**Turnosync: Aplicación android para la gestión de turnos laborales**

Trabajo de Fin de Grado

GRADO EN INGENIERÍA INFORMÁTICA

**Anexo I – Plan del Proyecto Software**



Septiembre de 2018

Autor

Julio García Valdunciel

Tutor

Iván Álvarez Navia

Tabla de contenido

Tabla de ilustraciones

# Introducción

En este anexo se presentarán los detalles sobre la planificación temporal del proyecto “Turnosync: Aplicación android para la gestión de turnos laborales”.

Para el desarrollo se ha tratado de seguir una metodología ágil de tipo SCRUM, siguiendo un desarrollo iterativo e incremental. Cada iteración o “Sprint” tiene una duración aproximada de 3 semanas y en ella se pasa por distintas fases de desarrollo.

Cada iteración pretende desarrollar una parte funcional de la aplicación dando resultado a un prototipo con prestaciones funcionales y usables hasta cierto punto.

La planificación se presentará mediante un listado de tareas organizadas en *sprints* y a su vez en fases. Además se usará un diagrama de Gant para visualizar el rango temporal.

Además de la planificación temporal se comentará sobre la viabilidad del proyecto.

Este documento se organizará de la siguiente manera:

* Proceso.
* Planificación.
* Estudio de viabilidad.

# Proceso SCRUM

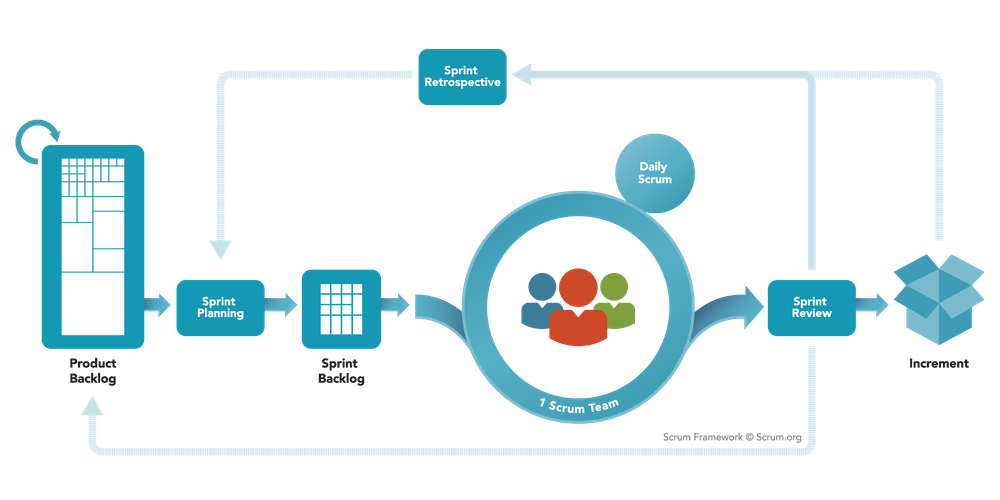


Ilustración - Metodología SCRUM

Siguiendo un desarrollo iterativo e incremental el desarrollo del proyecto se divide en iteraciones o “Sprints” de aproximadamente 3 semanas de duración. Cada sprint está orientado a desarrollar un conjunto de funcionalidades que se unifican en una funcionalidad mayor próxima a los distintos objetivos de la aplicación. Además cada sprint puede llevar menor o mayor tiempo dependiendo de la magnitud de la funcionalidad a desarrollar.

Puesto que el equipo de desarrollo está únicamente compuesto por el alumno Julio García, algunas de las características de la metodología no se cumplirán de forma rigurosa al no necesitar de coordinación con otras personas.

## 2.1 Product Backlog

Se trata de las tareas definidas a partir de los objetivos. Se han organizado en forma de historias de usuario en las cuales se indica que funcionalidad debe estar disponible para el usuario para cumplir la tarea que se propone.

