

Desarrollo de una herramienta web para la gestión de proyectos colaborativos: SGPC

Samuel Torres Fernández

Resumen— El siguiente proyecto abarca la creación de un Software de Gestión de Proyectos Colaborativos (SGPC) enfocados a metodologías ágiles. La idea nace de la experiencia del alumno en su entorno laboral, donde se hace evidente la necesidad de un software de gestión que apoye la metodología de desarrollo establecida. Se trata de un proyecto de carácter personal, cuya finalidad principal es emular las funcionalidades básicas para el funcionamiento de una herramienta de gestión y la adquisición de conocimientos sobre las tecnologías utilizadas por parte del alumno.

Palabras clave—Metodología ágil, TFG, Kanban, Scrum, Trello, Jira, herramienta, framework, front-end, back-end, BBDD, interfaz, mockup, MVC, kernel, ORM, LEMP, HTTP, entidad, persistencia.

Abstract— The following project involves the creation of a collaborative project management software focused on agile methodologies. The idea comes out of the student's experience in their work environment, where exists the need for management software tool that supports the established development methodology. It is a personal Project whose main purpose is to emulate the basic functionalities of a project management tool and the acquisition of knowledge about the technologies used by the student.

Keywords— Agile methodology, TFG, Kanban, Scrum, Trello, Jira, tool, framework, front-end, back-end, DB, interface, mockup, MVC, kernel, ORM, LEMP, HTTP, entity, persistence.



1 INTRODUCCIÓN

Debido al gran flujo de información que se produce durante el proceso de desarrollo de software, ha adquirido cada vez más importancia la elección y el uso de una buena herramienta de gestión de proyectos colaborativos.

La finalidad de este tipo de herramientas radica en facilitar la comunicación y el control de las tareas, así como de los miembros implicados en el desarrollo de un proyecto con el fin de cumplir con los tiempos de desarrollo establecidos.

La motivación de este proyecto nace fruto de la experiencia personal en mi entorno de trabajo, donde la ausencia de una herramienta de gestión, combinada con una mala especificación de la metodología de trabajo, conlleva continuamente retrasos en las entregas.

Es por ello que, aplicando los conocimientos adquiridos de Ingeniería del Software, he decidido desarrollar un proyecto de carácter personal, el cual consiste en la creación de un software de gestión de proyectos colaborativos enfocado a metodologías ágiles, tomando como referencia las herramientas web más comúnmente utilizadas.

Con la finalidad de facilitar la lectura y comprensión del artículo, este se ha desglosado en los siguientes apartados:

1. Introducción
2. Estado del arte
3. Objetivos

4. Metodología
5. Planificación
6. Desarrollo del proyecto
7. Resultados
8. Líneas futuras
9. Conclusiones
10. Agradecimientos
11. Bibliografía

2 ESTADO DEL ARTE

La creación de software es un proceso sistemático sujeto a un gran número de riesgos durante su desarrollo que pueden llevar al fracaso de un proyecto si no se escoge una metodología de trabajo adecuada.

Actualmente las metodologías tradicionales han dado paso a las metodologías ágiles, tales como Kanban[1] o Scrum[2], las cuales solventan muchas de las carencias de las tradicionales. Estas metodologías se apoyan en el uso de herramientas de gestión, las cuales se han convertido en el eje de la correcta coordinación y desarrollo de los tiempos de un proyecto. Para comprender la importancia de las herramientas de gestión se debe distinguir entre los tres conceptos principales que forman parte del desarrollo de un proyecto:

- Proceso: Describe las actividades a realizar para obtener el producto final.
- Método: Indica cómo se van a llevar a cabo las actividades de proceso.
- Herramienta: Proporciona soporte al proceso y los métodos.

- E-mail de contacto: torres.88.bcn@gmail.com
- Mención realizada: Ingeniería del Software
- Trabajo tutorizado por: Lluís Gesa Boté, UAB (DCC)
- Curso 2016/17

Se pueden encontrar gran cantidad de herramientas web para la gestión de proyectos colaborativos de carácter ágil, tanto gratuitos como de pago.

Como línea de investigación para el desarrollo de este proyecto se escogieron dos de los más extendidos en su uso: Trello[3] y Jira[4]. El primero de uso libre y el segundo de pago.

El objetivo de ambas herramientas es el de representar el trabajo como una tarjeta en un tablero o muro de proyecto en el cual los miembros del equipo pueden realizar un seguimiento del progreso de forma visual, agilizando así los procesos. A grandes rasgos estos sistemas se componen de los siguientes elementos:

- Proyectos
- Miembros
- Tareas (Tarjetas)
- Categorías (Listados que engloban tareas)
- Reportes (Gráficas)

Estos elementos tienen sus respectivos atributos y son altamente modificables. Se permite la creación de múltiples proyectos a los cuales se les puede asignar miembros de trabajo con roles distintivos. Estos proyectos permiten la creación de tarjetas o tareas, las cuales se pueden categorizar, temporizar y asignar a miembros del equipo de desarrollo.

La gran diferenciación entre ambas herramientas es el enfoque en la metodología. Mientras que Trello apuesta únicamente por el método Kanban, Jira además de permitir la utilización de ese tipo de tableros, permite la creación de proyectos tipo Scrum.

Pese a compartir muchos de sus conceptos, las dos metodologías tienen enfoques diferentes, los cuales no deben confundirse.

TABLA 1: DIFERENCIAS CLAVE ENTRE KANBAN Y SCRUM

	Kanban	Scrum
Cadencia	Flujo continuo	Sprints de longitud fija periódicos
Publicaciones	Entrega continua	Al final de cada sprint
Métricas clave	Tiempo de ciclo	Velocidad
Cambios	Cambios posibles en cualquier momento	Se debe evitar los cambios durante un sprint

Cabe destacar que, pese a que las dos herramientas ofrecen gran cantidad de funcionalidades, Jira destaca por encima de Trello siendo actualmente la herramienta de gestión de software con mayor número de ventas del mercado. Esta herramienta ofrece más de 1000 extensiones que la convierten en la aplicación web de gestión de proyectos más completa hasta la fecha.

Quizá su única desventaja frente a Trello es que requiere de un plan de pago mensual para su utilización, y

tal vez la complejidad de la interfaz, puesto que ofreciendo tantas funcionalidades la curva de aprendizaje de la herramienta para el usuario es mucho mayor que la de Trello.

3 OBJETIVOS

El objetivo principal de este proyecto es desarrollar una herramienta de gestión de proyectos colaborativos enfocada a metodologías ágiles, así como aplicar los conocimientos adquiridos de ingeniería del software, sirviéndome al mismo tiempo de aprendizaje personal en el uso de las tecnologías seleccionadas.

