



1-3-2014

# Creación de un gestor de proyectos web

Memoria

Universitat Politècnica de Catalunya (UPC)

Facultat d'informàtica de Barcelona (FIB)

Grado en Ingeniería Informática

Ingeniería del Software



Autor:  
Pablo Pascual Vázquez

Director:  
Carles Farré Tost



*Me gustaría agradecer a Carles Farré Tost su ayuda, supervisión y disponibilidad en el desarrollo de todo el proyecto.*

*También me gustaría agradecer el soporte de mis amigos y familia tanto en el proyecto como durante toda la carrera.*

*Espero que todos estén igual de satisfechos que yo con el resultado del proyecto.*

## Resumen

Durante mis años de universidad me he encontrado muchos trabajos o proyectos en grupo para los cuales una herramienta de gestión de proyectos me hubiera sido muy útil, sin embargo, las herramientas existentes requerían de unos previos conocimientos que no teníamos y otras eran de pago, lo que llevo al uso de otras herramientas que poco tenían que ver con la gestión de proyectos para cubrir esa necesidad. Estos acontecimientos me llevaron a pensar en crear un gestor de proyectos en el cual todos los estudiantes, en este caso, de un mismo proyecto pudieran estar al día.

En esta memoria se puede ver como a partir de la idea principal se ha estudiado el mercado y analizado para encontrar una solución original.

Posteriormente se puede ver toda la especificación del software, las tecnologías que se utilizaran y por último el lector podrá leer como se ha ido implementando para terminar teniendo un gestor de proyectos funcional.

La aplicación final permite a los usuarios tener proyectos en común y estar todos ellos al día, permite asignar tareas a cada usuario, definir prioridades, asignar roles a los usuarios e incluye las funcionalidades más básicas de todos los gestores de proyectos.

Esta aplicación está pensada para su uso gratuito y para favorecer la eficiencia a la hora de realizar proyectos sobre todo pensando en el ámbito universitario.

## Resum

Durant els meus anys d'universitat m'he trobat molts treballs o projectes en grup per als quals una eina de gestió de projectes m'hagués estat molt útil , però, les eines existents requerien d'uns previs coneixements que no teníem i altres eren de pagament , el que porta a l'ús d'altres eines que poc tenien a veure amb la gestió de projectes per cobrir aquesta necessitat . Aquests esdeveniments em van portar a pensar a crear un gestor de projectes en el qual tots els estudiants , en aquest cas , d'un mateix projecte puguin estar al dia.

En aquesta memòria es pot veure com a partir de la idea principal s'ha estudiat el mercat i analitzat per trobar una solució original.

Posteriorment es pot veure tota l'especificació del programari, les tecnologies que s'utilitzaran i finalment el lector podrà llegir com s'ha anat implementant per acabar tenint un gestor de projectes funcional.

L'aplicació final permet als usuaris tenir projectes en comú i estar tots ells al dia , permet assignar tasques a cada usuari , definir prioritats , assignar rols als usuaris i inclou les funcionalitats més bàsiques de tots els gestors de projectes.

Aquesta aplicació està pensada per al seu ús gratuït i per afavorir l'eficiència a l'hora de realitzar projectes sobretot pensant en l'àmbit universitari.

## Abstract

During my college years I found many works or group projects for which a management tool project would have been useful, however, existing tools require some prior knowledge that we hadn't and others were payment tools, I lead to the use of other tools that had little to do with project management to meet that need. These events led me to think about creating a project manager in which all students, in this case, are up to date

In this memory can be seen as following the main idea has been studied and analysed the market to find an original solution.

Then you can see all the software specification, technologies that were used and finally the reader can read how having been implemented to end in a functional project manager.

The final application allows users to share projects and be all day , allowing each user to assign tasks, set priorities, assign roles to users and includes the most basic features of all project managers.

This application is intended for free use and to promote efficiency in performing projects especially considering the university field.

## Índice

Resumen .....	I
Resum .....	II
Abstract .....	III
1 Introducción .....	8
2 Contexto .....	9
2.1 Partes interesadas.....	9
2.1.1 Usuario.....	9
2.1.2 Anunciantes .....	9
2.1.3 Autor.....	9
2.2 Estado del arte .....	10
2.2.1 Definición de gestor de proyectos.....	10
2.2.2 Roles en un proyecto .....	10
2.2.3 Características principales de los gestores de proyectos actuales .....	10
2.2.4 Ventajas e inconvenientes de los gestores de proyectos .....	11
2.2.5 Gestores de proyectos web actuales .....	12
2.2.6 Proyección futura de los gestores de proyectos .....	13
3 Justificación .....	14
4 Objetivos.....	15
5 Alcance.....	16
5.1 Riesgos .....	16
5.2 Metodología.....	17
5.3 Herramientas de seguimiento .....	18
5.4 Validación.....	19
6 Planificación.....	20

6.1	Recursos.....	20
6.1.1	Recursos humanos.....	20
6.1.2	Recursos materiales.....	20
6.2	Fases del proyecto .....	21
6.2.1	Estructura de la aplicación .....	21
6.2.2	Registro e identificación de usuarios .....	21
6.2.3	Creación y edición de proyectos .....	22
6.2.4	Creación y edición de tareas .....	22
6.2.5	Creación de deadlines e hitos.....	22
6.2.6	Mejoras en la interface.....	23
6.2.7	Conectividad desde Android a la base de datos .....	23
6.2.8	Creación aplicación Android.....	23
6.2.9	Finalización de la aplicación Android .....	23
6.3	Riesgos en la planificación .....	24
6.4	Diagrama de Gantt inicial .....	26
6.5	Diagrama de Gantt final.....	27
7	Presupuesto.....	29
7.1	Recursos humanos.....	29
7.2	Recursos materiales.....	29
7.3	Costes imprevistos.....	30
7.4	Coste total.....	31
7.4.1	Coste del desarrollo del proyecto .....	31
7.4.2	Coste de mantenimiento por año .....	31
7.5	Viabilidad económica.....	33
8	Sostenibilidad .....	34

9	Especificación de requisitos aplicación web y Android.....	35
9.1	Requisitos funcionales .....	35
9.1.1	Diagrama de casos de uso .....	35
9.1.2	Casos de uso .....	37
9.1.3	Modelo conceptual.....	62
9.1.4	Modelo del comportamiento .....	64
9.2	Requisitos no funcionales .....	78
9.2.1	Requisitos de percepción .....	78
9.2.2	Requisitos de capacidad de uso y humanidad .....	79
9.2.3	Requisitos de desempeño .....	82
9.2.4	Requisitos operacionales y ambientales .....	87
9.2.5	Requisitos de preservación y soporte .....	90
9.2.6	Requisitos de seguridad.....	92
9.2.7	Requisitos culturales y políticos .....	96
9.2.8	Requisitos legales .....	96
10	Justificación de las tecnologías escogidas .....	98
11	Diseño arquitectónico aplicación web .....	99
11.1	Patrones de diseño.....	99
11.1.1	Modelo Vista Controlador .....	99
11.1.2	Active Record.....	100
11.2	Capa de presentación.....	101
11.3	Diseño interno.....	107
11.3.1	Modelo de comportamiento .....	107
11.4	Capa de datos.....	114
12	Diseño arquitectónico Android y API .....	116

12.1	Patrones de diseño.....	117
12.1.1	REST .....	117
12.2	Capa de presentación.....	118
12.3	Diseño interno.....	120
12.3.1	Modelo de comportamiento .....	121
12.4	Capa de datos.....	123
13	Modelo de despliegue Aplicación web y Android .....	124
14	Implementación y desarrollo.....	125
14.1	Iteración 1: Estructura de la aplicación.....	125
14.1.1	Planificación de la iteración.....	125
14.1.2	Planificación de entregas.....	125
14.1.3	Diseño e implementación.....	126
14.1.4	Testing .....	127
14.2	Iteración 2: Registro e identificación de usuarios.....	127
14.2.1	Planificación de la iteración.....	127
14.2.2	Planificación de entregas.....	128
14.2.3	Diseño e implementación.....	128
14.2.4	Testing .....	129
14.3	Iteración 3: Creación y edición de proyectos.....	129
14.3.1	Planificación de la iteración.....	129
14.3.2	Planificación de entregas.....	130
14.3.3	Diseño e implementación.....	130
14.3.4	Testing .....	130
14.4	Iteración 4: Creación y edición de tareas.....	131
14.4.1	Planificación de la iteración.....	131

14.4.2	Planificación de entregas.....	131
14.4.3	Diseño e implementación.....	132
14.4.4	Testing .....	132
14.5	Iteración 5: Creación de deadlines e hitos.....	133
14.5.1	Planificación de la iteración.....	133
14.5.2	Planificación de entregas.....	133
14.5.3	Diseño e implementación.....	133
14.5.4	Testing .....	134
14.6	Iteración 6: Mejoras en la interface.....	134
14.6.1	Planificación de la iteración.....	134
14.6.2	Planificación de entregas.....	134
14.6.3	Diseño e implementación.....	135
14.6.4	Testing .....	135
14.7	Iteración 7: Conectividad desde Android a la base de datos.....	135
14.7.1	Planificación de la iteración.....	135
14.7.2	Planificación de entregas.....	136
14.7.3	Diseño e implementación.....	137
14.7.4	Testing .....	137
14.8	Iteración 8: Creación aplicación Android .....	138
14.8.1	Planificación de la iteración.....	138
14.8.2	Planificación de entregas.....	138
14.8.3	Diseño e implementación.....	138
14.8.4	Testing .....	139
14.9	Iteración 9: Finalización de la aplicación Android.....	140
14.9.1	Planificación de la iteración.....	140

14.9.2	Planificación de entregas.....	140
14.9.3	Diseño e implementación.....	140
14.9.4	Testing .....	140
15	Aspectos legales .....	141
15.1	Base de datos .....	141
15.2	Cookies .....	141
16	Visión al futuro .....	143
17	Conclusiones.....	145
18	Bibliografía.....	148
19	Glosario.....	151
20	Anexos .....	154
20.1	INFORMACION CORPORATIVA - Política de cookies .....	154
20.2	AVISO LEGAL.....	156
20.3	Código fuente .....	161
20.4	Manual de usuario.....	161

## ÍNDICE DE FIGURAS

FIGURA 1 DIAGRAMA DE GANTT .....	26
FIGURA 2 DIAGRAMA DE GANTT FINAL .....	27
FIGURA 3 GRÁFICO VIABILIDAD .....	33
FIGURA 4 DIAGRAMA DE CASOS DE USO .....	36
FIGURA 5 MODELO CONCEPTUAL .....	63
FIGURA 6 DIAGRAMA FUNCIONAMIENTO PATRÓN MVC [13] .....	99
FIGURA 7 MAPA NAVEGACIÓN APLICACIÓN WEB.....	101
FIGURA 8 PANTALLA PRINCIPAL (SCREENSHOT).....	102
FIGURA 9 PANTALLA ÍNDICE DE PROYECTOS (SCREENSHOT) .....	103
FIGURA 10 PANTALLA PROYECTO (SCREENSHOT) .....	104
FIGURA 11 PANTALLA FORMULARIO PROYECTO (SCREENSHOT).....	105
FIGURA 12 PANTALLA TAREA (SCREENSHOT) .....	106
FIGURA 13 DISEÑO INTERNO APLICACIÓN WEB.....	107
FIGURA 14 CAPA DE DATOS APLICACIÓN WEB .....	115
FIGURA 15 DIAGRAMA ANDROIDCLIENT + API .....	116
FIGURA 16 STORYBOARDS APLICACIÓN ANDROID .....	119
FIGURA 17 DISEÑO INTERNO CLIENTE ANDROID Y API .....	120
FIGURA 18 MODELO DE DESPLIEGUE.....	124

## Índice de tablas

TABLA 1 FORTALEZAS Y DEBILIDADES DE GESTORES DE PROYECTOS ACTUALES .....	12
TABLA 2 COSTES RECURSOS HUMANOS .....	29
TABLA 3 COSTES RECURSOS MATERIALES.....	30
TABLA 4 COSTES IMPREVISTOS .....	31
TABLA 5 COSTES DEL DESARROLLO DEL PROYECTO.....	31
TABLA 6 COSTES MANTENIMIENTO PRIMER AÑO.....	32
TABLA 7 COSTES E INGRESOS ESTIMADOS DE LA APLICACIÓN.....	33
TABLA 8 MÉTODOS API .....	118

## 1 Introducción

Este trabajo es la primera parte del proyecto de final de carrera creación de un gestor de proyectos web. En este proyecto se creará una aplicación web que implementará un gestor de proyectos, también se desarrollará una pequeña aplicación Android para poder conectarse a la aplicación y poder estar al día del proyecto en el que un usuario esté involucrado.

El objetivo es crear un gestor multiplataforma con el fin de que todos los usuarios puedan tener acceso a sus proyectos desde cualquier lugar y en cualquier momento y desde cualquier plataforma ya sea móvil, tablet u ordenador, por otro lado también se quiere conseguir un gestor sencillo y ágil que permita un uso muy práctico y rápido por tal de minimizar la curva de aprendizaje del uso de la aplicación y garantice que los proyectos puedan ser creados de forma rápida y eficiente a la vez.

## 2 Contexto

### 2.1 Partes interesadas

#### 2.1.1 Usuario

Los usuarios serían los principales interesados en el sistema debido a que son los que lo usarán, dentro de ellos podemos encontrar dos tipos de actores:

- **Persona individual:** Un usuario que sin pertenecer a ninguna empresa, como podría ser un estudiante, se registra en nuestro sistema con el fin de gestionar sus proyectos.
- **Empresa:** Se considera empresa a un conjunto de usuarios que pertenecen a la misma compañía y por tanto sus proyectos serán probablemente comunes entre ellos y que usan el sistema para gestionar sus proyectos.

