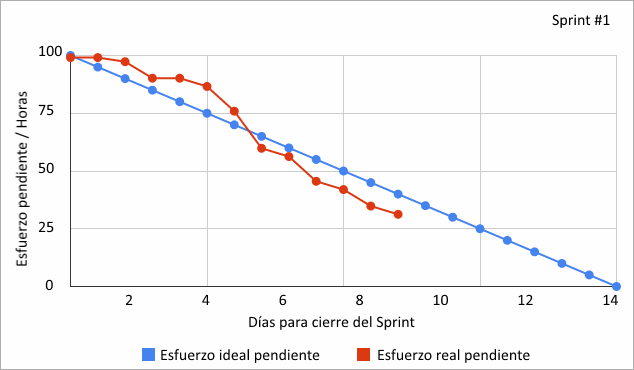
### Arquitectura y diseño técnico





### Seguimiento y control

Usaremos gráficos del tipo “*burndown*” para poder hacer un seguimiento y control del estado del sprint en cada momento. Permite tener una idea del estado actual y de la velocidad de trabajo.



### 

### Gestión de Riesgos

Se identificarán los potenciales riesgos del proyecto basados en una taxonomía.

Se efectuará un seguimiento semanal de los mismos.

A cada riesgo identificado se le indicará su probabilidad y su impacto con la siguiente nomenclatura: “bajo”, “medio” “alto”.

Su composición según esta tabla nos da la exposición (nivel de riesgo para el proyecto).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Impacto | | |
| Bajo | Medio | Alto |
| Probabilidad | Alto |  |  |  |
| Medio |  |  |  |
| Bajo |  |  |  |

En caso de tener un riesgo en el cuadrante rojo, se ejecutará el plan de contingencia previsto.

### Comunicación

* Comunicación externa:
  + Informes de avance: alcance completado (estado del backlog), riesgos de interés para el cliente, estado de los cambios solicitados, etc.
  + Reuniones con el cliente
  + Minutas: destacando puntos tratados, acuerdos y compromisos.
* Comunicación interna:
  + ***Slack*** chat como medio principal
  + Reuniones de planificación semanales
  + **E-mails**

### Calidad Pruebas

* De proceso: usamos SCRUM para nuestro workflow y visualizar el estado del mismo.

Tableros con estado de las tareas. Backlog, In Progress, Done

El estado “In Progress” ya incluirá la etapa de testing.

* De producto:
  + Criterios de aceptación para cada funcionalidad (con esto además especificamos los requerimientos no funcionales asociados).
  + Valores aceptables para las métricas identificadas.
  + Bugs cargados para seguimiento dentro de la herramienta TargetProcess.
  + Plan de pruebas:
    - una prueba unitaria para cada story implementada (mínimo).
    - Criterios de aprobación: acordar con el cliente (ej: totalidad de funcionalidad planificada implementada con X cantidad de bugs abiertos).
    - Clasificación de bugs: “críticos” y “no críticos”, estéticos, etc.

### Trazabilidad

* Matriz de trazabilidad de requerimientos: para cada requerimiento se lleva un registro del estado del mismo y de los documentos asociados (feature y stories/actividades asociadas, mockups de ui, casos de prueba y resultados de ejecución, bugs asociados, aceptación del cliente, etc)

### 

### Documentación

* Técnica:
  + diagrama de alto nivel de la arquitectura
  + servicios de la API con ejemplos de ejecución.
* De usuario:
  + descripción del uso para cada funcionalidad implementada (manual de usuario).