Tras el estudio previo del estado del arte, se tomó la decisión de enfocar las funcionalidades del proyecto a la metodología Kanban, con la posibilidad de incluir funcionalidades de Scrum como líneas futuras de desarrollo, consiguiendo así un software capaz de realizar gestiones de ambos tipos de metodologías.

Para lograr el objetivo general del proyecto, se estimó que este debía cumplir como mínimo los siguientes objetivos específicos:

1. Permitir el registro e inicio de sesión a usuarios
2. Soportar la creación de múltiples proyectos y sus respectivos atributos
3. Permitir el control de los miembros de un proyecto
4. Permitir la creación y seguimiento de tareas, así como de sus atributos
5. Permitir el control de los miembros de una tarea
6. Permitir la categorización de tareas dentro de un proyecto
7. Permitir la creación de reportes de un proyecto
8. Soportar la persistencia de datos

Inicialmente el proyecto no tiene como objetivo salir al mercado ni ser competidor de las herramientas que han servido de referencia para el desarrollo de las funcionalidades del mismo. Únicamente trata de cubrir los aspectos básicos necesarios para la gestión de un proyecto grupal, permitiendo al mismo tiempo familiarizarme con las tecnologías que utilizo en mi entorno laboral y acentuando la importancia de un buen seguimiento de la metodología de desarrollo de proyectos.

De todos modos, no se descarta como línea futura la opción de llegar a publicar el proyecto en el momento que se consigan las mejoras suficientes como para considerar dicha posibilidad.

4 METODOLOGÍA

Para la elección de la metodología de trabajo se ha tenido en cuenta que se trata de un proyecto individual y que los requisitos recogidos inicialmente pueden estar sujetos a variaciones durante el desarrollo del proyecto. Por lo tanto, se decidió que el método de desarrollo debía alejarse de los ciclos de vida tradicionales tales como el modelo secuencial, utilizados para proyectos de gran envergadura con requisitos muy bien definidos.

Ya que el proyecto trata de desarrollar una herramienta web de gestión de proyectos con carácter de metodología ágil y que creo firmemente en la importancia de este tipo de metodologías, para el desarrollo del TFG se ha utilizado una variación del método Scrum.

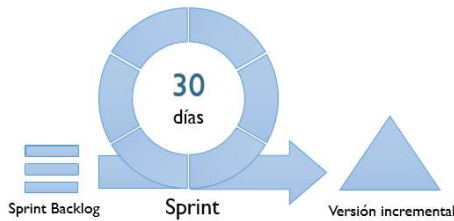


Fig. 1. Diagrama de proceso SCRUM

Emulando dicha metodología, el alumno ha asumido todos los roles de un equipo de desarrollo y ha decidido, bajo la supervisión del tutor, qué tareas debía realizar durante cada sprint.

El orden de desarrollo de las tareas se ha elegido acorde con la planificación temporal y la dificultad estimada por el alumno para la realización de las mismas, respetando sus dependencias temporales.

Durante las fases iniciales se decidió establecer una duración por sprint mensual, ya que es el intervalo de tiempo que transcurre entre cada entrevista obligatoria entre el alumno y el tutor del proyecto. El objetivo de estas reuniones ha sido realizar una revisión de los avances realizados y establecer las siguientes tareas a realizar por parte del alumno.

El tutor y el alumno han mantenido una comunicación activa mediante mensajería móvil y correo electrónico durante todo el transcurso del proyecto, realizando revisiones continuas de los documentos generados del proyecto y ayudando así a iterar las distintas versiones antes de cada entrega.

Tanto la documentación como el código Fuente de la aplicación se encuentran disponibles en un repositorio de GIT al cual tienen acceso tanto el alumno como el tutor. Toda esta documentación está debidamente diferenciada mediante la nomenclatura indicada en el informe inicial del proyecto.

Finalmente, se ha decidido mantener el repositorio público para ofrecer la posibilidad al alumno, una vez terminado el proyecto, de utilizarlo como referencia en su currículo como una muestra de conocimientos adquiridos.

5 PLANIFICACIÓN

Para la realización de la planificación del proyecto y su desarrollo se ha diferenciado entre las tres fases que suelen marcar todo proceso de desarrollo de software:

1. Análisis de requisitos
2. Diseño
3. Codificación

Paralelamente a las dos primeras etapas del proyecto se decidió realizar una investigación de tecnologías y desarrollo de un prototipo inicial de la herramienta web.

Dicho prototipo ha servido para establecer las bases sobre las cuales se construye la aplicación actual y ha permitido sobrepasar los objetivos establecidos en el informe inicial del proyecto.

5.1 Planificación temporal

La decisión de emplear una metodología de trabajo ágil en el desarrollo del proyecto era incompatible con la realización de una planificación estricta de las tareas del mismo.

Aun así, en base a los objetivos definidos y las fechas de entrega se definieron los hitos más importantes del proyecto, así como el listado de tareas generales necesarias a realizar para el éxito del proyecto. Dichas tareas se estimaron temporalmente y ayudaron a la generación de un diagrama temporal de Gantt.

Nombre de tarea	Comienzo	Fin
Reunión inicial con el tutor	vie 16/09/16	vie 16/09/16
Generación informe inicial	lun 19/09/16	dom 02/10/16
Entrega informe inicial	dom 02/10/16	dom 02/10/16
Investigación y prototipado inicial	dom 02/10/16	jue 03/11/16
Análisis de requisitos	vie 16/09/16	jue 06/10/16
Generación Documento de Requisitos	lun 19/09/16	sáb 15/10/16
Diseño	vie 07/10/16	jue 27/10/16
Generación Documento de Diseño	lun 17/10/16	vie 11/11/16
Codificación	vie 04/11/16	jue 26/01/17
Generación informe de progreso 1	mar 18/10/16	dom 06/11/16
Entrega informe de progreso 1	dom 06/11/16	dom 06/11/16
Generación informe de progreso 2	mar 29/11/16	dom 18/12/16
Entrega informe de progreso 2	dom 18/12/16	dom 18/12/16
Generación propuesta Informe Final	mié 28/12/16	dom 22/01/17
Entrega Propuesta Informe Final	dom 22/01/17	dom 22/01/17
Generación de presentación	mar 31/01/17	dom 05/02/17
Entrega Propuesta presentación	dom 05/02/17	dom 05/02/17
Entrega final	mar 07/02/17	mar 07/02/17
Generación poster	lun 06/02/17	dom 12/02/17
Entrega Poster	dom 12/02/17	dom 12/02/17
Preparación defensa del proyecto	sáb 11/02/17	mar 14/02/17
Defensa del proyecto	vie 17/02/17	vie 17/02/17

Fig. 2. Planificación de los hitos del proyecto

5.2 Tecnologías y recursos utilizados

Durante la fase de investigación se seleccionaron las tecnologías a utilizar para el desarrollo del proyecto. También se creó un documento de configuración del entorno de programación donde se especifica cómo configurar todas las herramientas necesarias para poder continuar con el desarrollo del proyecto.