#### 2.1.2 Anunciantes

Los anunciantes serán aquellas empresas o personas que quieran incorporar sus anuncios a la aplicación.

#### 2.1.3 Autor

El autor, que soy yo, ejerzo el rol de desarrollador, analista y tester<sup>1</sup>. Soy la parte más interesada en este proyecto debido a que mi objetivo es realizar todo este trabajo en el tiempo estimado por tal de conseguir mi titulación en el Grado que estoy cursando al mismo tiempo que intento solucionar los problemas actuales en este tipo de servicios con innovación y sostenibilidad.

---

<sup>1</sup> Encargado de hacer las pruebas a un software y valorar su correcto funcionamiento.

## 2.2 Estado del arte

### 2.2.1 Definición de gestor de proyectos

Un gestor de proyectos es una herramienta que trata de facilitar la organización y distribución de un proyecto y que a su vez es capaz de seguir un proyecto y ser capaz de detectar atrasos o problemas en cuanto a tiempo y costes del proyecto.

En sus inicios los gestores de proyectos eran softwares de escritorio, un ejemplo muy conocido es Microsoft Project, que almacenaban los datos del proyecto dentro de servidores propios de la empresa, sin embargo, a día de hoy podemos ver gestores de proyectos online o en la nube lo que elimina el coste para las empresas en cuanto a servidores y facilita el acceso a todos los miembros de un proyecto. [1]

### 2.2.2 Roles en un proyecto

En un proyecto podemos encontrar diferentes roles, ya que no todo el mundo tiene las mismas funciones dentro un mismo proyecto, los siguientes desempeños son una generalización ya que dependiendo del ámbito del trabajo algunos papeles pueden no existir o tener otro nombre.

**Gerente del proyecto:** Es un rol muy importante, es el encargado de gestionar la planificación y el trabajo, asignar tareas de acuerdo a las habilidades de los miembros, controlar constantemente el estado del proyecto y tiene que ser capaz de resolver problemas que puedan surgir.

**Líder de equipo:** Este miembro es el encargado del correcto funcionamiento de su equipo y el representante de este, su función es saber en detalle cuales son las funciones de su equipo y cuáles son las tareas a realizar, para ellos ha de tener grande habilidades de trabajo en equipo y ser capaz de solucionar problemas internos en su equipo.

**Miembros del equipo:** Los miembros del equipo son todos aquellos que forma parte del proyecto y que tienen que ser asignados a tareas específicas en función de sus habilidades. [2]

### 2.2.3 Características principales de los gestores de proyectos actuales

Los gestores de proyectos actuales son herramientas capaces de:

- Detectar solapamientos entre tareas.
- Ver que miembros del proyecto tienen cargas muy altas de trabajo y cuales tienen cargas muy bajas.
- Crear diagramas de Gantt<sup>2</sup>.
- Analizar el proyecto día a día y poder estimar posibles retrasos o si el proyecto se está llevando al día.
- Para todo ello por supuesto necesitan un feedback<sup>3</sup> constante de los miembros del proyecto.

#### 2.2.4 Ventajas e inconvenientes de los gestores de proyectos

##### 2.2.4.1 Ventajas

- Todos los miembros pueden estar al día del proyecto.
- Encontrar errores de planificación o posibles retrasos antes de que estos ocurran.
- Tener feedback constante de la situación del proyecto.
- Ayuda al gestor de proyectos a la elaboración de una correcta planificación.

##### 2.2.4.2 Inconvenientes

- Si el servidor falla el gestor no puede dar feedback.
- Dependencia de un ordenador.

---

<sup>2</sup> Herramienta gráfica cuyo objetivo es mostrar el tiempo de dedicación previsto a diferentes tareas en un plazo de tiempo total determinado.

<sup>3</sup> Retro alimentación, en nuestro caso el usuario debe ir introduciendo de manera continua nueva información al software.

## 2.2.5 Gestores de proyectos web actuales

En este apartado se analizan los diferentes gestores de proyectos que ya existen en la actualidad, para ello uso una tabla con sus fortalezas y debilidades.

Nombre	Fortalezas	Debilidades
<b>Active Collab [3]</b>	Distribución de permisos y roles. Control de versiones para el caso de programadores. Creación de facturas. Estimaciones de tiempo y costes. Integración con mail.	Es de pago. No tiene integración en plataformas móviles. Su uso no es fácil e intuitivo.
<b>Assembla [4]</b>	Control de problemas. Control de versiones.	Es de pago. No tiene integración en plataformas móvil. Muy orientado a proyectos de software.
<b>BaseCamp [5]</b>	Sencillez. Integración en plataformas móviles.	Es de pago. Su uso no es fácil e intuitivo.
<b>CentralDesktop [6]</b>	Se adapta al tipo de proyecto.	Es de pago. No tiene integración en plataformas móviles.
<b>Atlassian [7]</b>	Usa herramientas para ayuda en el desarrollo. Muy orientado a programadores.	Es de pago. No tiene integración en plataformas móviles.
<b>Kapost [8]</b>	Capaz de hacer análisis del proyecto.	Es de pago. No tiene integración en plataformas móviles.
<b>Producteev [9]</b>	Es gratuito. Tiene integración en plataformas móviles. Tiene API para el desarrollo.	Su uso no es fácil e intuitivo.
<b>Redbooth [10]</b>	Muy orientado a programadores.	Es de pago. Tiene poca integración en plataformas móviles.
<b>TeamLab [11]</b>	Multi-idioma. Capaz de manejar correo.	Es de pago. No tiene integración en plataformas móviles.
<b>Asana [12]</b>	Integración en plataformas móviles. Tiene API para el desarrollo.	Su uso no es fácil e intuitivo.

Tabla 1 Fortalezas y debilidades de gestores de proyectos actuales

### 2.2.6 Proyección futura de los gestores de proyectos

Después de analizar los gestores de proyectos web más usados que hay actualmente, se puede concluir que la proyección futura de estos está fijada en la nube, es decir, se busca que todos los miembros o usuarios de la aplicación puedan acceder, modificar o añadir datos del proyecto en cualquier momento y desde cualquier lugar, para todo esto es necesario crear aplicaciones basadas en “cloud computing<sup>4</sup>”. Por tanto probablemente cada vez más los gestores de proyectos sean capaces de ofrecer nuevos servicios en la nube y busquen añadir funciones más propias de Dropbox <sup>5</sup>o GoogleDocs <sup>6</sup> que de gestores de proyectos.

De otro lado también se puede ver que la integración con las diferentes plataformas existentes actualmente en el mercado, como móviles o tablets, también es un factor muy importante de cara el futuro, cada vez más la gente quiere poder consultar sus datos de interés desde su propio móvil o tablet, debido a esto, es un factor importante para el futuro de los gestores de proyectos una buena integración con todas las plataformas existentes. Uno de los objetivos principales de este proyecto es solventar este problema.

---

<sup>4</sup> Nuevo paradigma que permite ofrecer servicios de computación a través de internet, todo aquello que puede ofrecer un sistema informático se ofrece como servicio de manera que todos los usuarios puedan acceder a él.

<sup>5</sup> Servicio de almacenamiento en la nube, cada usuario almacena sus datos en la nube y puede acceder a ellos desde cualquier lugar en cualquier momento. Para más información: [www.dropbox.com](http://www.dropbox.com)

<sup>6</sup> Servicio de Google usado para crear archivos tipo Word, Excel etc con la capacidad de ser vistos y modificados por varios usuarios al mismo tiempo.

### 3 Justificación

Actualmente las personas y las empresas buscan cada vez más tener todos sus datos en la nube<sup>7</sup> con el fin de ahorrar espacio en sus ordenadores y servidores y por la facilidad de acceder a sus datos desde cualquier lugar en cualquier momento y desde cualquier dispositivo.

Por este motivo y los mencionados en la introducción, he decidido crear un gestor de proyectos web, con el fin de ayudar a las personas o empresas a crear nuevos planes donde todos sus miembros estén siempre al día de la situación del mismo, donde cualquier individuo pueda ver cuál es su rol y sus tareas e intentar de esta manera que ningún usuario de un proyecto se encuentre en una situación de no saber qué hacer.

Este proyecto de final de grado por tanto se centrará en crear una aplicación web que implementará un gestor de proyectos buscando la máxima disponibilidad para todos sus usuarios tanto en tiempo y lugar como desde cualquier dispositivo y buscando también la sencillez y el rápido aprendizaje en el uso de la aplicación.

Supongamos ahora que un grupo de estudiantes universitarios tienen que hacer un trabajo en su universidad, el trabajo probablemente tenga una planificación sencilla comparándolo con una planificación empresarial y seguramente todos ellos dispondrán de dispositivos móviles o incluso tablets. Dada esta situación probablemente los estudiantes no quieran perder mucho tiempo aprendiendo el funcionamiento de un gestor de proyectos cualquiera y sería una gran ventaja poder acceder a los datos de su trabajo desde sus dispositivos móviles, este es el principal objetivo de la aplicación a desarrollar, conseguir que estos estudiantes en poco tiempo tengan una planificación creada y que puedan acceder desde sus móviles en cualquier momento.

---

<sup>7</sup> Representa el concepto de tener documentos o servicios en internet y de manera remota en vez de en un propio ordenador o servidor.

## 4 Objetivos

Del punto anterior podemos sacar por tanto que los objetivos clave de este proyecto son:

- El gestor de proyectos será gratuito para los usuarios, de manera que los ingresos se consigan mediante otras vías como podrían ser la publicidad o cobrando solo por el uso empresarial.
- Tendrá una disponibilidad 24/7<sup>8</sup> y desde cualquier lugar desde el primer momento que sea disponible para el público.
- La aplicación será accesible desde dispositivos móviles y tablets desde el primer momento que sea disponible para el público.
- La aplicación seguirá buenos patrones de diseño en todo momento y por tanto será usable.
- La aplicación será sencilla de usar y su curva de aprendizaje será mínima, un nuevo usuario no debería tardar más de 30 minutos en aprender el funcionamiento completo del software después de leer la ayuda.

---

<sup>8</sup> Las 24 horas del día, los 7 días de la semana.

## 5 Alcance

La aplicación web tiene la finalidad de ser usada tanto por particulares, como podrían ser estudiantes, como por empresas y tiene que ofrecer las funcionalidades necesarias por tal de que sea una aplicación funcional, atractiva y de fácil manejo. Se ha de ofrecer a los usuarios la posibilidad de crear proyectos, estar al día de proyectos en los que son participes y por supuesto tener acceso 24/7 desde cualquier lugar.

### 5.1 Riesgos

En este proyecto se pueden encontrar ciertos riesgos que hay que tener en cuenta de manera continua durante el desarrollo y que pueden afectar al producto final, los riesgos que podemos encontrarnos son los siguientes:

- **Ineficiencia:** Se trata de una aplicación que se usará por muchos usuarios, es necesario que la aplicación sea rápida, eficaz y que el usuario no tenga que malgastar tiempo en ella. Para evitar esto, debo intentar que la ejecución de las partes más costosas en cuanto a tiempo de nuestra aplicación estén programadas de la manera más eficiente posible.
- **Servicio incorrecto:** También habrá que cuidar que los servidores donde este alojada la aplicación puedan dar un servicio correcto frente a la comunidad de usuarios que exista en todo momento, es decir, que los servidores tengan la capacidad suficiente para atender a todas las peticiones de los usuarios y por supuesto sin exceder en el uso de recursos.
- **Tiempo excesivo:** Para finalizar este proyecto se estima de un tiempo límite fijado con anterioridad, por tanto es imprescindible tener el proyecto acabado para una fecha estimada. Para controlar esto, la propia metodología XP ya indica que al final de cada iteración hay que ir viendo las partes obtenidas e ir estimando si el software podrá estar acabado para la fecha prevista, en caso contrario habría que ir modificando la planificación para hacerlo posible.
- **Seguridad:** La aplicación contendrá futuros proyectos tanto de empresas como de personas, son datos sensibles<sup>9</sup> que deberán almacenarse correctamente y

---

<sup>9</sup> Datos sobre una persona o empresa que no desean que sean vistos por otros sin previa autorización.

garantizando siempre que ningún usuario no autorizado pueda acceder a dicha información. Para garantizar esto durante el desarrollo se tendrán en cuenta varios aspectos de seguridad informática.

- **Financiación:** La idea es que la aplicación sea totalmente o al menos parcialmente gratuita como se comenta en el apartado [2 objetivos](#), sin embargo el mantenimiento y el alojamiento tienen unos costes que se han de poder cubrir. No hay manera segura de garantizar la financiación pero sería interesante añadir publicidad desde el primer momento por tal de intentar cubrir esos costes.
- **Legalidad:** La aplicación contiene bases de datos y almacena información sensible de los usuarios, debido a esto, es importante estar al corriente de la legalidad vigente. Para garantizar esto se cumplirá la LOPD <sup>10</sup>(Ley orgánica de protección de datos) por tal de garantizar el correcto funcionamiento en España y posteriormente en caso de usarse en otros países también se mirara sus propias leyes.

## 5.2 Metodología

Para el desarrollo de este proyecto debido a que juego el rol de desarrollador y cliente a la vez, he decidido usar la metodología eXtreme Programming <sup>11</sup>(XP) adaptada a la situación.

En la metodología eXtreme Programming se necesita un equipo que incluya:

- El cliente, quien es el encargado de proveer los requerimientos<sup>12</sup>, establecer las prioridades y guiar el proyecto.
- Los programadores, son los encargados desarrollar el software.
- Los analistas, que son los encargados de ayudar al cliente definiendo requerimientos
- El coach, que es el encargado de mantener el equipo y el rumbo del proyecto.

---

<sup>10</sup> Ley orgánica de protección de datos, contiene todas las leyes que las bases de datos han de cumplir para estar acorde a la ley española.