Son del tipo: El <Actor> podrá <funcionalidad> para realizar <tarea>

A continuación se indican las primeras tareas propuestas:

* El usuario podrá introducir un email, contraseña y nombre a mostrar para registrarse
* El usuario podrá usar su email y contraseña para iniciar sesión
* El usuario podrá usar una cuenta de google para iniciar sesión
* El usuario podrá desplegar un menú lateral para navegar entre pantallas
* El usuario podrá desplegar un menú lateral para ver sus datos
* El usuario podrá usar el menú lateral para cerrar sesión
* El usuario podrá usar una pantalla de configuración para cambiar la configuración de la aplicación
* El usuario podrá usar la pantalla de configuración para cambiar su email y contraseña
* El usuario podrá usar la pantalla principal para crear nuevos grupos de trabajo
* El usuario responsable podrá usar una opción para invitar usuarios al grupo de trabajo
* El usuario responsable podrá usar una opción para crear turnos y asociarlos a un usuario
* El usuario podrá usar la pantalla de “mi calendario” para ver sus turnos

Posteriormente estas tareas serán divididas en *sprints* según estén asociadas entre ellas.

## Sprints

Como se ha comentado, el desarrollo se lleva a cabo en *sprints*. Estos *sprints* tienen como objetivo implementar una sección de la aplicación y se da como concluido una vez se tenga una versión funcional y utilizable que haya pasado por unas pruebas para comprobar que funciona correctamente.

Las tareas a desarrollar en los *sprints* se eligen partiendo del *product backlog* e incluyendo las tareas que estén relacionadas con la funcionalidad que se tiene como objetivo.

El desarrollo del *sprint* se ha dividido en las siguientes fases que no están presentes al completo en cada uno de ellos:

* **Aprendizaje previo**: Se adquieren conocimientos necesarios para el desarrollo del sprint.
* **Modelado de negocio**: Principalmente se caracteriza por la reunión con el cliente para la captura de requisitos, su refinamiento y para consultar la opinión del trabajo realizado.
* **Análisis y diseño**: Se realizan las tareas de ingeniería del software donde se formalizan los requisitos y se estructuran los componentes del software.
* **Implementación**: Concentran las tareas de desarrollo del código que otorgan funcionalidad al software.
* **Pruebas**: Se realizan pruebas para comprobar la correcta funcionalidad del código implementado.

## Roles

Los roles dentro de un equipo SCRUM son los siguientes:

* **Product owner**: Es el representante o representantes de los clientes que dictaminan las funcionalidades del software. En este caso sería el propio alumno que es quien se ha propuesto realizar el proyecto y ha sido quien ha tenido contacto con quien podría considerarse el cliente, un familiar con experiencia en un entorno de trabajo regido por turnos, concretamente en enfermería.
* **Scrum master**: Es la persona que gestiona el correcto desarrollo del software tratando de solucionar los problemas que surjan y asegurándose que se cumplen las reglas. Corresponde al alumno ser su propio líder puesto que es el responsable del desarrollo del proyecto.
* **Equipo de trabajo**: Son el grupo de personas con los conocimientos necesarios que desarrollan el software de forma conjunta. En este caso solo está compuesto por el alumno Julio García.

# Planificación

A partir de los objetivos se realiza una planificación aproximada de la duración del desarrollo del proyecto. Se ha utilizado la herramienta Microsoft Project.

## Calendario

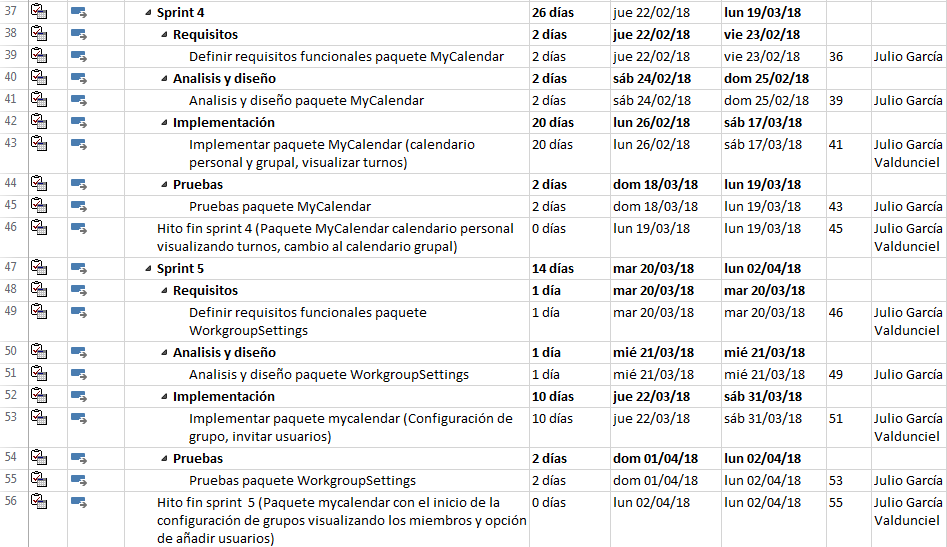
El proyecto se planifica comenzando en noviembre de 2017 y está estimado su finalización en junio de 2018. Se han tenido en cuenta en cuenta como días hábiles los fines de semana y se ha excluido un periodo de preparación de exámenes antes de las convocatorias de enero y de finales de mayo. Los días coloreados son de excepción.