Gran parte de estas herramientas han sido seleccionadas debido a que ya tenía cierto grado de conocimiento de las mismas puesto que utilizo un entorno de desarrollo similar en mi puesto de trabajo.

Las herramientas y tecnologías utilizadas se citan a continuación:

1. Comunicación con el tutor:
 - a. WhatsApp
 - b. Mail institucional UAB
2. Control de versiones
 - a. Repositorio web GitHub
3. Generación de documentación
 - a. Microsoft Office 2016
 - b. Microsoft Project
 - c. Herramienta web Cacao
 - d. Herramienta web SequenceDiagrams
4. Entorno de desarrollo
 - a. Sistema Operativo Ubuntu 16.04
 - b. Framework PHP Symfony2
 - c. Framework front-end Bootstrap 3
 - d. Servidor web NGINX
 - e. IDE Netbeans 8
 - f. BBDD relacional MySQL
 - g. Lenguajes: PHP, HTML5, CSS, JS, XML
 - h. Selenium

5.2 Documentos generados

Durante el transcurso del proyecto se ha generado una serie de documentos listados en la *Tabla 2*.

TABLA 2: DOCUMENTACIÓN GENERADA

Nombre	Nomenclatura
Documento de Requisitos	SGPC_RD
Documento de Diseño	SGPC_DD
Documento de Configuración de Entorno	SGPC_DCE
Reportes	SGPC_RPT_[Nº]

6 DESARROLLO DEL PROYECTO

Gracias al prototipo inicial que se creó durante las primeras fases de desarrollo, el proyecto actualmente se encuentra en un estado de completitud que supera los objetivos iniciales establecidos. Por ese motivo esta sección cubre únicamente el desarrollo principal que trata de cubrir los objetivos iniciales. Para conocer el estado actual de la aplicación se debe consultar la sección 7.

Acorde a la distinción entre las etapas del proyecto realizada en la fase de planificación, el desarrollo se divide en las fases de análisis de requisitos, diseño y codifica-

ción. También se dedica un apartado de esta sección (6.4) a la fase de testeo de funcionalidades implementadas.

6.1 Análisis de requisitos

Los requisitos específicos del sistema se definieron en la primera fase del proyecto acorde a los objetivos que debe cumplir el mismo y están categorizados en función de las clases o entidades del sistema a las cuales pertenecen. Estos se encuentran recogidos en el Documento de Requisitos del proyecto. Algunos de los más generales son citados a continuación.

6.1.1 Requisitos funcionales

RF.01.01 El Sistema debe permitir el concepto de usuario. Cada usuario debe ser único, identificado por un nombre y correo electrónico únicos.

RF.03.01 El Sistema debe permitir el concepto de proyecto. Cada proyecto tiene propiedades tales como miembros del mismo, categorías, tareas y reportes. Este será único en el Sistema y tendrá un único usuario creador.

RF.04.01 El sistema debe permitir el concepto de tablón de proyectos. Este debe contener una vista y acceso a todos los proyectos del usuario en cuestión.

RF.05.01 El sistema debe permitir el concepto de Categoría. Se trata de listados vinculados a un proyecto padre, que a su vez contienen tareas.

RF.06.01 El sistema debe permitir el concepto de Tarea. Cada tarea estará vinculada a una categoría padre y esta podrá tener asignada miembros del proyecto. La tarea dispone de atributos tales como el nombre, estado actual, descripción o miembros.

RF.06.05 El sistema debe permitir la creación de tareas que consuman tiempo para realizar así su seguimiento.

RF.07.01 El sistema debe permitir el concepto de reportes. Cada reporte estará vinculado a un proyecto y mostrará información visual acerca del mismo.

6.1.2 Requisitos no funcionales

RNF.01 El sistema debe estar accesible siempre que se disponga de acceso a internet y el sistema se encuentre activo.

RNF.02 El sistema debe mantener la integridad de los datos. Para ello serán debidamente almacenados y codificados en caso de ser datos sensibles de usuario.

RNF.03 El sistema debe ser accesible independientemente del navegador web utilizado para acceder al mismo.

RNF.04 El sistema debe funcionar sin producir errores en su flujo de funcionamiento.

RNF.05 El software debe permitir una interacción intuitiva y eficiente con el usuario, todo ello utilizando una interfaz adecuada y un correcto diseño de software.

6.2 Diseño

Durante la fase de diseño se ha definido el sistema detalladamente construyendo así una representación que ha servido de referencia para el desarrollo de la fase de codificación del proyecto.

6.2.1 Arquitectura del sistema

La arquitectura del sistema define los elementos estructurales principales del software, así como las relaciones existentes entre los mismos. En esta sección se ofrece una visión general de los niveles que forman parte de la estructura de la herramienta web que se ha desarrollado.

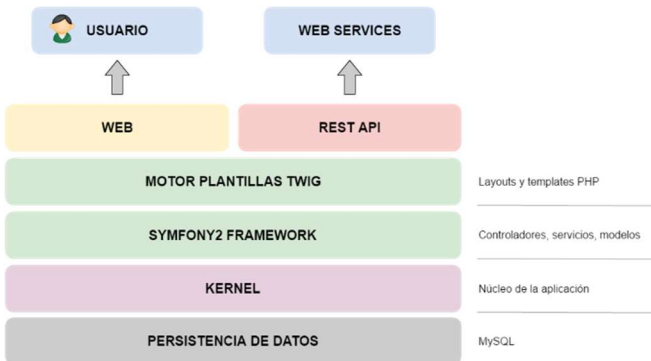


Fig. 4. Diagrama de arquitectura del sistema

La interacción entre el cliente (navegador utilizado por el usuario) y el servidor de la aplicación se realiza mediante el protocolo HTTP. Es por ello que se utiliza el estilo de arquitectura REST.