<sup>11</sup> eXtreme Programming es una metodología de desarrollo orientada a la ingeniería de software, es una metodología ágil y se diferencia principalmente por dar más énfasis a la adaptabilidad que a la previsibilidad.

<sup>12</sup> Se entiende por requerimientos aquello que el software debe ser capaz de hacer.

Sin embargo, tendré todos los roles.

Para esta metodología se han de tener en cuenta dos tipos de planificaciones: la planificación de la entrega y la planificación de la iteración.

**La planificación de la iteración:** Aquí es donde recae la planificación de la construcción del software, en plazos de 2 semanas entregaré una parte útil y usable del software. Durante esta fase presento las características deseadas para las siguientes dos semanas y posteriormente lo descompondré en tareas y estimaré los costes. Por lo tanto, me comprometo a terminar ciertas características basándome en la cantidad de trabajo hecho en la iteración anterior.

**La planificación de la entrega:** Es donde decidiré qué parte del proyecto se debe hacer, cuál será su dificultad y estimara un plan para el proyecto, estos planes de entrega pueden ser un tanto imprecisos debido a que aún no puedo estimar con exactitud a qué velocidad se desarrollará el proyecto.

### 5.3 Herramientas de seguimiento

Gracias a los dos pasos mencionados en el apartado anterior, yo, ejerciendo el rol de cliente, podré ver mucha información y tendré excelente flexibilidad para guiar el proyecto. Cada 2 semanas haré visible el progreso pudiendo evitar términos como “esta al 90%” ya que en este tipo de metodologías el trabajo está completado o no lo está.

Las entregas definidas en este tipo de metodología son dos que serán definidas de la siguiente manera:

En la primera, **al final de cada iteración se entregará software** testeado (probado) y funcionando de forma que yo y el director del proyecto podamos ver el avance del proyecto.

En la segunda, **diariamente se entregaran pequeñas partes del software**, para ver la velocidad de desarrollo en todo momento.

Para llevar a cabo un correcto seguimiento se utilizará Github <sup>13</sup>para acceder al código y las reuniones o bien serán presenciales o bien se usara el email.

---

<sup>13</sup> Servicio en la nube para almacenar código y hacerlo accesible a más usuarios, para más información: [www.github.com](http://www.github.com)

## 5.4 Validación

Al seguir la metodología eXtreme Programming se usarán también parte de sus métodos de validación.

Se practicará el **TDD (desarrollo guiado por pruebas)**, es decir, después de cada pequeña parte del software se harán test para comprobar que esas partes funcionan al 100% y por tanto ya son válidas para la entrega final de cada iteración, por tanto al final de cada iteración se tendrá unas horas dedicadas a testear todo aquello hecho en la misma iteración.

Se usaran **pruebas automatizadas**, las pruebas automatizadas son pruebas definidas tanto por el cliente como por el desarrollador en las que se considera que si son correctamente pasadas el software cumple correctamente con la funcionalidad probada.

Para el desarrollo con eXtreme Programming también es importante seguir **estándares del código** y crear un código que siempre siga las buenas practicas debido a que una buena programación facilita el entendimiento en terceras partes y mejora la capacidad de encontrar un error.

## 6 Planificación

Este proyecto debe contener 450 horas estimadas de dedicación, es un proyecto que no está empezado y la intención es finalizarlo en estas horas y antes del 1 de Julio ya que debe ser defendido antes de esta fecha. Para llevar a cabo estas condiciones he estimado la siguiente planificación.

El proyecto empezará durante el mes de febrero, se dedicaran 50 horas semanales desde entonces intentado repartirlas en fragmentos de 6 horas diarias repartidos a su vez entre 5 días.

Cada iteración se calcula que tendrá 3 horas dedicadas a la planificación de la entrega, 3 horas dedicadas a la planificación de la iteración (se explica en el apartado metodología que se hace en ambas fases), 40 horas dedicadas al diseño y la implementación que podrán quedar repartidas según la necesidad de cada fase y 4 horas para hacer las pruebas del software. Siguiendo estas iteraciones el proyecto estará acabado en 9 iteraciones.\*

**6h planificaciones + 40h diseño e implementación + 4h pruebas = 50 horas.\***

**50 horas \* 9 iteraciones = 450 horas.\***

\*estos tiempos son estimados y pueden estar sujetos a cambios.

### 6.1 Recursos

#### 6.1.1 Recursos humanos

Para el desarrollo de esta aplicación se necesitará principalmente al autor, ya que seré el cliente, desarrollador y tester de la aplicación y por tanto soy el recurso más importante del proyecto junto con el tutor del proyecto que supervisará todo el desarrollo.

#### 6.1.2 Recursos materiales

En cuanto a recursos materiales, nos encontraremos el equipo que utilizaré, el consumo de electricidad y posteriormente a la finalización del proyecto los costes de los

servidores, todos ellos están más detallados en el apartado [7 presupuesto](#). El software que se usará para el desarrollo es gratuito y por tanto no generará costes adicionales.

## 6.2 Fases del proyecto

Debido a la metodología usada aún no se pueden establecer fechas exactas para el inicio o finalización de las fases, sin embargo, como disponemos de 9 iteraciones a continuación se detallará que se hará en cada iteración, cuál será su dificultad y su prioridad.

### 6.2.1 Estructura de la aplicación

<i>En esta fase se creará el esqueleto de la aplicación, a partir de esta fase surgirá la aplicación.</i>	
<i>Requisitos</i>	<i>Ninguno</i>
<i>Dificultad</i>	<i>Media</i>
<i>Prioridad</i>	<i>Alta</i>
<i>Tareas</i>	<p><i>Crear una base de datos con MySql.</i></p> <p><i>Crear el proyecto siguiendo el patrón MVC<sup>14</sup>.</i></p> <p><i>Instalar todo el software necesario para el desarrollo.</i></p>

### 6.2.2 Registro e identificación de usuarios

<i>En esta fase se implementará el registro de usuarios y su identificación en el sistema.</i>	
<i>Requisitos</i>	<i>5.2.1</i>
<i>Dificultad</i>	<i>Alta</i>
<i>Prioridad</i>	<i>Alta</i>
<i>Tareas</i>	<p><i>Crear los modelos de la base de datos necesarios.</i></p> <p><i>Implementar el registro de usuario.</i></p> <p><i>Implementar la identificación de usuarios en la aplicación.</i></p> <p><i>Implementar "cookie<sup>15</sup>" para recordar al usuario.</i></p> <p><i>Implementar servicio de mensajería para dar de alta o recuperar contraseñas.</i></p>

<sup>14</sup> Patrón modelo vista controlador, es un patrón para la programación orientado a la programación de aplicaciones web.

<sup>15</sup> Pequeño archivo almacenado en el ordenador del cliente que contiene información necesaria para el correcto funcionamiento de la aplicación. En este caso contendrá los credenciales del usuario para recordarlo en posteriores entradas a la aplicación.

### 6.2.3 Creación y edición de proyectos

<i>En esta fase se implementará la creación y edición de proyectos en la aplicación.</i>	
<i>Requisitos</i>	5.2.1 5.2.2
<i>Dificultad</i>	Baja
<i>Prioridad</i>	Media
<i>Tareas</i>	<i>Crear los modelos de la base de datos necesarios.</i> <i>Implementar el modelo del proyecto.</i> <i>Implementar la vista del proyecto.</i> <i>Implementar el controlador del proyecto.</i>

### 6.2.4 Creación y edición de tareas

<i>En esta fase se implementará la creación y edición de tareas en la aplicación.</i>	
<i>Requisitos</i>	5.2.1 5.2.2 5.2.3
<i>Dificultad</i>	Media
<i>Prioridad</i>	Media
<i>Tareas</i>	<i>Crear los modelos de la base de datos necesarios.</i> <i>Implementar el modelo de la tarea.</i> <i>Implementar la vista de la tarea.</i> <i>Implementar el controlador de la tarea.</i> <i>Implementar la dependencia entre tareas.</i>

### 6.2.5 Creación de deadlines e hitos

<i>En esta fase se implementará la creación y edición de proyectos en la aplicación.</i>	
<i>Requisitos</i>	<i>Todas las anteriores.</i>
<i>Dificultad</i>	Baja
<i>Prioridad</i>	Baja
<i>Tareas</i>	<i>Crear los modelos de la base de datos necesarios.</i> <i>Implementar las características de hitos<sup>16</sup> y deadline<sup>17</sup>s en la aplicación.</i>

<sup>16</sup> Dentro de un proyecto, un hito se considera un conjunto de tareas cuya finalización forman parte de una parte muy importante del mismo.

<sup>17</sup> Un “deadline” es un término anglosajón que simboliza el tiempo límite, en el caso de los proyectos suele referirse a la fecha de finalización de una tarea o del proyecto en sí.

## 6.2.6 Mejoras en la interface<sup>18</sup>

<i>En esta fase se dará el toque final a la interfaz gráfica, para que sea usable y atractiva para el usuario.</i>	
Requisitos	5.2.1 5.2.2 5.2.3 5.2.4
Dificultad	Media
Prioridad	Media
Tareas	Comprobar si sigue buenas prácticas de usabilidad y si no es así implementarlas. Comprobar que es atractiva para el usuario.

## 6.2.7 Conectividad desde Android a la base de datos

<i>En esta fase se creará la estructura de una aplicación Android y se creará la conectividad a la base de datos de la aplicación.</i>	
Requisitos	5.2.1
Dificultad	Media
Prioridad	Alta
Tareas	Crear la estructura de la aplicación Android. Implementar la conectividad con la base de datos. Implementar la identificación de usuarios desde Android.

## 6.2.8 Creación aplicación Android

<i>En esta fase se implementará la aplicación Android con todas las funcionalidades.</i>	
Requisitos	5.2.1 5.2.7
Dificultad	Alta
Prioridad	Alta
Tareas	Implementar funcionalidades necesarias para la aplicación Android.

## 6.2.9 Finalización de la aplicación Android

<i>En esta fase se implementará la interface de la aplicación Android y se harán pruebas en dispositivos móviles.</i>	
Requisitos	5.2.1 5.2.7 5.2.8
Dificultad	Baja
Prioridad	Baja
Tareas	Implementar interface de la aplicación Android. Comprobar correcto funcionamiento en dispositivos Android.

<sup>18</sup> Parte visual de un software, es decir, donde el usuario interactúa y ve la información.

### 6.3 Riesgos en la planificación

Pese a todos los controles mencionados en la metodología siguen habiendo riesgos en la planificación que se tienen que encontrar y evaluar, estos riesgos vienen descritos a continuación con una explicación y con algunas soluciones que se podrían aplicar en caso de que surgieran.

**Problemas en la primera iteración:** La primera iteración es la que más define la continuidad y la estimación del proyecto, ya que a partir de su finalización, como ya se ha dicho, se acercará de forma más precisa la planificación. Sin embargo, puede ocurrir que en esta primera iteración pierda más tiempo del establecido adaptándose al ambiente de programación o que mi productividad no sea la que puedo llegar a conseguir cuando coja soltura en el desarrollo. **Solución:** Con el fin de evitar esto se intentará separar el trabajo productivo del perdido en adaptación de la primera iteración para una estimación de tiempos más eficaz.

**Variabilidad de la complejidad:** No todas las iteraciones tienen la misma complejidad, eso puede ser un problema, debido a que una iteración con un alto nivel de complejidad puede llevar a necesitar más tiempo. **Solución:** Para paliar este problema volvemos a hacer hincapié en la flexibilidad de la metodología, si me encontrará con una iteración muy costosa intentaría adaptar las horas para seguir cumpliendo los plazos estimados, en caso de imposibilidad se eliminaría alguna funcionalidad de baja prioridad que afectase lo menos posible al producto final.

**Desconocimiento sobre el lenguaje:** Dado que no soy un experto en los lenguajes de programación que se usarán, pese a tener un buen conocimiento, podría darse la situación que alguna parte del proyecto desconociera como desarrollarla. **Solución:** Dispongo de libros sobre los lenguajes de programación que se utilizarán y conozco distintos sitios web donde obtener información.

**Necesidad de nuevas funcionalidades:** Puede ocurrir que durante el desarrollo de algunas funcionalidades surja la necesidad de implementar otras funcionalidades para el correcto desarrollo de la primera. **Solución:** Durante las reuniones de planificación

siempre se acotarán las funcionalidades concretas de manera que se programe en la misma iteración todo lo necesario para que sea 100% funcional sin dependencias.

Los riesgos mencionados si son bien tratados y las soluciones pensadas son eficaces no contribuirán costes adicionales, en caso de no poder ser solventados comprenderán un incremento en los gastos de personal y maquinaria. Sin embargo, debido a la metodología usada y que las iteraciones de dificultad más alta no son consecutivas es muy asequible modificar las horas de duración de las tareas con mayor dificultad quitando a sus posteriores de dificultad inferior por tal de no afectar al tiempo establecido para finalizar el proyecto y no contribuir en un mayor gasto de recursos.

## 6.4 Diagrama de Gantt inicial

A continuación se muestra el diagrama de Gantt para la planificación inicial.