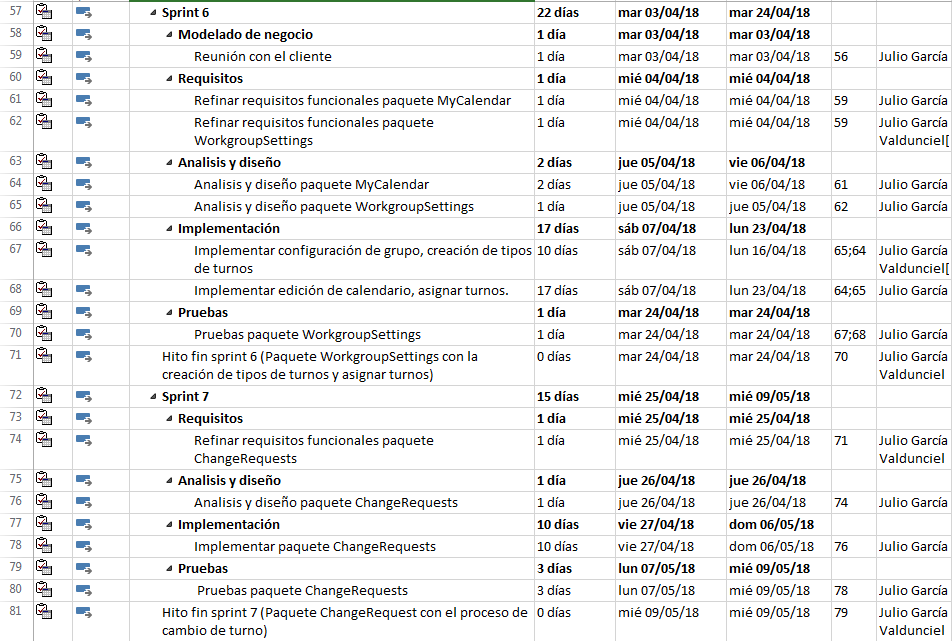
       

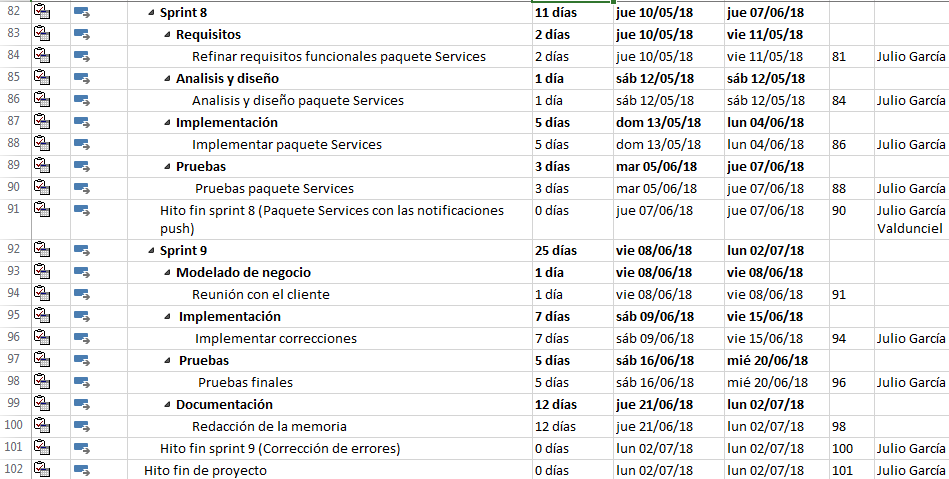
Ilustración - Calendario del proyecto

## Planificación de tareas.



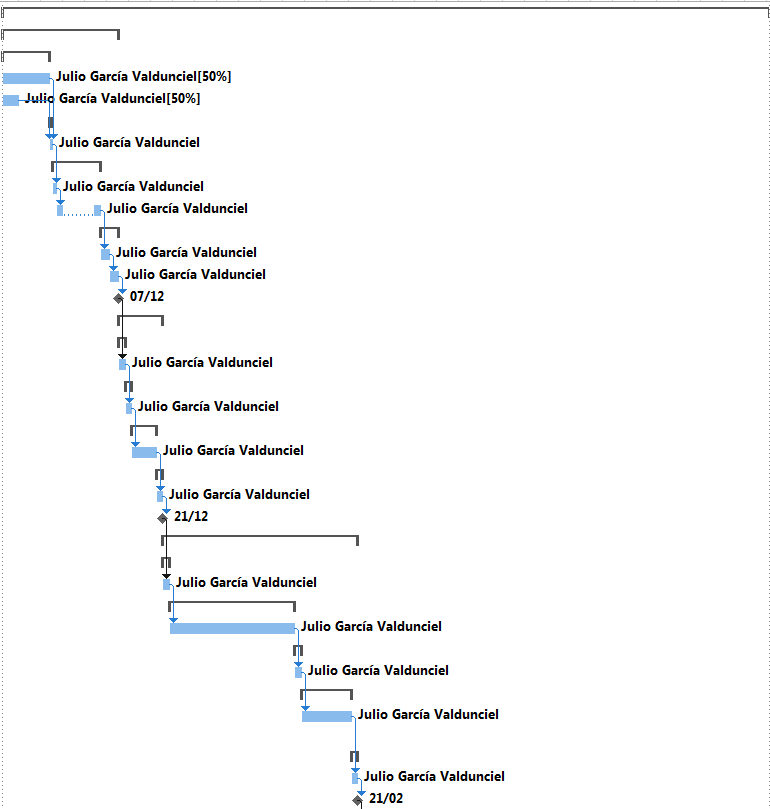


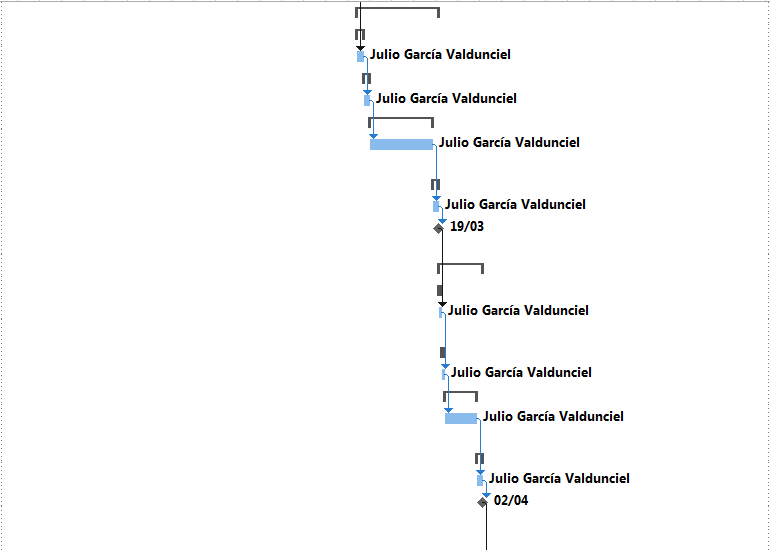


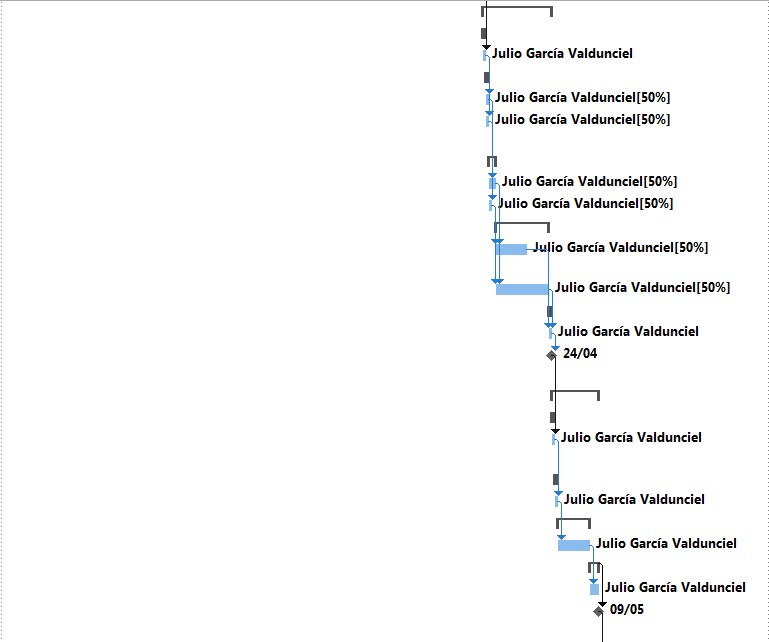


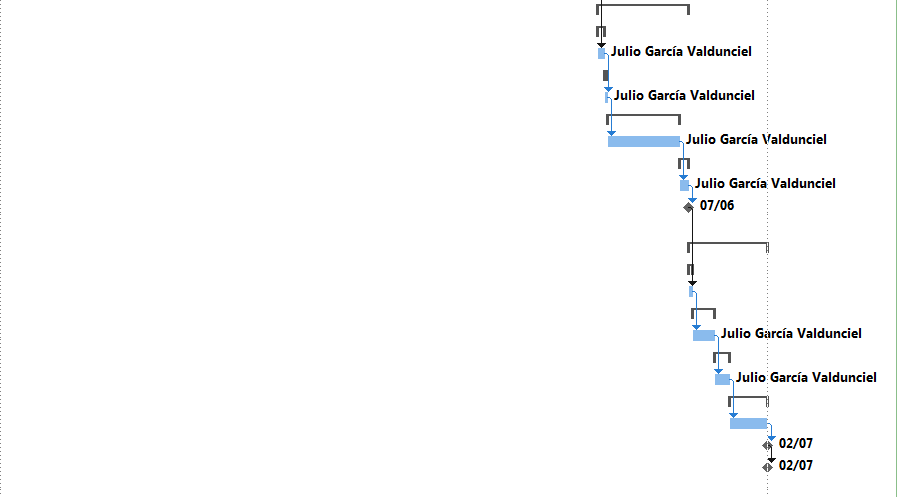
Las primeras iteraciones están orientadas a la parte de definición de requisitos y aprendizaje mientras que las últimas están más enfocadas a la construcción y pulir errores.

## Diagrama de Gant









# Estudio de viabilidad

En este apartado se comentará la viabilidad del proyecto teniendo en cuenta los recursos disponibles para llevarlo a cabo y la viabilidad de la aplicación en el mercado.

## Recursos

En cuanto a recursos se puede considerar un proyecto viable teniendo en cuenta que las herramientas utilizadas para el desarrollo serán gratuitas, el alojamiento de la base de datos se hará utilizando la plataforma Firebase que posee un plan gratuito a pequeña escala y en recursos humanos el único coste es el tiempo empleado por el propio alumno, un coste necesario para el desarrollo del TFG que se puede considerar de coste cero.

## Mercado

En cuanto a mercado se han investigados alternativas similares y lo habitual son aplicaciones que permiten anotar el calendario personal y que no disponen de comunicación con el coordinador aunque si existen.

Se ha tenido contacto con varias personas con experiencia en un entorno laboral regido por turnos que han buscado una aplicación que les facilite ver su propio horario, estar actualizados y poder realizar cambios de turno de forma dinámica. Esta aplicación se ha basado en gran parte en sus necesidades.

Por estas razones se considera que el proyecto es factible.