

# Planificación Y Metodología



Directed by:

- Carlos Lobato
- Pablo Gamarro Lozano
- Julio Mérida Hoyos
- Jesús Martín
- Francisco Jimenez
- José Ramón Casero
- Adrian Sevilla

Change Your City

# **Índice**

<b>1. Metodología</b>	<b>2</b>
<b>2. Descripción del proceso de planificación</b>	<b>3</b>
<b>a. Reuniones llevadas a cabo</b>	<b>3</b>
<b>3. Planificación del proyecto</b>	<b>4</b>
<b>4. Backlog</b>	<b>4</b>
<b>5. Primer Sprint</b>	<b>5</b>
<b>6. Análisis de Riesgos</b>	<b>6</b>
<b>a. Identificación de Riesgos</b>	<b>6</b>

## **Índice Extra**

<b>1. Tabla de Reuniones y puntos Tratados</b>	<b>3</b>
<b>2. Diagrama de Gantt Planificación proyecto</b>	<b>4</b>
<b>3. Diagrama de Gantt Primer Sprint</b>	<b>5</b>
<b>4. Tablas Análisis de Riesgos</b>	<b>6</b>
<b>a. Riesgos y tipo de riesgos</b>	<b>6</b>
<b>b. Análisis de riesgos</b>	<b>7</b>
<b>c. Planificación del riesgo</b>	<b>7</b>
<b>d. Indicadores de riesgo</b>	<b>8</b>

## **Metodología**

Utilizaremos la metodología ágil de gestión de proyectos Scrum. Se realizará como mínimo una reunión por semana, donde se establecerán las tareas y se les serán asignadas a subgrupos en función de disponibilidad de horarios y conocimientos.

Nos hemos decantado por esta metodología debido a que somos un grupo de gran tamaño con unos conocimientos acerca de la gestión e implementación de software un poco prematuro y con Scrum de nuestro lado podremos organizarnos mucho mejor y alcanzar nuestros objetivos más eficazmente, el marcarnos fechas límite para entregas y distintos problemas que puedan surgir es una forma de acelerar el procedimiento, y así el cliente va viendo la evolución de las etapas y podrá evaluar si el producto se muestra como él esperaba.

## **Descripción del proceso de planificación**

Puesto que vamos a utilizar el modelo Scrum, nuestra planificación quedará reflejada en sprints. Cada dos semanas se cerrará un ciclo sprint y se empezará uno nuevo. Cada sprint consta de los siguientes apartados:

- Planificación: se parte del backlog y se escogen los trabajos más prioritarios para realizarse en el sprint actual. Usaremos los diagramas de Gantt para mostrar la planificación del sprint en el tiempo del calendario de actividades.
- Desarrollo
- Revisión: conocer qué se ha hecho, qué no se ha hecho, qué problemas surgieron y cómo se han solucionado.
- Valoración y mejora: identificar mejoras y modificaciones a introducir para optimizar los resultados

Cada planificación incluirá al menos un diagrama de gantt y se detallarán el estado del proyecto y que se espera realizar en la siguiente iteración. Se estimará el tiempo necesario para cada trabajo del sprint en función de una persona. Seguidamente, se asignará a cada trabajo el número de personas que se considere adecuado y se hará otra estimación para dicho número de personas. Se tendrá en cuenta que cuantas más personas trabajen en una tarea menor será el tiempo en finalizarla. Sin embargo, también se requerirá de un tiempo necesario para la cooperación y el acuerdo entre los miembros.

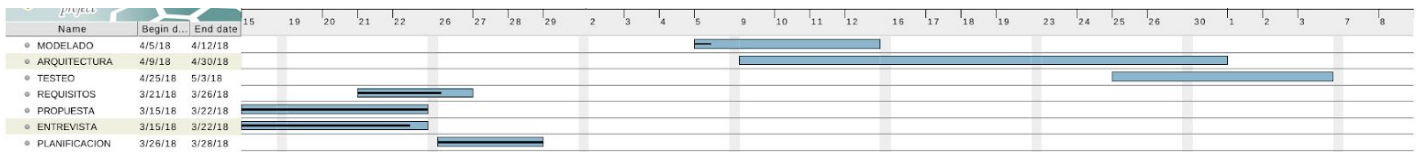
## Reuniones llevadas a cabo y puntos tratados

Fecha	Integrantes	Modo	Actividad
24-3-18	Fran, Jesus, Jose Ramon, Julio	Virtual	Entrevista
24-3-18	Adrian, Pablo	Virtual	Propuesta
4-4-18	Adrian, Julio	Fisica	Requisitos
5-4-18	Fran, Jesús, Carlos, Jose Ramon, Pablo	Fisica	Requisitos, casos de uso, ampliación de entrevista y propuesta, diagrama de gantt
6-4-18	Fran, Adrian, Julio	Fisica	Planificación y metodología.
7-4-18	Carlos, Jesús, Cherra	Virtual	Planificación y metodología. Modelado
16-4-18	José, Adrián, Jesús, Francisco, Julio, Carlos, Pablo	Física	Resolución de <i>issues</i>