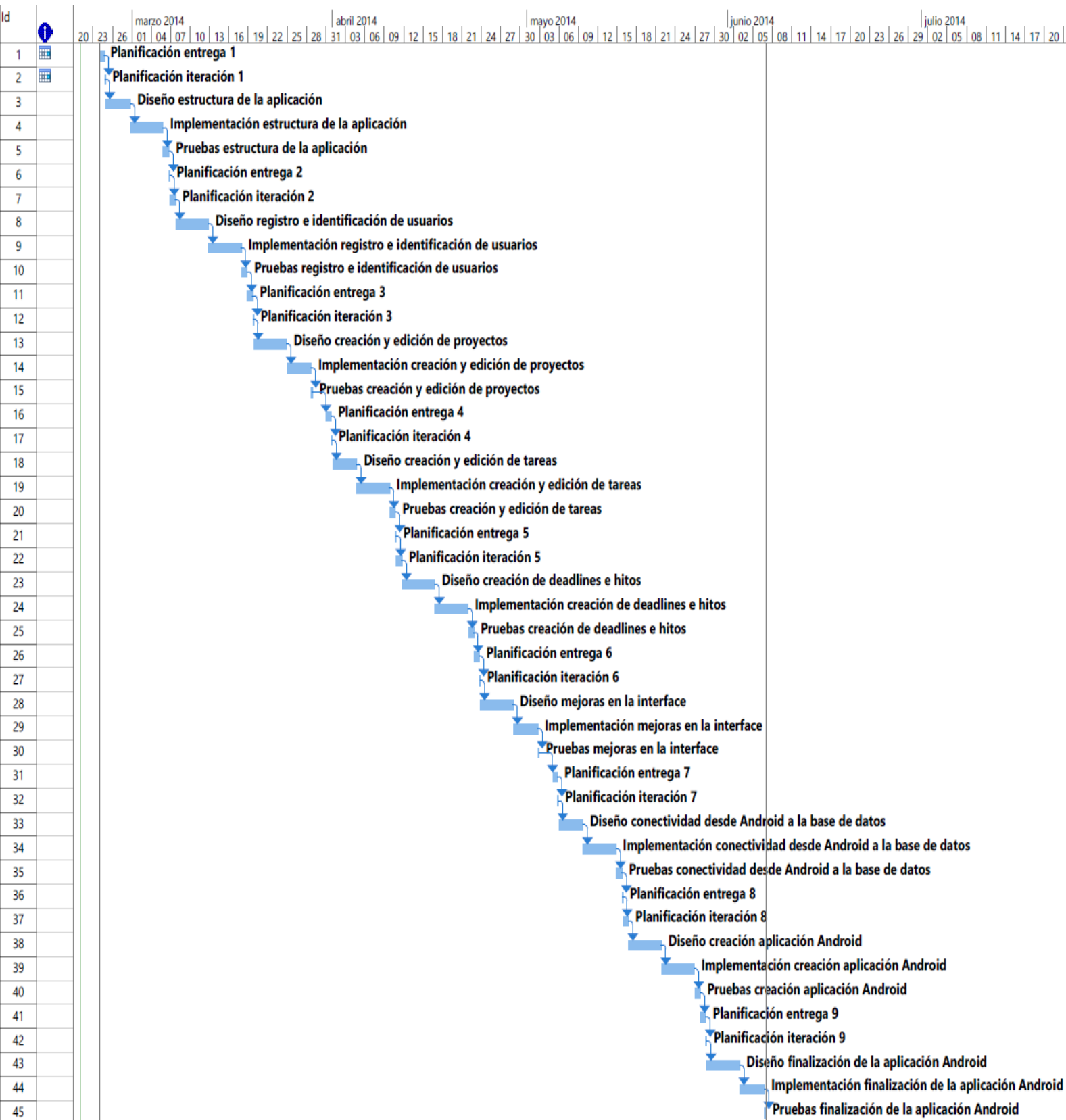


Figura 1 Diagrama de Gantt

## 6.5 Diagrama de Gantt final

A continuación se muestra el diagrama de Gantt resultante después del desarrollo de todo el proyecto con un comentario de los cambios que se han efectuado.



Figura 2 Diagrama de Gantt final

Se han marcado en **verde** las dos tareas que han visto cambios en sus tareas se detallan a continuación los cambios:

En la iteración 4, creación y edición de tareas se han añadido también las implementaciones de hitos, entregables y checklists debido a que se ha valorado que el tiempo estimado de implementación de la iteración era suficiente para desarrollar las funcionalidades mencionadas.

En la iteración 5, creación de deadlines e hitos se han realizado las funcionalidades de notificaciones, debido a que los hitos han sido implementados en la iteración anterior, por tanto, se han añadido las notificaciones que no habían sido consideradas en la planificación principal.

Se ha marcado en **rojo** la única iteración cuyo tiempo de implementación se ha visto aumentado, los motivos del retraso y la solución para continuar con el correcto desarrollo se explica en el punto [iteración 7: implementación conectividad desde Android a la base de datos.](#)

También hay que tener en cuenta que, a partir de la segunda tarea, las 6 horas previstas en planificación de entrega e iteración se han reducido a 3h y se han añadido a la implementación.

## 7 Presupuesto

### 7.1 Recursos humanos

El único coste humano involucrado en este proyecto es el autor. Para estimar el coste humano se ha consultado el convenio de empresas de ingeniería y oficinas de estudios técnicos, puesto que estaría en el nivel 1, licenciados y titulados 2º y 3º ciclo universitario y Analista, se le aplica un sueldo de 1.667,07 € brutos al mes, por tanto dividido entre las 160 horas mensuales de una jornada laboral completa, nos quedan 10,42€/h.

Personal	Coste
Autor	10,42€/h

Tabla 2 Costes recursos humanos

### 7.2 Recursos materiales

Los recursos materiales que se utilizarán para el desarrollo del proyecto los podemos dividir en cuatro partes:

**Software necesario para el desarrollo:** Aquí están incluidas todas las herramientas de software necesarias para programar, puesto que la aplicación estará programada usando Ruby on Rails, html5, css3y puede incluir partes de JQuery UI y JavaScript ninguno de los frameworks<sup>19</sup> necesarios para el desarrollo en estos lenguajes incluye ningún tipo de gasto ya que son gratuitos, por tanto estos gastos materiales no se incluirán en el presupuesto.

**Equipo utilizado por el desarrollador:** Esta parte incluye el ordenador utilizado y los gastos en electricidad que pueden generar el uso del equipo utilizado por el desarrollador. Para el coste del ordenador, sabiendo que costo **2000€** y que se puede **amortizar en 4 años**, dividimos su coste total entre el total de horas hábiles de estos 4 años.  $2000\text{€}/8064\text{h} = 0,25\text{€/h}$ . Para estimar el coste de la electricidad se ha consultado

---

<sup>19</sup> Herramienta de ayuda para el desarrollo de software

el consumo medio de un ordenador de las mismas características que ronda los 500w/h y he consultado la factura de luz más reciente para concluir que el coste es aproximadamente **0.20€/kwh**.

**Dispositivo Android:** A priori debido a que el entorno de desarrollo de Android ya incluye un emulador no sería necesario tener un dispositivo propio, la aplicación podrá ser perfectamente testada en el emulador y por tanto no generar coste.

Material	Coste
<b>Equipo</b>	0,25€/h
<b>Electricidad</b>	0,20€/h

*Tabla 3 Costes recursos materiales*

### 7.3 Costes imprevistos

Durante el desarrollo de este proyecto pueden surgir costes imprevistos que se intentarán encontrar en este apartado para poder pensar una posible solución a priori.

El principal problema que puede surgir son fallos en la planificación, sin embargo, como al principio de cada iteración se modificarán, si es necesario, los tiempos estimados por fase y se podrá decidir sobre las funcionalidades a implementar, en caso de producirse cambios solo afectarían a una iteración y no movería el resto de la planificación ya que al final de cada una recordar que el código que se entrega ya es completamente funcional. Si por cualquier imprevisto alguna iteración tuviera que durar más de lo previsto se intentaría eliminar alguna otra iteración por tal de no alargar más la planificación y los costes por consiguiente.

Gracias a estos métodos de control y la flexibilidad que da el seguir la metodología escogida los costes imprevistos se pueden considerar casi nulos siempre y cuando se respete la planificación establecida, que por supuesto será uno de los principales objetivos durante el desarrollo del proyecto, sin embargo, se añadirá el coste equivalente a media iteración más por si alguna iteración se alargara un poco más de la cuenta o pudieran encontrarse problemas inevitables como pudieran ser un corte en el suministro eléctrico.

Recurso	Coste (25h)
Autor	260,5€
Electricidad	5€
Equipo	6,25€
<b>Total</b>	<b>271,75€</b>

Tabla 4 Costes imprevistos

## 7.4 Coste total

### 7.4.1 Coste del desarrollo del proyecto

Juntando los costes desglosados en los dos apartados anteriores se ha construido la siguiente tabla con todos los costes del proyecto.

Recurso	Coste (450h)
Equipo	202,5€
Autor	4689€
Costes imprevistos	271,75€
<b>Total sin I.V.A</b>	<b>5163,25€</b>
<b>I.V.A</b>	<b>1084,28€</b>
<b>Total</b>	<b>6247,53€</b>

Tabla 5 Costes del desarrollo del proyecto

### 7.4.2 Coste de mantenimiento por año

Durante un año se tendrá que pagar el alquiler de servidores proporcionado por Heroku, se tiene en cuenta que la aplicación funcionara 24/7, por tanto partiendo de que un año tiene 8760h y que el coste por hora es de 0,15€/h aproximadamente con la suficiente potencia como para cubrir a un número elevado de usuarios se crea la siguiente tabla.

Recurso	Coste (8760h)
Servidor	1086€
I.V.A	<b>228€</b>
<b>Total</b>	<b>1314€</b>

*Tabla 6 Costes mantenimiento primer año*

## 7.5 Viabilidad económica

El proyecto se intentará financiar desde el inicio con publicidad utilizando servicios como google AdSense<sup>20</sup>, sin embargo el cálculo de las ganancias potenciales es bastante complicado e inexacto así que he diseñado un gráfico y una tabla donde se expresan las ganancias que debería generar la publicidad para poder mantener el coste de la aplicación y recuperar la inversión inicial.

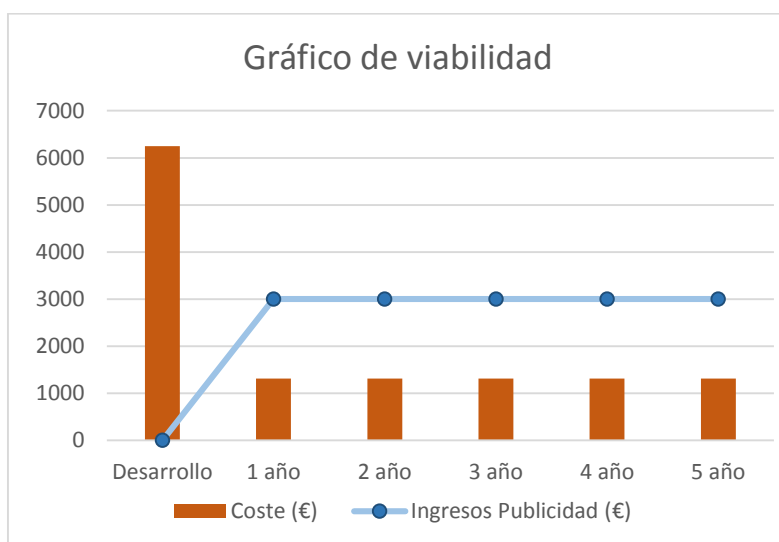


Figura 3 Gráfico viabilidad

En el gráfico y la tabla mostrada se puede ver como estimando unos 3000€ de ingreso anuales por publicidad, cifra bastante alcanzable, se puede llegar a amortizar la inversión inicial en 5 años. Es muy probable que a medida que

	Coste (€)	Ingresos (€)
Desarrollo	6247,53€	0€
1 año	1314€	3000€
2 año	1314€	3000€
3 año	1314€	3000€
4 año	1314€	3000€
5 año	1314€	3000€
<b>Total</b>	<b>12817,53€</b>	<b>15000€</b>

Tabla 7 Costes e ingresos estimados de la aplicación

crezca la comunidad de usuarios en la aplicación se empiecen a interesar también pequeñas empresas a las que se podría pedir un pago y por tanto generar más beneficio y que además aportarían una cantidad mayor de usuarios y por tanto los ingresos estimados podrían crecer y por consiguiente favorecer a la amortización del proyecto.

<sup>20</sup> Servicio de Google que añade publicidad automáticamente a cualquier sitio web siempre y cuando cumpla sus condiciones de uso.

## 8 Sostenibilidad

En esta sección se adentrará el tema de la sostenibilidad, debido a su importancia actual en el marco global y a la responsabilidad social, como se verá en adelante el proyecto tratará de ser sostenible e intentará estar comprometido con la sociedad.

Anteriormente se ha podido comprobar que la mayoría de costes que genera nuestra aplicación son costes de recursos humanos por tanto no implica ningún coste medioambiental. Sin embargo, aparte de los costes humanos también se ha de contar con el impacto medioambiental que generarán los servidores donde se alojara la aplicación web, por ende, he escogido los servidores de Heroku, entre otras razones, por la capacidad de escalar la aplicación bajo demanda, en otras palabras, solo generara consumo de infraestructura en medida que la aplicación lo necesite por ese motivo el impacto medioambiental será el mínimo necesario para el correcto funcionamiento de la aplicación.

Referente a la responsabilidad social, debido a la situación mundial actual, la aplicación se mantendrá gratuita para el uso personal y por tanto cualquier persona de cualquier nivel cultural o social podrá acceder a ella y tener sus proyectos sin ningún tipo de preocupación, este proyecto se compromete con los más desfavorecidos.

Para controlar y cuantificar la responsabilidad social del proyecto, debido a que es el único factor que repercute directamente en el proyecto, se propone hacer una encuesta a los usuarios que contenga las preguntas: ¿Por qué usas esta aplicación?, ¿Estarías dispuesto a pagar por el uso de la aplicación? ¿Por qué?, ¿Usas la aplicación para proyectos propios, como herramienta de ayuda en los estudios, o como empresa? Gracias a las respuestas de los usuarios se podría determinar el impacto social de la aplicación y si realmente cumple su responsabilidad social de ayudar a por ejemplo estudiantes que por lo general no tienen muchos ingresos o bien a gente que tiene proyectos personales pero pocos recursos para llevarlos a cabo y si realmente el uso de la aplicación recae en su uso gratuito como se plantea en los objetivos.