El framework Symfony2[5], usado para el desarrollo, provee una alternativa de PHP que facilita la interacción entre las peticiones y respuestas HTTP, ofreciendo cierto nivel de abstracción en las operaciones más comunes. El flujo de las peticiones/respuestas realizado con Symfony2 se puede resumir en los siguientes pasos:

1. El cliente envía una petición HTTP
2. Cada petición ejecuta el "Front Controller" del framework, el cual pasa la información de la solicitud.
3. Internamente, Symfony utiliza las rutas y los controladores definidos por el programador para crear la respuesta de la página.
4. Symfony transforma la respuesta en contenido HTTP el cual se envía de vuelta al cliente.

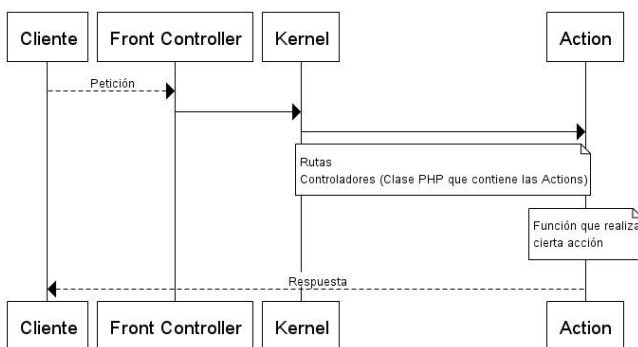


Fig. 4. Flujo de peticiones/respuestas con Symfony2

Todos los elementos encargados del funcionamiento de dicho flujo, desde las entidades hasta los controladores

del núcleo de la aplicación, forman lo que se denomina el back-end del sistema. Este será el encargado de enviar la información necesaria a la vista del mismo.

La vista del sistema o front-end es la que ofrece la interacción con el usuario renderizando todos los objetos devueltos por el servidor de la aplicación.

Para el desarrollo de esta parte se utilizó el framework Bootstrap 3, el cual es el más popular en el desarrollo web responsivo con HTML, CSS y JS.

También se contó con la ayuda del motor de plantillas Twig, incluido en el framework Symfony, que ayuda a simplificar la lógica de los templates de la aplicación, ofreciendo extensiones de gran ayuda que facilitan el uso de PHP.

6.2.2 Infraestructura del sistema

Debido a la naturaleza de la herramienta, se ha decidido establecer una infraestructura LEMP[8] para el desarrollo del proyecto. Las siglas LEMP hacen referencia a los componentes que la forman (Linux, Nginx, MySQL, PHP).

- **Sistema Operativo:** El S.O. elegido para el desarrollo del proyecto es Linux, concretamente la distribución 16.04. Esta elección se debe a su buen rendimiento y fiabilidad en el desarrollo web.
- **Servidor web:** El servidor web es el encargado de contener todas las páginas de nuestra herramienta. Se ha utilizado Nginx puesto que ofrece mejor rendimiento que Apache.
- **Base de datos:** Aquí residen todos los datos del sistema, es decir todas las tablas (entidades, registros relaciones y atributos) de la aplicación. El motor de la base de datos escogido es MySQL.
- **Lenguaje que construye la aplicación:** El núcleo de la aplicación está construido en PHP, apoyado por el framework Symfony2, el cual facilita su aplicación utilizando el paradigma del MVC.

6.2.3 Diseño de la base de datos

En función de los objetivos del proyecto y tras el análisis de las herramientas que han servido de referencia para el desarrollo de este proyecto, se determinaron una serie de entidades que debía tener como mínimo la aplicación.

Se realizó un primer diseño de la base de datos mediante un diagrama E-R que fue incluido en la generación del Documento de Diseño del proyecto. Las entidades principales que inicialmente se estimó que debían formar la base de datos son las siguientes:

- **User:** La entidad de usuario tiene una relación "Many to Many" tanto con los proyectos como con las tareas, almacenados respectivamente en las tablas project_user y task_user gracias a sus claves primarias (ID).
- **Project:** Un proyecto tiene asignado un creador

(owner_id) en el momento de su creación. Tiene además un nombre, una descripción y un identificador único que sirve para relacionarlo con las demás entidades. Esta entidad construye una relación “One-to-many” con las categorías, internamente llamadas “listings”.

- **Listing:** Las categorías o “listings” tienen un proyecto padre asignado en el momento de su creación. Disponen de una relación “One-to-many” con las tareas del proyecto, ya que una categoría está formada de agrupaciones de tareas.
- **Task:** La entidad tarea tiene una relación con su listado padre (listing_id) y con uno o varios usuarios mediante la tabla task_user.

El estado actual de la base de datos se puede consultar en la sección 7. El diagrama E-R está disponible en el anexo A1 de este artículo.

6.2.1 Diseño de la interfaz

El diseño de interfaces tiene como propósito facilitar la utilización del sistema a los usuarios, cubriendo las entradas y salidas de información entre ambos. La interfaz debe cumplir con los requisitos no funcionales definidos en el Documento de Requisitos del proyecto.

Una vez recogidos los requisitos pertinentes se realizó el diseño de la vista de usuario de la herramienta web. Puesto que dispongo de cierta experiencia en diseño de interfaces de usuario y tenía bastante claro que quería una aplicación con una apariencia similar a la de Trello por su simplicidad y usabilidad, la generación de mockups fue el primer paso realizado durante la fase de diseño.

Los mockups se realizaron con la herramienta web “Cacoo” y sirvieron para plasmar la información necesaria y acciones básicas que debía ofrecer la vista del sistema al usuario. Estos, también sirvieron de base para la posterior creación de maquetas y diseño aplicando lenguaje CSS.

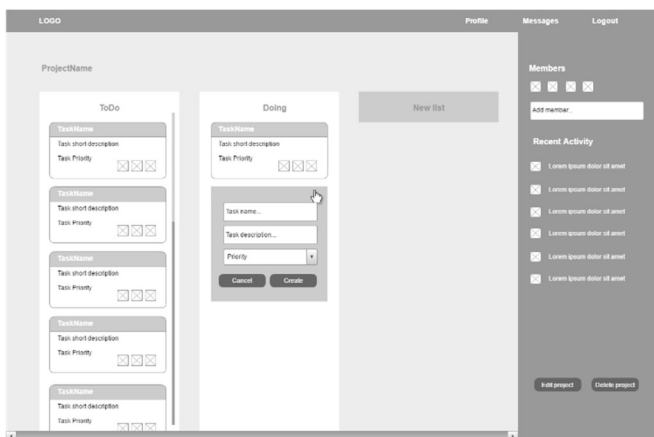


Fig. 3. Ejemplo de mockup de interfaz de usuario

6.3 Codificación

Al inicio de la fase de codificación se decidió disminuir la

duración de los sprints de un mes a dos semanas. Esta decisión se tomó debido a que cada funcionalidad a implementar está formada por diversos cambios en el código y se necesitaba gestionar cuidadosamente las subidas de versiones al repositorio del proyecto. Este control de versiones resultaba más fácil si los sprints eran más reducidos.