## Proceso del reparto de tareas

Nos organizaremos dividiendo las tareas en subgrupos para evitar debates innecesarios producidos por la diversidad de opiniones. Nos disponemos a tratar todas las tecnologías todos los miembros del grupo, dado que queremos aprenderlas y esto también facilitará el diseño y la corrección de errores, por lo que en principio no hay un establecimiento de roles. Simplemente dividiremos las tareas y una vez acabadas intercambiaremos los borradores (es decir, las distintas versiones de los archivos recién finalizadas por un subgrupo) para corregir posibles errores que se hayan pasado por alto, expresiones mal escritas o que podrían derivar en malentendimiento. Para que el reparto de tareas sea equitativo se tendrá en cuenta la memoria de reuniones llevadas a cabo con las tareas ejercidas, mostrada en el punto anterior.

## Planificación general a grandes rasgos del proyecto



TESTEO	4/25/18	5/3/18
REQUISITOS	3/21/18	3/26/18
PROPUESTA	3/15/18	3/22/18
ENTREVISTA	3/15/18	3/22/18
PLANIFICACION	3/26/18	3/28/18

## Backlog

- Como usuario, quiero poder registrarme en la aplicación
  - Estimación 10 días.
- Como usuario, quiero poder iniciar propuestas.
  - Estimación 20 días.
- Como usuario, quiero acceder al conjunto de propuestas de mi zona.
  - Estimación 10 días.
- Como usuario, quiero reportar las propuestas y seleccionar el motivo.
  - Estimación 15 días.
- Como usuario, quiero firmar las propuestas que desee apoyar.
  - Estimación 3 días.
- Como usuario, quiero filtrar las propuestas según la fecha, según la localización o según su clasificación.
  - Estimación 5 días.
- Como usuario, quiero disponer de un perfil.
  - Estimación 10 días.
- Como administrador quiero poder hacer envíos de correos electrónicos a los usuarios registrados.
  - Estimación 2 días.

## Primer Sprint

El primer sprint se va a centrar en desarrollar el sistema de login de la aplicación así como en el sistema de creación de propuestas. Para ello, en primer lugar nos documentaremos acerca del framework Spring, aunque su aprendizaje estará presente durante todo el sprint. Abordaremos las siguientes tareas tomando en base los requisitos ya definidos:

### Estimación para una persona (con nuestros conocimientos actuales).

- Documentación y pruebas de Spring (14 días)
- Desarrollo del login/registro (7 días)
  - Acceso a registro (~1 día)
  - Acceso a inicio de sesión (~1 día)
  - Formulario de registro (< 1 día)
  - Backend de registro (1.5 día)
  - Backend de inicio sesión (< 1 día)
  - Procesar respuesta de login/registro y mandar a frontend (~1 día)
  - Integración y pruebas (~ 2 días)
- Desarrollo de la creación de propuestas (8 días)
  - Documentación y testeo de subida de propuestas (1 día)
  - Formulario de propuesta (< 1 día)
  - Acceso a creación de propuestas (Frontend) (2 días)
  - Almacenamiento de propuestas (Backend) (2 días)
  - Mostrar propuestas (Frontend) (< 1 día)
  - Aplicación de filtros a las propuestas (< 1 día)
  - Integración y pruebas (1 día)

### Estimación para el equipo completo (7 personas)

- Documentación y pruebas de Spring (14 días) todos trabajamos en paralelo compartiendo documentación que consideremos importante.
- Desarrollo del login/registro (5 días) 4 personas
  - Documentación y testeo del login y registrarse
  - API acceso a registro
  - API acceso a inicio de sesión
  - Formulario de registro
  - Backend de registro
  - Backend de inicio sesión
  - Procesar respuesta de login/registro y mandar a frontend
  - Integración y pruebas
- Desarrollo de la creación de propuestas (6 días) 4 personas

- Documentación y testeo de subida de propuestas
- Formulario de propuesta
- Acceso a creación de propuestas (Frontend)
- Almacenamiento de propuestas (Backend)
- Mostrar propuestas (Fronted)
- Aplicación de filtros a las propuestas
- Integración y pruebas

El equipo de login y propuestas tendría en común al menos un programador, para agilizar el entendimiento entre las dos partes, esto se podría dar en todos los tramos de desarrollo de código.



## Análisis de riesgos

En este epígrafe procederemos a dar una relación detallada de los posibles riesgos que se pudieran dar durante el desarrollo de nuestro proyecto y la solución que deberíamos dar en el supuesto de darse cada uno de ellos.

## Identificación de Riesgos

RIESGOS Y TIPO DE RIESGOS	
Tipos de riesgos	Posibles riesgos
Estimación	1. Se subestima el tiempo necesario para la implementación de cada una de las partes del back-end / front-end.
Tecnológicos	1. La base de datos que usemos no será capaz de soportar un gran número de acciones simultáneas. 2. El código de cada programador es caótico lo que produce un mayor tiempo de desarrollo al dificultar la comprensión.
Personal	1. Alguien que tenía unos conocimientos más avanzados sobre una tecnología enferma y no puede asistir. 2. Por problemas personales una persona que tenía el cometido de aprender una tecnología concreta para la posterior implementación no ha podido cumplir con ello.