## 9 Especificación de requisitos aplicación web y Android

En este apartado se definen los requisitos funcionales y no funcionales que debe tener la aplicación.

### 9.1 Requisitos funcionales

#### 9.1.1 Diagrama de casos de uso

Dentro de la aplicación encontramos diferentes tipos de usuario:

**Usuarios registrados:** Son aquellos que están registrados en el sistema y quedan subdivididos en tres categorías en función de sus permisos dentro de un proyecto, están diferenciados en 3 categorías:

Primera categoría (**Jefe de proyecto**): Puede hacer todo lo que quiera en el proyecto (crear tareas, asignar gente, eliminar tareas...)

Segunda categoría (**Gestor de proyecto**): Esta categoría no puede asignar gente al proyecto, pero sí a las tareas. Puede hacer todo lo demás.

Tercera categoría (**Trabajador**): Solo pueden ver el proyecto y modificar el estado de un entregable o checklist para darlo por válido, no puede crear nuevas tareas ni asignar gente ni a tareas ni al proyecto, es decir, es un usuario registrado sin privilegios.

**Usuario desconocido:** Este grupo pertenece a aquellos usuarios que no están registrados en el sistema.

**Usuario Android:** Este grupo son los usuarios que se conectan desde la aplicación Android, como se puede ver solo implementa un par de casos de uso de la aplicación en general.

A continuación se muestra el diagrama de casos de uso para todos los usuarios.



### 9.1.2 Casos de uso

A continuación se detallan todos los casos de uso, para ello se crea una tabla con un valor numérico para identificar cada caso de uso, los actores que realizan el caso de uso, las condiciones que se han de cumplir antes y después de realizar el caso de uso, y por último como se desarrolla la interacción entre usuario y sistema.

Registrar usuario	
Número: <b>1</b>	Actor primario: <b>Usuario Desconocido</b>
Precondición: <b>El usuario no está registrado.</b>	
Poscondición: <b>El usuario está registrado.</b>	
Disparador: <b>El usuario quiere registrarse en el sistema.</b>	
Escenario Principal: <b>1 El usuario rellena el formulario de registro de usuario e indica al sistema que quiere registrarse.</b> <b>2 El sistema valida los datos y envía un email de confirmación.</b>	
Extensiones: <b>2a El sistema detecta algún error en alguno de los campos.</b> <b>2a1 El sistema notifica al usuario los datos incorrectos.</b> <b>2a2 El sistema redirige al usuario al paso 1.</b>	

Iniciar sesión	
Número: <b>2</b>	Actor primario: <b>Usuario Registrado, Usuario Android</b>
Precondición: <b>El usuario está registrado.</b>	
Poscondición: <b>El usuario tiene una sesión en el sistema.</b>	
Disparador: <b>El usuario quiere iniciar sesión en el sistema.</b>	
Escenario Principal: <b>1 El usuario rellena el formulario de inicio de sesión e indica al sistema que quiere iniciar sesión.</b> <b>2 El sistema valida los datos y redirige al usuario a la página principal.</b>	
Extensiones: <b>2a El sistema detecta algún error en alguno de los campos.</b> <b>2a1 El sistema notifica al usuario los datos incorrectos.</b> <b>2a2 El sistema redirige al usuario al paso 1.</b>	

Cerrar sesión	
Número: <b>3</b>	Actor primario: <b>Usuario Registrado</b>
Precondición: <b>El usuario ha iniciado sesión.</b>	
Poscondición: <b>El usuario deja de tener una sesión en el sistema.</b>	
Disparador: <b>El usuario quiere cerrar sesión en el sistema.</b>	
Escenario Principal: <b>1 El usuario notifica al sistema que quiere cerrar sesión.</b> <b>2 El sistema cierra elimina la sesión del usuario.</b>	
Extensiones:	

## Crear nuevo proyecto

Número: **4**

Actor primario: **Usuario Registrado**

Precondición: **El usuario ha iniciado sesión.**

Poscondición: **Se ha creado un nuevo proyecto.**

Disparador: **El usuario quiere crear un nuevo proyecto.**

Escenario Principal:

- 1 El usuario notifica al sistema que quiere crear un nuevo proyecto.**
- 2 El sistema muestra el formulario de creación de un nuevo proyecto.**
- 3 El usuario rellena el formulario y notifica al sistema la finalización.**
- 4 El sistema valida los datos y crea el nuevo proyecto.**

Extensiones:

**3a El usuario cancela la creación del proyecto.**

**3a1 El usuario notifica que cancela la creación del proyecto al sistema.**

**3a2 El caso de uso acaba.**

**4a El sistema detecta algún error en los campos.**

**4a1 El sistema notifica al usuario los datos incorrectos.**

**4a2 El sistema redirige al usuario al paso 3.**

## Editar proyecto

Número: 5

Actor primario: **Usuario Registrado\***

Precondición: **El usuario ha iniciado sesión.**

**Existe el proyecto que el usuario desea modificar.**

**\*El usuario tiene permiso para modificar el proyecto.**

Poscondición: **Se ha modificado el proyecto.**

Disparador: **El usuario quiere editar un proyecto.**

Escenario Principal:

- 1 El usuario notifica al sistema que quiere modificar un proyecto.**
- 2 El sistema muestra el formulario de modificación del proyecto.**
- 3 El usuario rellena el formulario y notifica al sistema la finalización.**
- 4 El sistema valida los datos y modifica el proyecto.**

Extensiones:

**3a El usuario cancela la edición del proyecto.**

**3a1 El usuario notifica que cancela la creación del proyecto al sistema.**

**3a2 El caso de uso acaba.**

**4a El sistema detecta algún error en los campos.**

**4a1 El sistema notifica al usuario los datos incorrectos.**

**4a2 El sistema redirige al usuario al paso 3.**

## Eliminar proyecto

Número: **6**

Actor primario: **Usuario Registrado\***

Precondición: **El usuario ha iniciado sesión.**

**Existe el proyecto que el usuario desea eliminar.**

**\*El usuario tiene permiso para eliminar el proyecto.**

Poscondición: **Se ha eliminado el proyecto.**

Disparador: **El usuario quiere eliminar un proyecto.**

Escenario Principal:

**1 El usuario notifica al sistema que quiere eliminar un proyecto.**

**2 El sistema pide la confirmación al usuario para eliminar el proyecto.**

**3 El usuario confirma la eliminación.**

**4 El sistema elimina el proyecto.**

Extensiones:

**3a El usuario no confirma la eliminación.**

**3a1 El usuario no confirma la eliminación.**

**3a2 El caso de uso acaba.**

## Asignar miembro a un proyecto

Número: 7

Actor primario: **Usuario Registrado\***

Precondición: **El usuario ha iniciado sesión.**

**Existe el proyecto al que el usuario desea asignar a otro usuario.**

**\*El usuario tiene permiso para asignar un usuario a un proyecto.**

Poscondición: **Se ha creado una nueva asignación entre un usuario y el proyecto.**

Disparador: **El usuario quiere asignar un usuario a un proyecto.**

Escenario Principal:

- 1 El usuario notifica al sistema que quiere asignar un usuario a un proyecto.**
- 2 El sistema muestra el formulario de asignación.**
- 3 El usuario rellena el formulario y notifica al sistema la finalización.**
- 4 El sistema valida los datos y crea la nueva asignación.**

Extensiones:

**4a El usuario que se ha asignado no existe.**

**4a1 El sistema guarda la asignación para cuando se registre el usuario.**

**4a2 El caso de uso acaba.**

## Editar miembro de un proyecto

Número: **8**

Actor primario: **Usuario Registrado\***

Precondición: **El usuario ha iniciado sesión.**

**Existe el proyecto al que el usuario desea asignar a otro usuario.**

**Existe la asignación entre el usuario y el proyecto.**

**\*El usuario tiene permiso para asignar un usuario a un proyecto.**

Poscondición: **Se ha editado la asignación del usuario al proyecto.**

Disparador: **El usuario quiere editar la asignación de un usuario a un proyecto.**

Escenario Principal:

- 1 El usuario notifica al sistema que quiere editar una asignación.**
- 2 El sistema muestra el formulario de modificación de asignación.**
- 3 El usuario rellena el formulario y notifica al sistema la finalización.**
- 4 El sistema valida los datos y modifica la asignación.**

Extensiones:

**2a El usuario que se ha asignado no existe.**

**2a1 El sistema guarda la asignación para cuando se registre el usuario.**

**2a2 El caso de uso acaba.**

## Eliminar miembro de un proyecto

Número: **9**

Actor primario: **Usuario Registrado\***

Precondición: **El usuario ha iniciado sesión.**

**Existe la asignación entre el usuario y el proyecto.**

**\*El usuario tiene permiso para asignar un usuario a un proyecto.**

Poscondición: **Se ha eliminado la asignación del usuario al proyecto.**

Disparador: **El usuario quiere eliminar la asignación de un usuario a un proyecto.**

Escenario Principal:

- 1 El usuario notifica al sistema que quiere eliminar una asignación.**
- 2 El sistema pide la confirmación al usuario para eliminar la asignación.**
- 3 El usuario confirma la eliminación.**
- 4 El sistema elimina la asignación.**

Extensiones:

**3a El usuario cancela la eliminación de la asignación.**

**3a1 El usuario notifica que cancela la eliminación de la asignación al sistema**

**3a2 El caso de uso acaba**

## Crear nueva tarea

Número: **10**

Actor primario: **Usuario Registrado\***

Precondición: **El usuario ha iniciado sesión.**

**Existe el proyecto al que el usuario desea añadir una nueva tarea.**

**\*El usuario tiene permiso para crear una nueva tarea en el proyecto.**

Poscondición: **Se ha creado una nueva tarea.**

Disparador: **El usuario quiere añadir una nueva tarea a un proyecto.**

Escenario Principal:

- 1 El usuario notifica al sistema que quiere crear una nueva tarea.**
- 2 El sistema muestra el formulario de creación de una nueva tarea.**
- 3 El usuario rellena el formulario y notifica al sistema la finalización.**
- 4 El sistema valida los datos y crea la nueva tarea.**

Extensiones:

**3a El usuario cancela la creación del hito.**

**3a1 El usuario notifica que cancela la creación de la tarea al sistema.**

**3a2 El caso de uso acaba.**

**4a El sistema detecta algún error en los campos.**

**4a1 El sistema notifica al usuario los datos incorrectos.**

**4a2 El sistema redirige al usuario al paso 3.**

## Editar tarea

Número: **11**

Actor primario: **Usuario Registrado\***

Precondición: **El usuario ha iniciado sesión.**

**Existe la tarea que el usuario desea modificar.**

**\*El usuario tiene permiso para modificar la tarea.**

Poscondición: **Se ha modificado la tarea.**

Disparador: **El usuario quiere modificar una tarea.**

Escenario Principal:

- 1 El usuario notifica al sistema que quiere modificar una tarea.**
- 2 El sistema muestra el formulario de modificación de una tarea.**
- 3 El usuario rellena el formulario y notifica al sistema la finalización.**
- 4 El sistema valida los datos y modifica la tarea.**

Extensiones:

**3a El usuario cancela la modificación del hito.**

**3a1 El usuario notifica que cancela la modificación de la tarea al sistema.**

**3a2 El caso de uso acaba.**

**4a El sistema detecta algún error en los campos.**

**4a1 El sistema notifica al usuario los datos incorrectos.**

**4a2 El sistema redirige al usuario al paso 3.**

## Eliminar tarea

Número: **12**

Actor primario: **Usuario Registrado\***

Precondición: **El usuario ha iniciado sesión.**

**Existe la tarea que el usuario desea eliminar.**

**\*El usuario tiene permiso para eliminar la tarea.**

Poscondición: **Se ha eliminado la tarea.**

Disparador: **El usuario quiere eliminar una tarea.**

Escenario Principal:

**1 El usuario notifica al sistema que quiere eliminar una tarea.**

**2 El sistema pide la confirmación al usuario para eliminar la tarea.**

**3 El usuario confirma la eliminación.**

**4 El sistema elimina la tarea.**

Extensiones:

**3a El usuario no confirma la eliminación.**

**3a1 El usuario no confirma la eliminación.**

**3a2 El caso de uso acaba.**

## Asignar miembro a una tarea

Número: **13**

Actor primario: **Usuario Registrado\***

Precondición: **El usuario ha iniciado sesión.**

**Existe la tarea a la que el usuario desea asignar a otro usuario.**

**\*El usuario tiene permiso para asignar un miembro a la tarea.**

Poscondición: **Se ha creado una nueva asignación entre un usuario y el proyecto.**

Disparador: **El usuario quiere asignar un usuario a una tarea.**

Escenario Principal:

- 1 El usuario notifica al sistema que quiere asignar un usuario a una tarea.**
- 2 El sistema muestra los usuarios que se pueden asignar a la tarea.**
- 3 El usuario indica que usuarios quiere asignar e indica la finalización.**
- 4 El sistema crea la nueva asignación.**

Extensiones:

## Eliminar miembro de una tarea

Número: **14**

Actor primario: **Usuario Registrado\***

Precondición: **El usuario ha iniciado sesión.**

**Existe la asignación entre el usuario y la tarea.**

**\*El usuario tiene permiso para eliminar un miembro a la tarea.**

Poscondición: **Se ha eliminado la asignación entre un usuario y la tarea.**

Disparador: **El usuario quiere eliminar una asignación entre un usuario y una tarea.**

Escenario Principal:

**1 El usuario notifica al sistema que quiere eliminar una asignación.**

**2 El sistema pide la confirmación al usuario para eliminar la asignación.**

**3 El usuario confirma la eliminación.**

**4 El sistema elimina la asignación.**

Extensiones:

**3a El usuario cancela la eliminación de la asignación.**

**3a1 El usuario notifica que cancela la eliminación de la asignación al sistema**

**3a2 El caso de uso acaba**

## Crear nuevo hito

Número: **15**

Actor primario: **Usuario Registrado\***

Precondición: **El usuario ha iniciado sesión.**

**Existe el proyecto al que el usuario desea añadir un nuevo hito.**

**\*El usuario tiene permiso para crear un nuevo hito en el proyecto.**

Poscondición: **Se ha creado un nuevo hito.**

Disparador: **El usuario quiere añadir un nuevo hito a un proyecto.**

Escenario Principal:

- 1 El usuario notifica al sistema que quiere crear un nuevo hito.**
- 2 El sistema muestra el formulario de creación de un nuevo hito.**
- 3 El usuario rellena el formulario y notifica al sistema la finalización.**
- 4 El sistema valida los datos y crea el nuevo hito.**

Extensiones:

**3a El usuario cancela la creación del hito.**

**3a1 El usuario notifica que cancela la creación del hito al sistema.**

**3a2 El caso de uso acaba.**

**4a El sistema detecta algún error en los campos.**

**4a1 El sistema notifica al usuario los datos incorrectos.**

**4a2 El sistema redirige al usuario al paso 3.**

## Editar hito

Número: **16**

Actor primario: **Usuario Registrado\***

Precondición: **El usuario ha iniciado sesión.**

**Existe el hito que el usuario desea modificar.**

**\*El usuario tiene permiso para modificar el hito.**

Poscondición: **Se ha modificado el hito.**

Disparador: **El usuario quiere modificar un hito.**

Escenario Principal:

- 1 El usuario notifica al sistema que quiere modificar un hito.**
- 2 El sistema muestra el formulario de modificación de un hito.**
- 3 El usuario rellena el formulario y notifica al sistema la finalización.**
- 4 El sistema valida los datos y modifica el hito.**

Extensiones:

**3a El usuario cancela la modificación del hito.**

**3a1 El usuario notifica que cancela la modificación del hito al sistema.**

**3a2 El caso de uso acaba.**

**4a El sistema detecta algún error en los campos.**

**4a1 El sistema notifica al usuario los datos incorrectos.**

**4a2 El sistema redirige al usuario al paso 3.**

Eliminar hito	
Número: <b>17</b>	Actor primario: <b>Usuario Registrado*</b>
<p>Precondición: <b>El usuario ha iniciado sesión.</b></p> <p><b>Existe el hito que el usuario desea eliminar.</b></p> <p><b>*El usuario tiene permiso para eliminar el hito.</b></p>	
Poscondición: <b>Se ha eliminado el hito.</b>	
Disparador: <b>El usuario quiere eliminar un hito.</b>	
<p>Escenario Principal:</p> <ol style="list-style-type: none"> <li><b>1 El usuario notifica al sistema que quiere eliminar un hito.</b></li> <li><b>2 El sistema pide la confirmación al usuario para eliminar el hito.</b></li> <li><b>3 El usuario confirma la eliminación.</b></li> <li><b>4 El sistema elimina el hito.</b></li> </ol>	
<p>Extensiones:</p> <ol style="list-style-type: none"> <li><b>3a El usuario no confirma la eliminación.</b> <ol style="list-style-type: none"> <li><b>3a1 El usuario no confirma la eliminación.</b></li> <li><b>3a2 El caso de uso acaba.</b></li> </ol> </li> </ol>	

## Crear nuevo entregable

Número: **18**

Actor primario: **Usuario Registrado\***

Precondición: **El usuario ha iniciado sesión.**

**Existe la tarea a la que el usuario desea añadir un nuevo entregable.**

**\*El usuario tiene permiso para crear un nuevo entregable en la tarea.**

Poscondición: **Se ha creado un nuevo entregable.**

Disparador: **El usuario quiere añadir un nuevo entregable a una tarea.**

Escenario Principal:

- 1 El usuario notifica al sistema que quiere crear un nuevo entregable.**
- 2 El sistema muestra el formulario de creación de un nuevo entregable.**
- 3 El usuario rellena el formulario y notifica al sistema la finalización.**
- 4 El sistema valida los datos y crea el nuevo entregable.**

Extensiones:

**3a El usuario cancela la creación del entregable.**

**3a1 El usuario notifica que cancela la creación del entregable al sistema.**

**3a2 El caso de uso acaba.**

**4a El sistema detecta algún error en los campos.**

**4a1 El sistema notifica al usuario los datos incorrectos.**

**4a2 El sistema redirige al usuario al paso 3.**

## Editar estado entregable

Número: **19**

Actor primario: **Usuario Registrado**

Precondición: **El usuario ha iniciado sesión.**

**Existe el entregable que el usuario desea modificar el estado.**

Poscondición: **Se ha modificado el estado del entregable.**

Disparador: **El usuario quiere modificar el estado de un entregable.**

Escenario Principal:

**1 El usuario notifica al sistema que quiere modificar el estado de un entregable.**

**2 El sistema modifica el estado del entregable.**

Extensiones:

## Eliminar entregable

Número: **20**

Actor primario: **Usuario Registrado\***

Precondición: **El usuario ha iniciado sesión.**

**Existe el entregable que el usuario desea eliminar.**

**\*El usuario tiene permiso para eliminar el entregable.**

Poscondición: **Se ha eliminado el entregable.**

Disparador: **El usuario quiere eliminar un entregable.**

Escenario Principal:

- 1 El usuario notifica al sistema que quiere eliminar un entregable.**
- 2 El sistema pide la confirmación al usuario para eliminar el entregable.**
- 3 El usuario confirma la eliminación.**
- 4 El sistema elimina el entregable.**

Extensiones:

- 3a El usuario no confirma la eliminación.**
  - 3a1 El usuario no confirma la eliminación.**
  - 3a2 El caso de uso acaba.**

## Crear nuevo checklist

Número: **21**

Actor primario: **Usuario Registrado\***

Precondición: **El usuario ha iniciado sesión.**

**Existe la tarea a la que el usuario desea añadir un nuevo checklist.**

**\*El usuario tiene permiso para crear un nuevo checklist en la tarea.**

Poscondición: **Se ha creado un nuevo checklist.**

Disparador: **El usuario quiere añadir un nuevo checklist a una tarea.**

Escenario Principal:

- 1 El usuario notifica al sistema que quiere crear un nuevo checklist.**
- 2 El sistema muestra el formulario de creación de un nuevo checklist.**
- 3 El usuario rellena el formulario y notifica al sistema la finalización.**
- 4 El sistema valida los datos y crea el nuevo checklist.**

Extensiones:

**3a El usuario cancela la creación del checklist.**

**3a1 El usuario notifica que cancela la creación del checklist al sistema.**

**3a2 El caso de uso acaba.**

**4a El sistema detecta algún error en los campos.**

**4a1 El sistema notifica al usuario los datos incorrectos.**

**4a2 El sistema redirige al usuario al paso 3.**

## Editar estado checklist

Número: **22**

Actor primario: **Usuario Registrado**

Precondición: **El usuario ha iniciado sesión.**

**Existe el checklist que el usuario desea modificar el estado.**

Poscondición: **Se ha modificado el estado del checklist.**

Disparador: **El usuario quiere modificar el estado de un checklist.**

Escenario Principal:

**1 El usuario notifica al sistema que quiere modificar el estado de un checklist.**

**2 El sistema modifica el estado del checklist.**

Extensiones:

## Eliminar checklist

Número: **23**

Actor primario: **Usuario Registrado\***

Precondición: **El usuario ha iniciado sesión.**

**Existe el checklist que el usuario desea eliminar.**

**\*El usuario tiene permiso para eliminar el checklist.**

Poscondición: **Se ha eliminado el checklist.**

Disparador: **El usuario quiere eliminar un checklist.**

Escenario Principal:

- 1 El usuario notifica al sistema que quiere eliminar un checklist.**
- 2 El sistema pide la confirmación al usuario para eliminar el checklist.**
- 3 El usuario confirma la eliminación.**
- 4 El sistema elimina el checklist.**

Extensiones:

- 3a El usuario no confirma la eliminación.**
  - 3a1 El usuario no confirma la eliminación.**
  - 3a2 El caso de uso acaba.**

Ver proyectos	
Número: <b>24</b>	Actor primario: <b>Usuario Registrado</b>
Precondición: <b>El usuario ha iniciado sesión.</b>	
Poscondición:	
Disparador: <b>El usuario quiere ver sus proyectos.</b>	
Escenario Principal: <b>1 El usuario notifica que quiere ver sus proyectos.</b> <b>2 El sistema muestra los proyectos del usuario.</b>	
Extensiones:	

Ver tareas	
Número: <b>25</b>	Actor primario: <b>Usuario Registrado</b>
Precondición: <b>El usuario ha iniciado sesión.</b> <b>El proyecto existe.</b>	
Poscondición:	
Disparador: <b>El usuario quiere ver las tareas de un proyecto.</b>	
Escenario Principal: <b>1 El usuario notifica que quiere ver las tareas de un proyecto</b> <b>2 El sistema muestra las tareas de un proyecto.</b>	
Extensiones:	

Ver diagrama de Gantt	
Número: <b>26</b>	Actor primario: <b>Usuario Registrado</b>
Precondición: <b>El usuario ha iniciado sesión. El proyecto existe.</b>	
Poscondición:	
Disparador: <b>El usuario quiere ver el diagrama de Gantt de un proyecto.</b>	
Escenario Principal: <b>1 El usuario notifica que quiere ver el diagrama de Gantt de un proyecto. 2 El sistema muestra el diagrama de Gantt de un proyecto.</b>	
Extensiones:	

Recibir notificaciones	
Número: <b>27</b>	Actor primario: <b>Usuario Registrado, Usuario Android</b>
Precondición: <b>El usuario ha iniciado sesión.</b>	
Poscondición:	
Disparador:	
Escenario Principal: <b>1 El sistema envía notificaciones al usuario acerca del estado de sus proyectos.</b>	
Extensiones:	

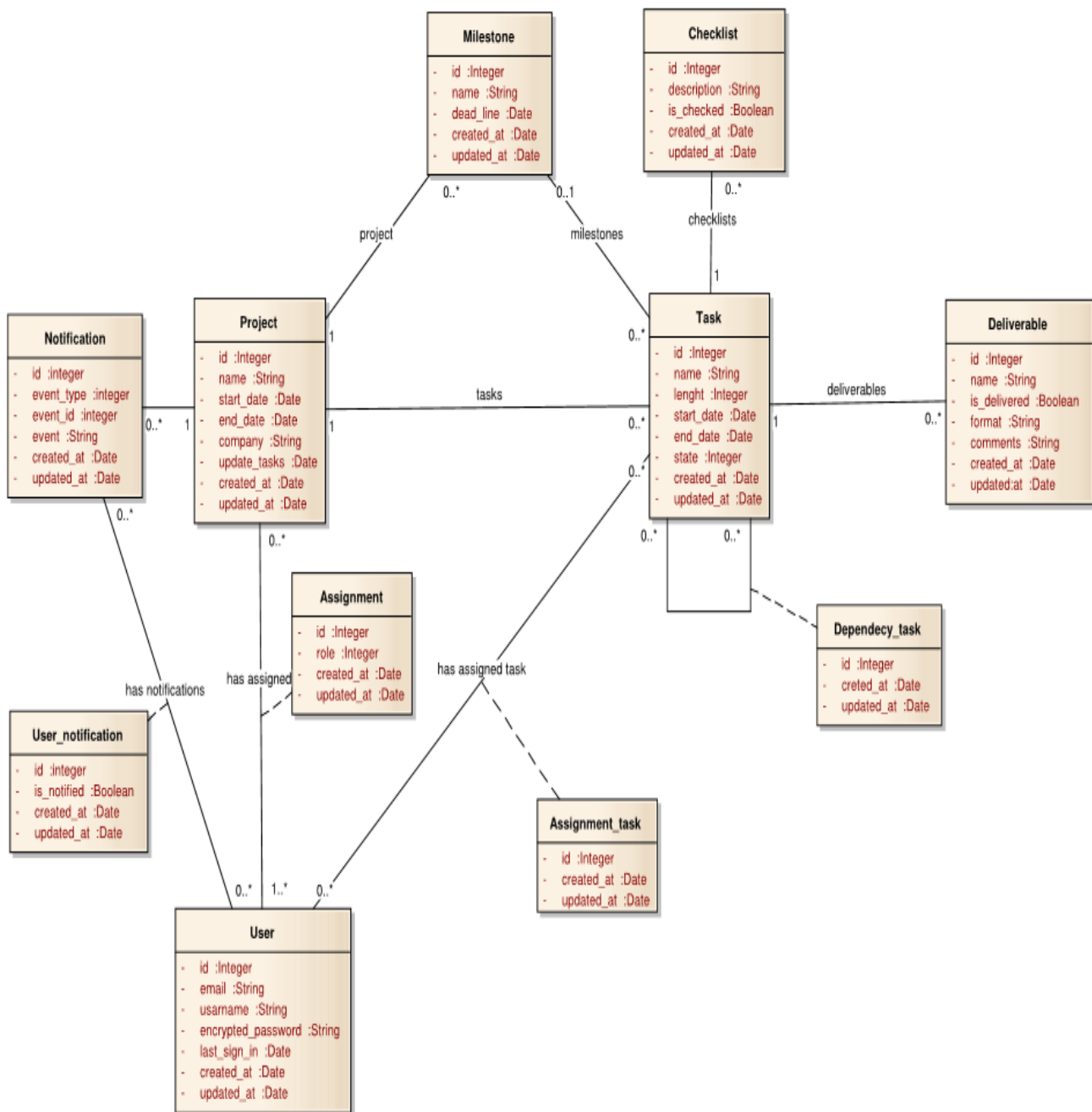
Ver ayuda	
Número: <b>28</b>	Actor primario: <b>Usuario Registrado</b>
Precondición: <b>El usuario ha iniciado sesión.</b>	
Poscondición:	
Disparador: <b>El usuario quiere ver la ayuda.</b>	
Escenario Principal: <b>1 El usuario notifica que quiere ver la ayuda.</b> <b>2 El sistema muestra la ayuda.</b>	
Extensiones:	

\*Consultar en el diagrama de casos de uso que tipo de usuario registrado puede realizar que caso de uso.