Para la codificación de la herramienta web trabajando con Symfony se debía escoger un orden o flujo de trabajo específico. En este caso se ha elegido el orden lógico especificado en la documentación del framework:

1. **Creación de entidad:** Las entidades son clases de PHP que corresponden a cada objeto o concepto de nuestra base de datos.
2. **Mapeo de la entidad:** Symfony trabaja con ORM, lo cual permite mediante anotaciones en las entidades establecer las relaciones de la base de datos.
3. **Persistencia de la entidad:** La consola de comandos de Symfony2 permite generar y persistir la entidad en la base de datos correspondiente a la clase y el mapeo generado.
4. **Creación de rutas:** Las rutas de Symfony relacionan las acciones de los controladores con la URL de la vista correspondiente a modo de identificador.
5. **Creación del controlador de la entidad:** Cada entidad debe tener un controlador. Este es una clase que contiene funciones PHP para las vistas que se encargan de leer las peticiones y devolver las respuestas.
6. **Creación de la vista:** La vista está formada por templates HTML/CSS. Estos renderizan la información enviada por los controladores para que el usuario pueda interactuar con el servidor de la aplicación.
7. **Pruebas:** Cada vez que se termina el desarrollo de una funcionalidad se debe testear la misma, así como las que tengan relación con ella. La siguiente sección detalla el testing realizado.

6.4 Test

El testeo es imprescindible a la hora de validar que una funcionalidad implementada está exenta de bugs, o no ha creado conflicto con otras funcionalidades. Para comprobar esto se ha recurrido al uso de tests unitarios y funcionales.

Cabe destacar que además de la utilización de tests, se ha recurrido al uso de “dumpers” de PHP durante el desarrollo de todas las funcionalidades, comprobando siempre por la consola del navegador que los datos de la respuesta del servidor eran los esperados por la petición y en el formato correspondiente (array,string, etc).

6.4.1 Test unitarios

Los test unitarios en Symfony2 consisten en probar una clase PHP específica. Para crear un test que pruebe una clase, se debe crear un archivo con el mismo nombre añadiendo el sufijo "Test".

El contenido de los tests debe ser la creación de una instancia de la entidad seguido del uso de asserts que comprueben si se cumple o no el comportamiento esperado de dicha entidad.

6.4.2 Tests funcionales

El uso de pruebas funcionales para web puede resultar muy tedioso. Para este proyecto se ha seleccionado el uso de tests de aceptación de PHP realizados con el software "Selenium".

Dicho software es un entorno de pruebas de software para aplicaciones basadas en web. Esta aplicación te permite "grabar" en el cliente (navegador) el flujo de acciones (peticiones) deseadas para posteriormente reproducirlo pulsando únicamente un botón. Esto permite probar si un flujo de trabajo sigue funcionando del mismo modo tras implementar una funcionalidad nueva en la herramienta.

7 RESULTADOS

Debido a que el proyecto ha avanzado más allá de los objetivos iniciales del mismo, en primer lugar, se analizan los resultados obtenidos respecto a los objetivos iniciales y a continuación se muestran los resultados adicionales.

7.1 Resultados iniciales

Los resultados obtenidos respecto a los objetivos iniciales del proyecto son los siguientes:

TABLA 3: FUNCIONALES PRINCIPALES DESARROLLADAS

Componentes	Funcionalidades desarrolladas	Objetivo cumplido
Seguridad	Registro e inicio de sesión de usuarios	1
Proyectos	Creación y eliminación de proyectos	2
Reportes	Visualización de reportes	7
Tareas	Creación/edición/archivado y seguimiento de tareas, así como su seguimiento	4, 6
Categorización	Creación de categorías y agrupación de tareas en su interior	4
Gestión miembros	Asignación de miembros tanto a nivel de proyectos como de tareas	3, 5
Persistencia de datos	El sistema soporta la persistencia de datos	8

La tabla anterior muestra las funcionalidades implementadas agrupadas por el componente al que pertenecen y referenciadas a su objetivo del proyecto correspondiente.

Se puede observar que los objetivos iniciales se han cumplido satisfactoriamente pese a la curva de aprendizaje de algunas de las tecnologías utilizadas o desconocimiento de implementación de algunas funcionalidades.

Tanto es así, que durante las últimas semanas de codificación del proyecto se decidió comenzar a implementar parte de las líneas futuras, creando una primera versión de gestión de proyectos con metodología Scrum, la cual actualmente se encuentra en desarrollo.

7.2 Resultados adicionales

Para realizar esta nueva fase de desarrollo se debía separar el proyecto conceptualmente en dos partes (proyectos Kanban y proyectos Scrum), lo cual ha conllevado cambios en la base de datos de la aplicación, así como las vistas y los controladores.

A grandes rasgos, las funcionalidades adicionales del proyecto hasta la fecha son las siguientes:

TABLA 4: FUNCIONALES ADICIONALES DESARROLLADAS

Componentes	Funcionalidades adicionales
Proyectos	Separación entre proyectos Kanban y Scrum en el momento de su creación.
Sprint	Creación de Sprints con tiempo determinado por el usuario y posibilidad de asignar tareas al mismo. Posibilidad de visualizar sprints anteriores al activo.
Reportes	Visualización de reportes de sprint con gráficas tipo Burndown.
Tareas	Separación entre tareas tipo kanban y scrum mediante herencia de entidades (POO). Creación de "logs" de trabajo de tareas necesario para la creación de reportes.
Categorización	Limitación de 3 categorías (To Do, Doing, Done) para los proyectos tipo Scrum.
Vista	Separación de los templates según el tipo de proyecto

7.3 Vistas del sistema

Una vez especificadas las funcionalidades implementadas en el sistema, esta sección muestra el resultado de algunas de las vistas de la aplicación, las cuales tendrán la interacción directa con el usuario de la misma.

Se ha intentando en la medida de lo posible respetar una estética similar a la de Trello, debido a su sencillez, pero a su vez aplicando un diseño inspirado en la guía de estilos de "MaterialDesign" de Google.