Requisitos	1. El cliente de nuestro producto pida un cambio en los requisitos en medio de un sprint. 2. Propuesta de nuevos requisitos que hagan muy complejo su implementación.
Organización	1. Un compañero tiene colaborar en otra tarea que no es la suya porque los compañeros no pueden sacarla adelante, dejando así su tarea desorganizada y con retraso.
Herramientas	1. La herramienta que pasará el esquema de la base de datos a código no produce un resultado correcto.

ANÁLISIS DE RIESGO		
Riesgos	Probabilidad	Efectos
Estimación (1)	Alta	Tolerable
Tecnológicos (1)	Muy baja	Tolerable
Tecnológicos (2)	Moderada	Serio
Personal (1)	Baja	Insignificante
Personal (2)	Baja	Catastrófico
Requisitos (1)	Moderado	Serio
Requisitos (2)	Moderado	Serio
Organización (1)	Muy alta	Tolerable
Herramientas (1)	Muy baja	Serio

Planificación del Riesgo	
Riesgos	Estrategia
Estimación (1)	1. Contratar un programador externo para colaborar e implementar dichas partes a tiempo. 2. Aumentar la productividad en otras secciones para dar más prioridad y personal a esta tarea posteriormente.



Tecnológicos (1)	<ol style="list-style-type: none"> <li>1. Buscar una base de datos que soporte el número de transacciones de nuestros usuarios</li> <li>2. Limitar el número de transacciones de los usuarios.</li> </ol>
Tecnológicos (2)	<ol style="list-style-type: none"> <li>1. Instruir a dicho empleado acerca de la importancia de un código limpio y cómo puede influir en el desarrollo y sobre todo en los tiempos de entrega.</li> </ol>
Personal (1)	<ol style="list-style-type: none"> <li>1. Motivar a los empleados a que se documenten mejor sobre esa tecnología y avancen en sus habilidades con ella.</li> <li>2. Contactar con algún programador conocido para que ayude al grupo con su experiencia.</li> <li>3. Si es muy urgente contratar a un programador externo para que supla esta experiencia que no tienen los demás.</li> </ol>
Personal (2)	<ol style="list-style-type: none"> <li>1. Dar una oportunidad al empleado y buscar algún tipo de formación que facilite el aprendizaje de la misma.</li> <li>2. Buscar al empleado que más experiencia tenga en esta tecnología para que pueda abarcar su parte del trabajo.</li> </ol>
Requisitos (1)	<ol style="list-style-type: none"> <li>1. Informar al cliente que cuando estamos en medio de un sprint no podemos reestructurar todo lo que teníamos organizado.</li> <li>2. Comentar al cliente que si no cambia de idea con respecto a esta parte esto supondrá un aumento en el tiempo de desarrollo y además un aumento en el coste estipulado.</li> </ol>
Requisitos (2)	<ol style="list-style-type: none"> <li>1. Incorporar a alguien a la plantilla con experiencia en ese nuevo campo para colaborar a realizarlo, e informar al cliente del aumento del coste de la aplicación por la contratación del mismo.</li> </ol>
Organización (1)	<ol style="list-style-type: none"> <li>1. Concienciar a los compañeros que han tenido que ser ayudados a que hay que arrimar el hombro y ayudar al damnificado en su desarrollo para llegar a tiempo.</li> </ol>
Herramientas (1)	<ol style="list-style-type: none"> <li>1. Probar a realizar la tarea con otra herramienta distinta, y en el caso de obtener el resultado escribir el código acerca de la creación de la base de datos e inserción de contenido a mano.</li> </ol>

Indicadores de riesgo	
Riesgos	Indicador

Estimación (1)	Entrega tardía del código o multitud de informes problemáticos sobre el software.
Tecnológicos (1)	Informes sobre problemas acerca del sobrecalentamiento del hardware hosteador ó notificación sobre demasiado tiempo de espera entre solicitud y solicitud.
Tecnológicos (2)	Incumplimiento de los tiempos estipulados en el back-log y la queja grupal de que el código de un programador específico sea caótico.
Personal (1)	No asistencia al trabajo, recibimiento de mensaje o informe acerca de su estado de salud.
Personal (2)	Disminución de la comunicación del programador con el grupo, incontactable en algún que otro día, desconocimiento del lenguaje técnico de esas tecnologías usadas en el grupo.
Requisitos (1)	Multitud de peticiones del cliente ó insatisfacción en bastantes informes de seguimiento.
Requisitos (2)	
Organización (1)	Incumplimiento de los tiempos estipulados en el back-log y la queja grupal por los retrasos a la hora de entregar código que al final se vuelven inexistentes.
Herramientas (1)	Errores en consultas a la base de datos en la fase de estados.