Todos los requisitos del sistema por tanto han sido convertidos a casos de uso.

### 9.1.3 Modelo conceptual

Debido a la metodología aplicada el modelo conceptual no ha sido estático desde el principio sino que ha ido aumentando durante los sprints, a continuación se muestra el modelo conceptual final.



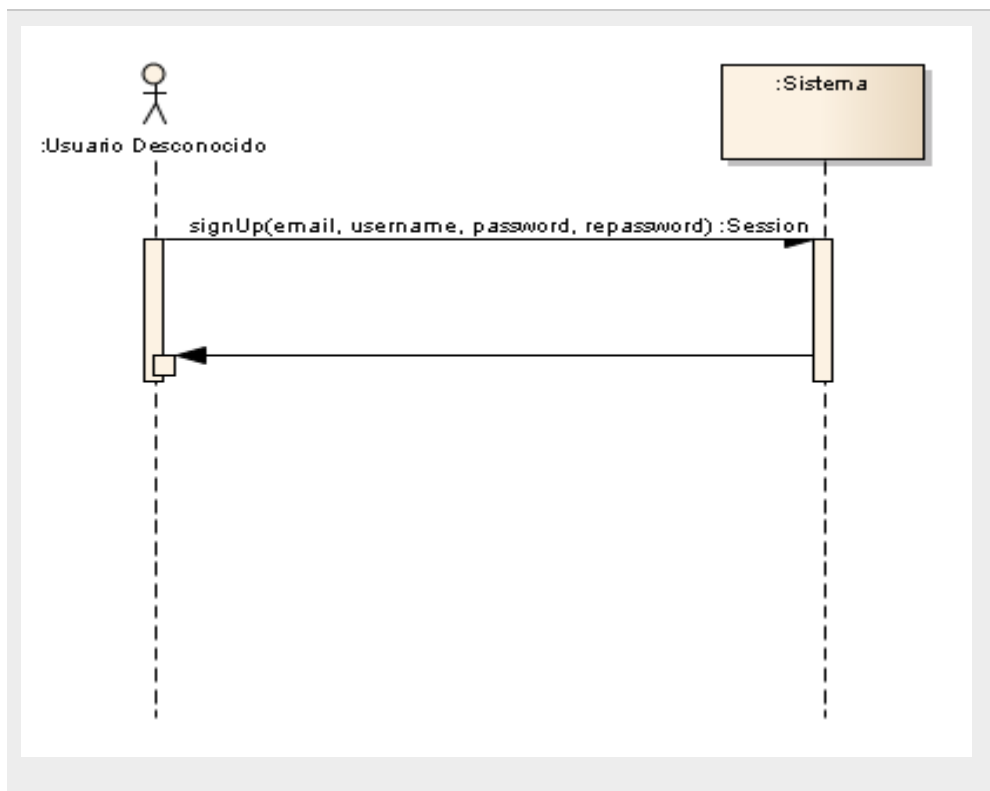
### Restricciones textuales

- RT1: La clave primaria de todas las clases es id.
- RT2: Assignment.role tiene valores [0..2].
- RT3: Task.state tiene valores [0..3].
- RT4: Un Task y un Milestone han de pertenecer al mismo proyecto.
- RT5: Un Notification y un User han de pertenecer al mismo proyecto.
- RT6: Las fechas de un Task tienen que estar contenidas en las fechas de su proyecto.
- RT7: Milestone.end\_line tiene que estar contenido en las fechas de su proyecto.
- RT8: Un Task y un User han de pertenecer al mismo proyecto.
- RT9: Dos Task con dependencia deben estar en el mismo proyecto.
- RT10: Dos Task con dependencia deben empezar una después de la otra en orden de dependencia lógico.