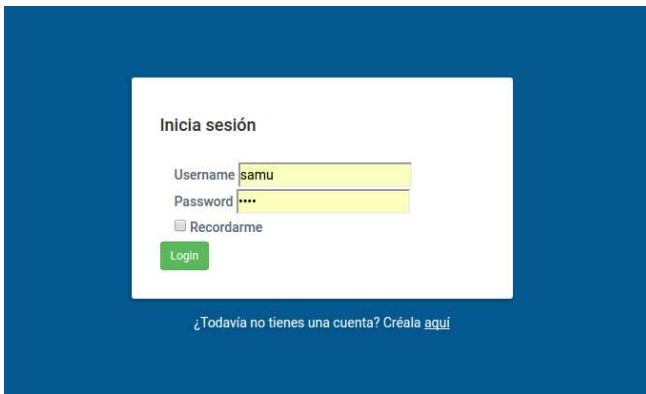


Fig. 6. Vista de login de usuario

Tanto la vista de registro como de inicio de sesión de usuario tienen el mismo aspecto, limpio y funcional para facilitar el acceso del usuario al sistema.

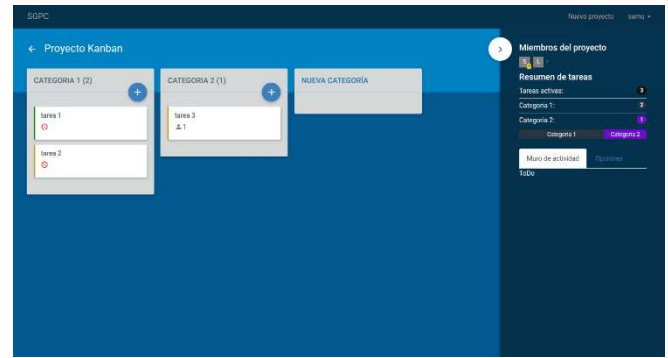


Fig. 9. Proyecto Kanban

La vista de proyecto tipo Kanban nos permite ver un pequeño reporte de las categorías y tareas activas a simple vista, así como crear nuevas y acceder al panel de edición del lateral derecho de la vista.

La creación de categorías no está limitada, así como tampoco la asignación de miembros a tareas.

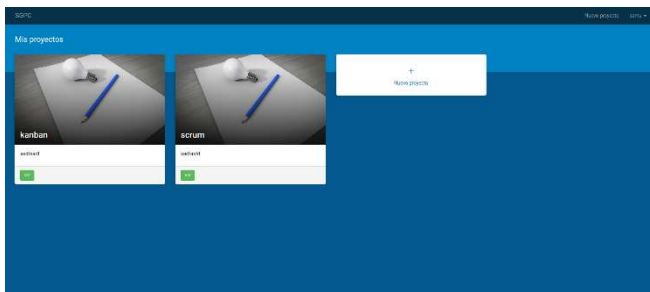


Fig. 7. Tablón de proyectos

La vista de tablón de proyectos es la pantalla principal del usuario visible una vez se inicia sesión en el sistema. Esta pantalla permite visualizar los proyectos vinculados al usuario (tanto creados por él como asignados), así como crear nuevos.

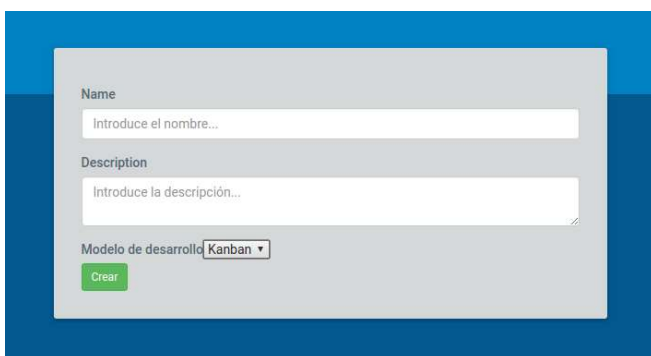


Fig. 8. Creación de proyecto

El formulario de creación de proyectos permite añadirle un nombre, una descripción y seleccionar el modelo de desarrollo que marcará la diferenciación del contenido de dicho proyecto.

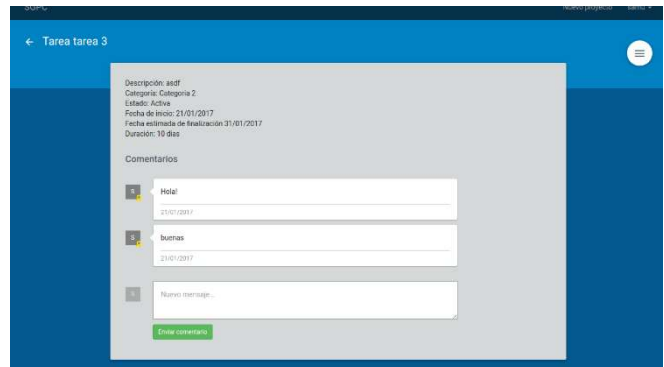


Fig. 10. Vista de tarea

El interior de una tarea permite ver tanto sus atributos como su estado. Además, permite acceder a su edición y realizar comentarios persistentes visibles por todos los miembros del proyecto.

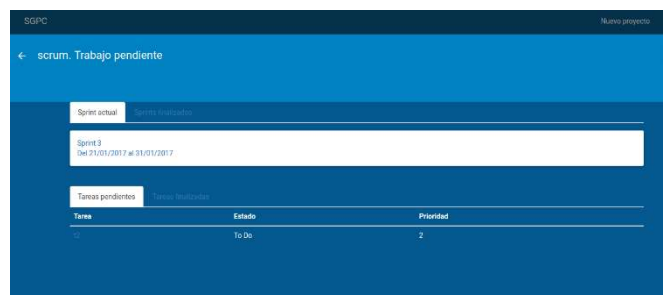


Fig. 11. Proyecto Scrum

La vista de proyecto Scrum nos permite crear tareas que se almacenan en un backlog. La creación de sprints se realiza también en esta vista y se permite añadir tareas a los mismos. Se pueden ver tanto las tareas como los sprints activos y/o finalizados.

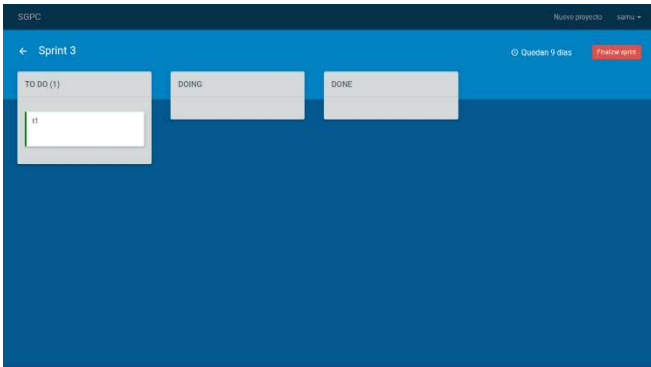


Fig. 12. Sprint

La vista de sprint es similar al muro de Kanban, a diferencia de que solo se disponen de las categorías ToDo, Doing y Done, las cuales no son ampliables ni editables.

El tiempo restante del sprint es mostrado en la cabecera de esta vista, así como la opción de finalizar sprint.

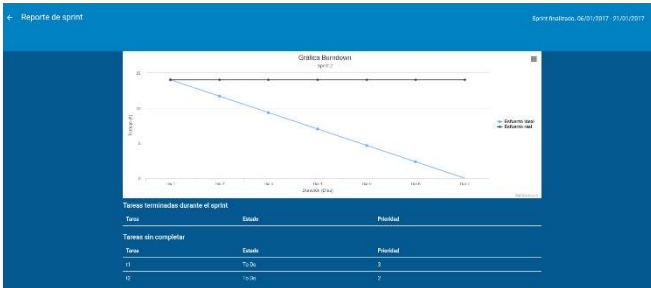


Fig. 13. Reporte burndown

Una vez finalizado el sprint se muestra un reporte mediante una gráfica tipo Burndown. Este tipo de gráficas muestra el trabajo realizado realmente frente al trabajo estimado en el inicio del sprint.

En el ejemplo de la imagen se puede ver una prueba donde no se ha consumido trabajo, la línea oscura no disminuye en el tiempo.

También se muestran las tareas que formaban dicho sprint y el estado en el que han quedado en el momento de finalizarlo.

7.4 Conocimientos aplicados

Uno de los objetivos generales del proyecto se basa en implementar los conocimientos adquiridos de ingeniería del software y el enriquecimiento a nivel personal durante el desarrollo del mismo.

Es precisamente este objetivo el que ha marcado el avance del proyecto desde el inicio del mismo, tanto para la creación de la documentación hasta llegar a la aplicación de la base teórica durante la fase de codificación.

La siguiente tabla muestra un desglose de los conocimientos generales aplicados en las distintas fases de desarrollo:

TABLA 5: CONOCIMIENTOS APLICADOS

Fase	Conocimientos aplicados
Análisis	Delimitación del contexto del sistema Análisis y distinción de requisitos Técnicas de recogida de requisitos Uso de lenguaje natural
Diseño	Diseño de datos Diseño arquitectónico Diseño de interfaces Uso del lenguaje natural
Implementación	Abstracción funcional y de datos Pruebas de software Lenguajes de programación Ser resolutivo

8 LÍNEAS FUTURAS

Pese a que en un principio no se contempla la posibilidad de publicar el proyecto ni llegar a ser digno competidor de las herramientas que han servido de referencia para su evolución, llegados a este punto me gustaría poder seguir desarrollando el mismo, enfocándome en terminar las funcionalidades adicionales de la gestión de proyectos Scrum y mejorar por otra parte la interfaz visual del usuario, implementando elementos como búsquedas, filtros o plugins de drag&drop que permitan la categorización de tareas tan solo arrastrándolas con un click del ratón.

La idea es facilitar más la usabilidad del sistema al usuario y terminar de pulir las funcionalidades ya implementadas, así como añadir otras que se barajaron, pero por falta de tiempo no fueron implementadas, tales como el muro de actividad.

Finalmente, llegado el punto en el que siento que el proyecto se puede dar por finalizado, me gustaría poder mantener el repositorio público y vincularlo en mi currículum para que sirva de referencia de mis conocimientos.

9 CONCLUSIONES

El tiempo y los recursos disponibles son posiblemente los factores más condicionantes a la hora de la correcta realización de un proyecto. Esta premisa sería la conclusión más relevante que se ha podido extraer durante estos meses de trabajo.

Gracias a que desde el inicio del proyecto ya se tenía una idea bastante clara de la herramienta que se quería realizar, no ha resultado demasiado difícil la realización de lo que se asume es la fase más importante del proyecto, la documentación. En ocasiones, contextualizar el sistema que se quiere desarrollar o establecer bien los requisitos y arquitectura del sistema puede suponer un verdadero quebradero de cabeza.

En este caso no se ha tenido que tratar con stakeholders, ya que tanto la idea como el desarrollo del proyecto ha sido propia, con aportaciones del tutor y con las he-

herramientas Trello y Jira como referencia (la primera en mayor medida). Este hecho ha evitado posibles problemas de comunicación entre un posible stakeholder y el ingeniero, con lo cual se ha facilitado el flujo de desarrollo del proyecto durante las fases iniciales.

Los objetivos del proyecto se han cumplido convenientemente en relación a la estimación temporal y además se ha podido comenzar a avanzar en las líneas futuras, por lo que el estado actual de la herramienta ya sobrepasa los objetivos generales del proyecto.

La metodología de trabajo ha sido adecuada ya que no ha sufrido modificaciones importantes ni ha provocado retrasos en los tiempos de desarrollo. Así mismo, los conocimientos en Ingeniería del Software adquiridos en el grado han acompañado todas las fases de desarrollo, habiendo tenido la necesidad de recuperar y revisar temario ya aprendido en años anteriores para poder avanzar con el proyecto.

Actualmente ya se dispone de una herramienta capaz de servir de ayuda en la gestión de proyectos ágiles en pequeños grupos de trabajo, y pese a estar en un estado de desarrollo superior al esperado, todavía se encuentra incompleto en relación a todas las funcionalidades que me gustaría que ofreciese. Es por ello que una vez terminado el periodo del TFG dedicaré parte de mi tiempo a mejorar el proyecto hasta conseguir una versión de la cual me pueda sentir orgulloso.

10 AGRADECIMIENTOS

Antes de concluir con el artículo, me gustaría mencionar a todos aquellos que han supuesto un apoyo durante la realización de este proyecto. Entre ellos a mi tutor Lluís Gesa, por estar siempre disponible cuando me surgía una duda incluso a altas horas de la noche y por exigirme siempre más para que el proyecto sea lo mejor posible. También al jefe de Tecnología de mi empresa, Juan Ramón, el cual me ha ayudado a enfocar ciertos aspectos del proyecto siempre que he acudido a él con alguna duda conceptual en los inicios del mismo.

11 BIBLIOGRAFÍA

- [1] Henrik Kniberg & Mattias Skarin. Kanban y Scrum, obteniendo lo mejor de ambos.
http://www.projectalis.com/documentos/KanbanVsScrum_Castellano_FINAL-printed.pdf
- [2] Ken Schwaber y Jeff Sutherland. La Guía de Scrum.
<https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>
- [3] Web oficial de Trello.
<https://trello.com>
- [4] Web oficial de Jira.
<https://es.atlassian.com/software/jira>
- [5] Symfony Book Version 2.8 (2016).
http://symfony.com/pdf/Symfony_book_2.8.pdf
- [6] A. Estivill y C. Urbano. Cómo citar recursos electrónicos.

- [http://www.ub.edu/biblio/citae-e.htm#\(8\)](http://www.ub.edu/biblio/citae-e.htm#(8))
- [7] Especificación de Requisitos según el estándar de IEEE 830
<http://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>
- [8] Tutorial sobre configuración de entorno LEMP
<https://www.digitalocean.com/community/tutorials/how-to-install-linux-nginx-mysql-php-lemp-stack-in-ubuntu-16-04>
- [9] Software Design Document (SDD) Template
https://www.sovannarith.files.wordpress.com/2012/07/sdd_template.pdf
- [10] Teoría de las asignaturas “Diseny del Software” y “Enginyeria del Software”.
<http://cv.uab.cat>
- [11] Ken Schwaber y Jeff Sutherland. La Guía de Scrum.
<https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>

ANEXO

A1. DIAGRAMA E-R ACTUAL DE LA BBDD

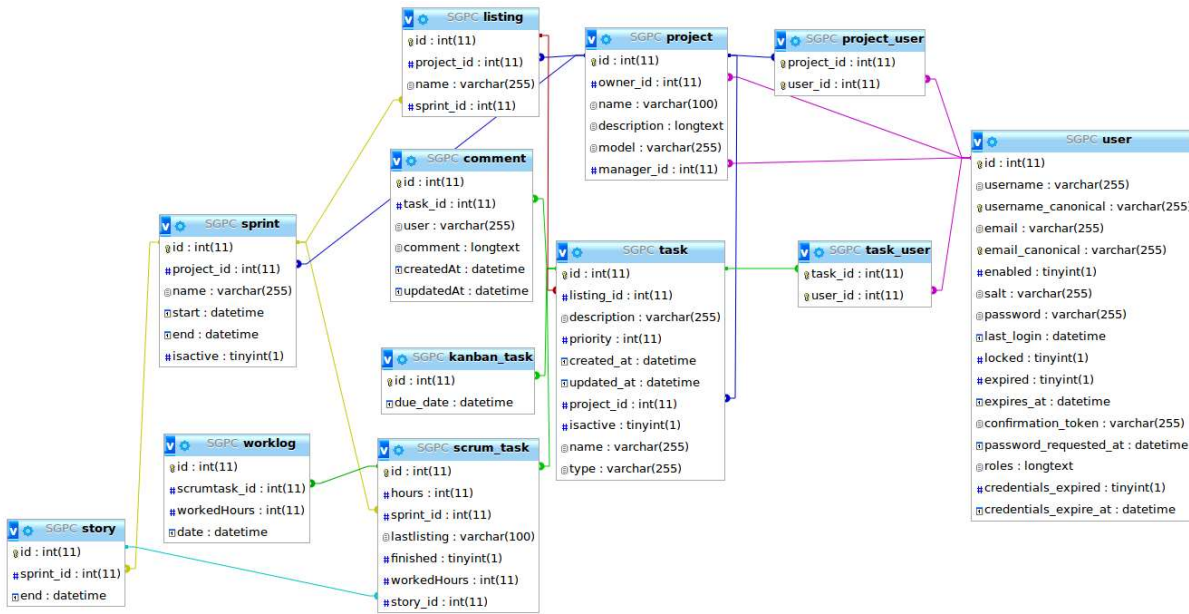


Fig. 14. Diagrama E-R actual de la BBDD

A2. DIAGRAMA DE GANTT DE LOS HITOS DEL PROYECTO

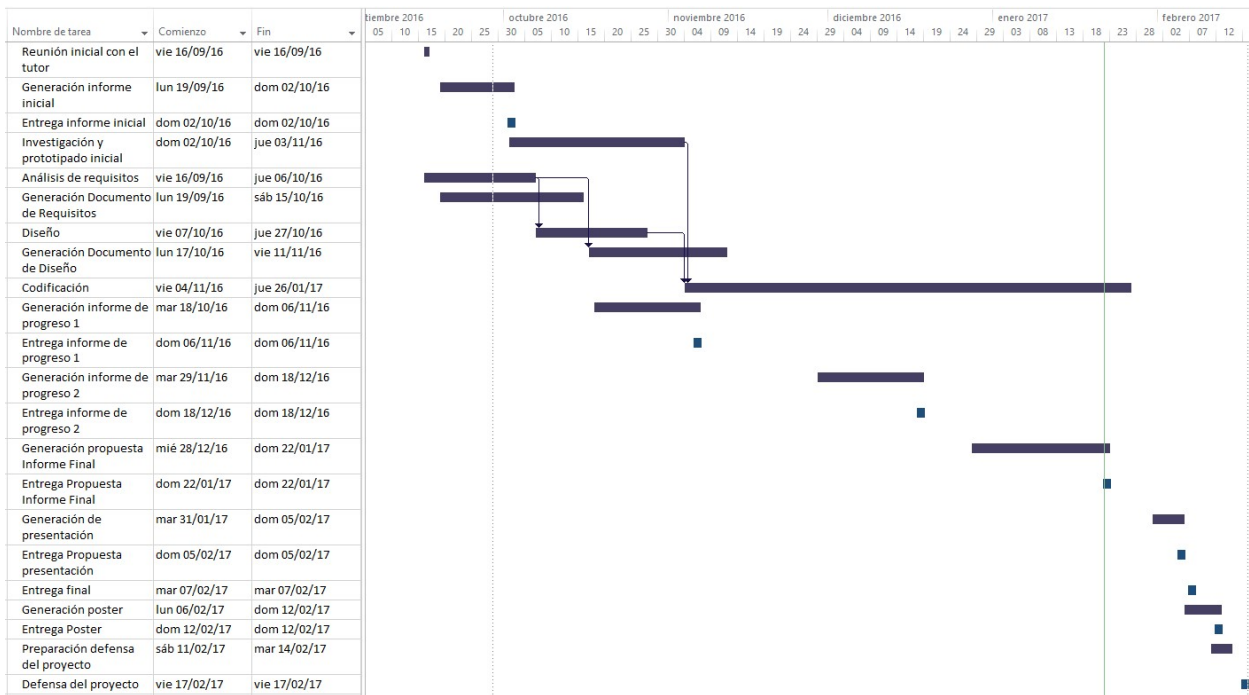


Fig. 15. Diagrama de Gantt