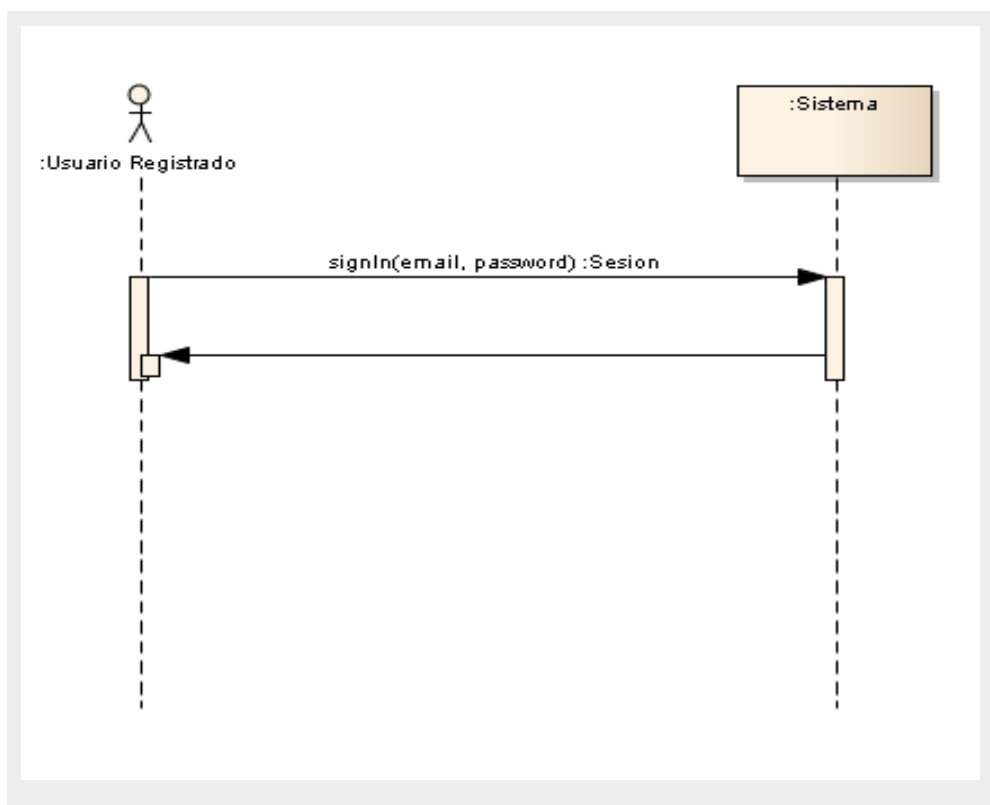
Figura 5 Modelo conceptual

## 9.1.4 Modelo del comportamiento

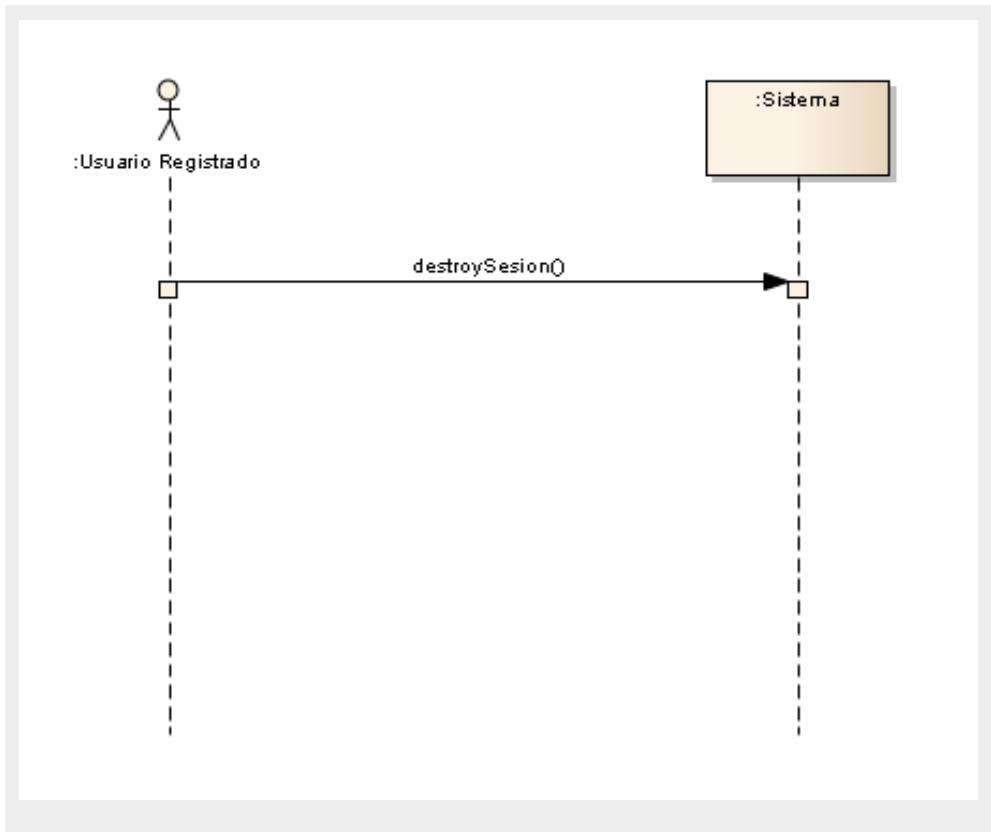
### 1. Registrar usuario



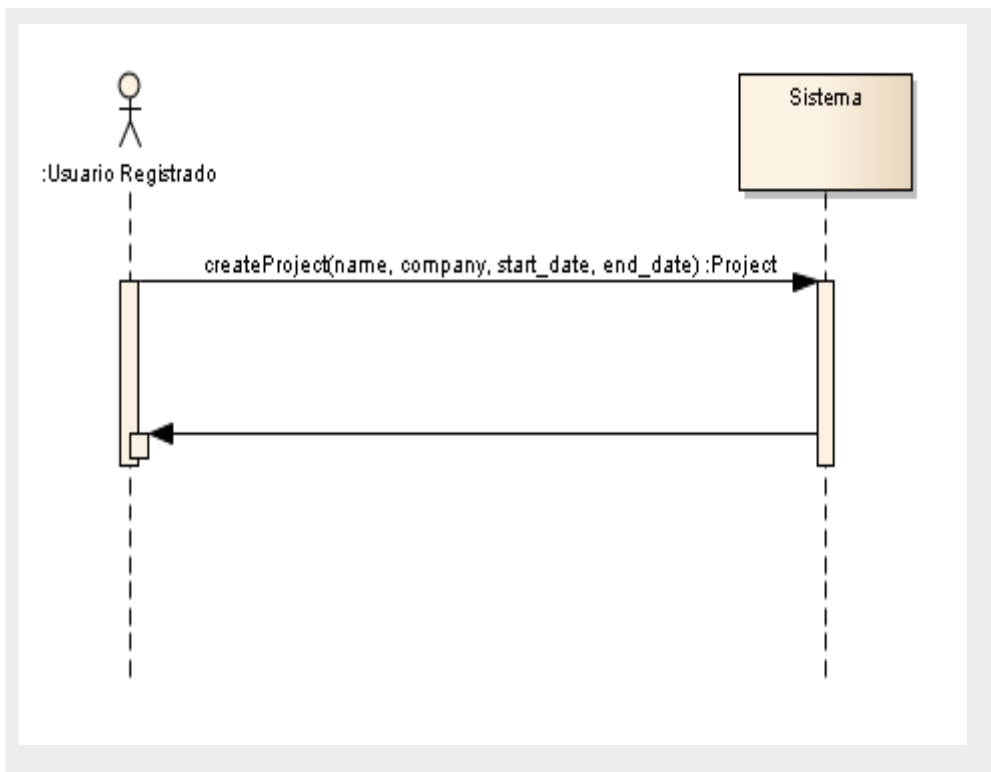
### 2. Iniciar sesión



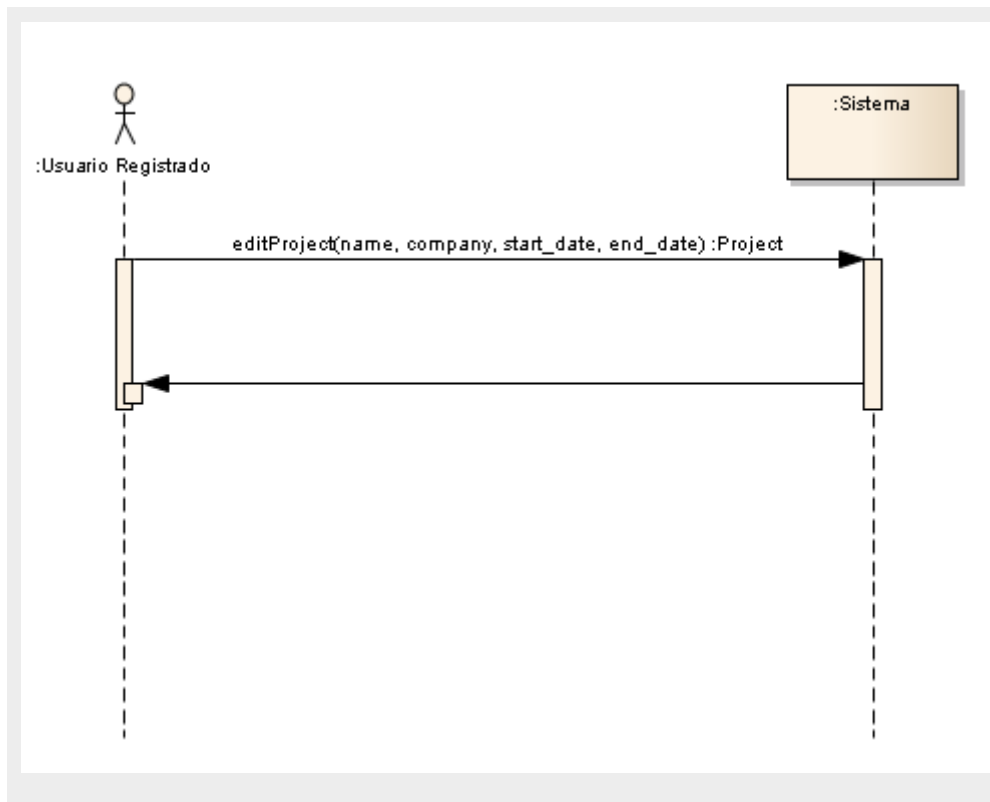
### 3. Cerrar sesión



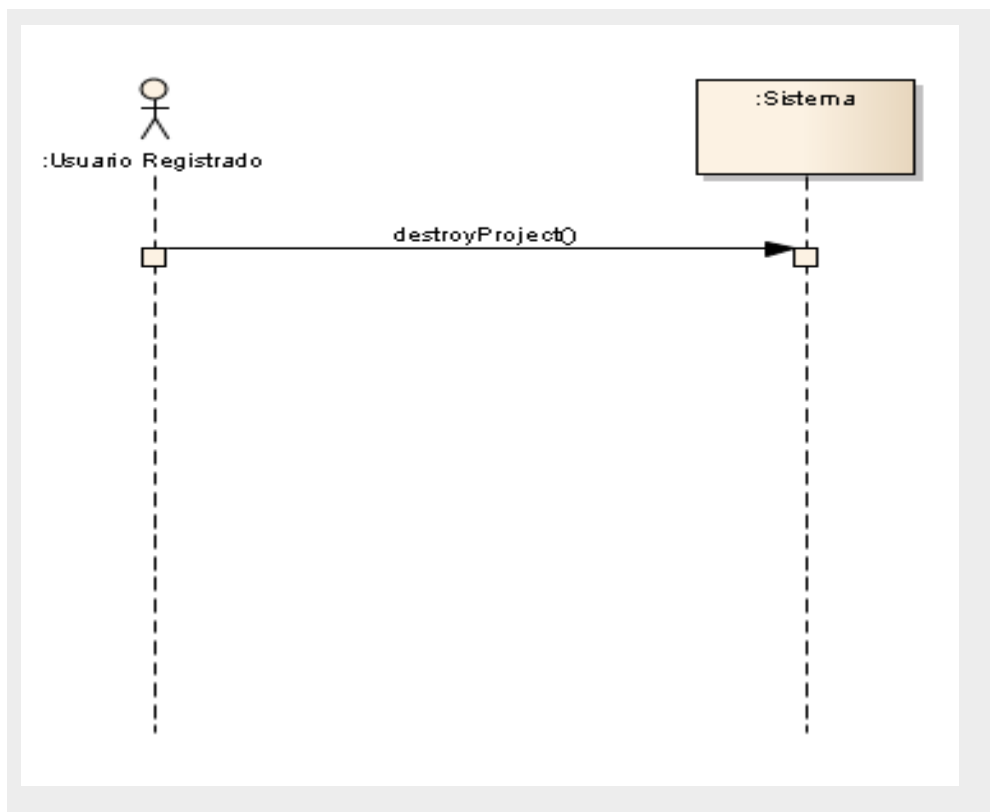
### 4. Crear nuevo proyecto



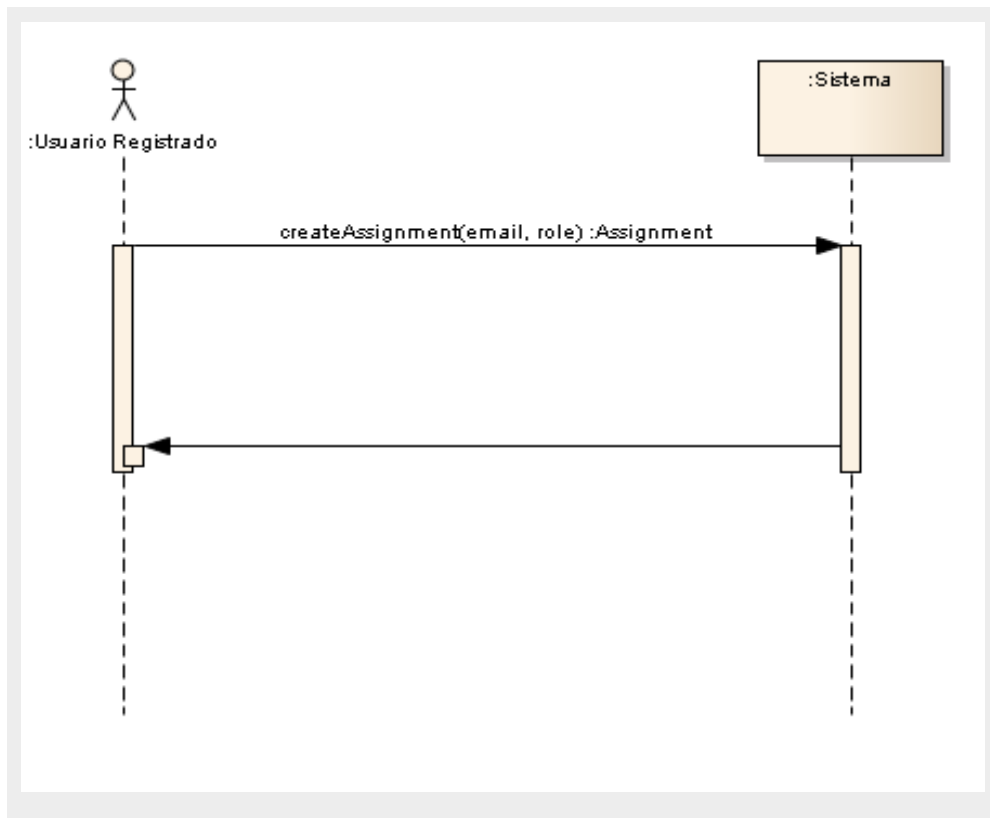
## 5. Editar proyecto



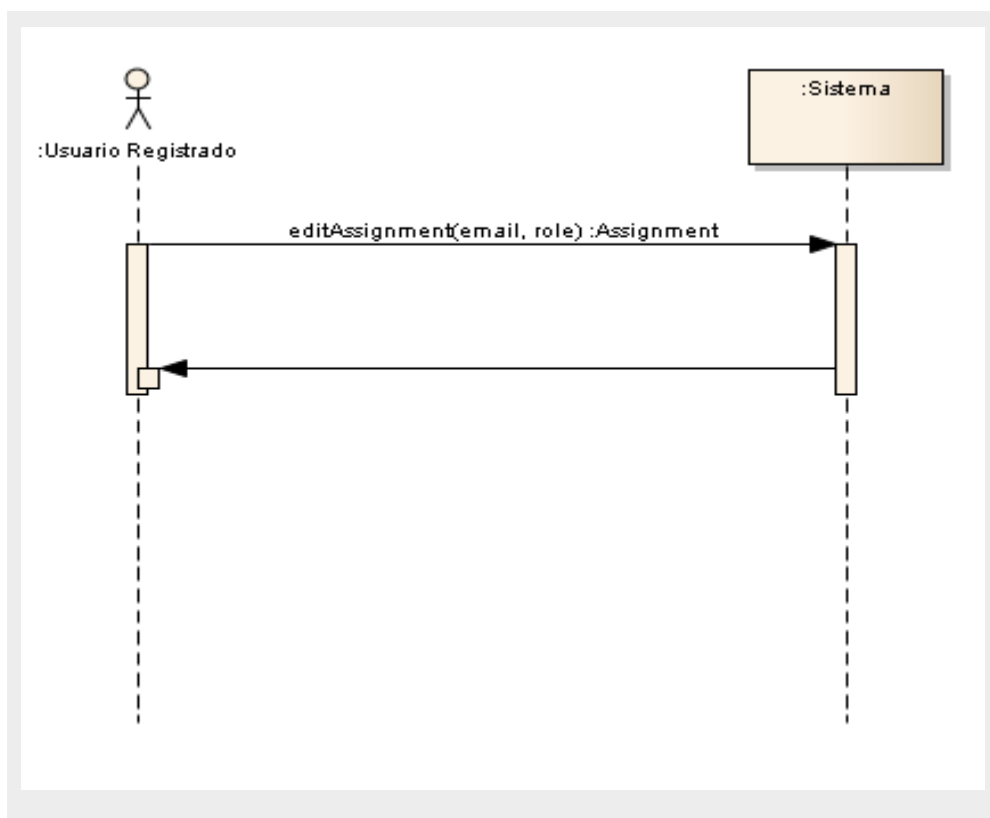
## 6. Eliminar proyecto



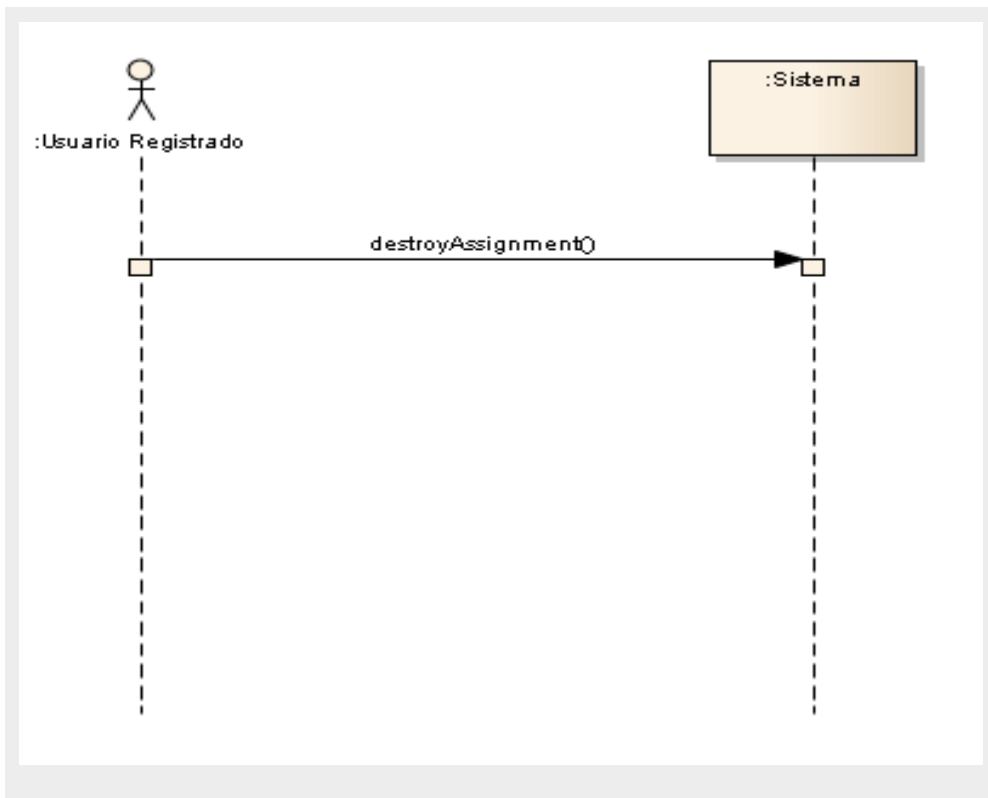
## 7. Asignar miembro a un proyecto



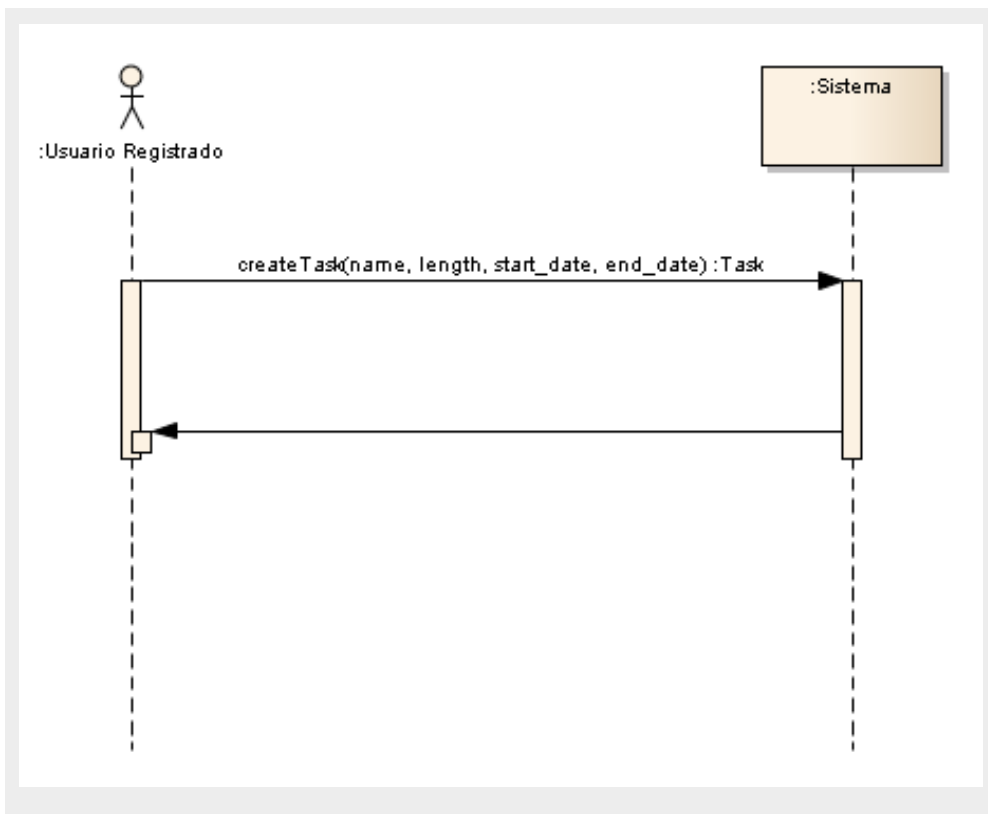
## 8. Editar miembro de un proyecto



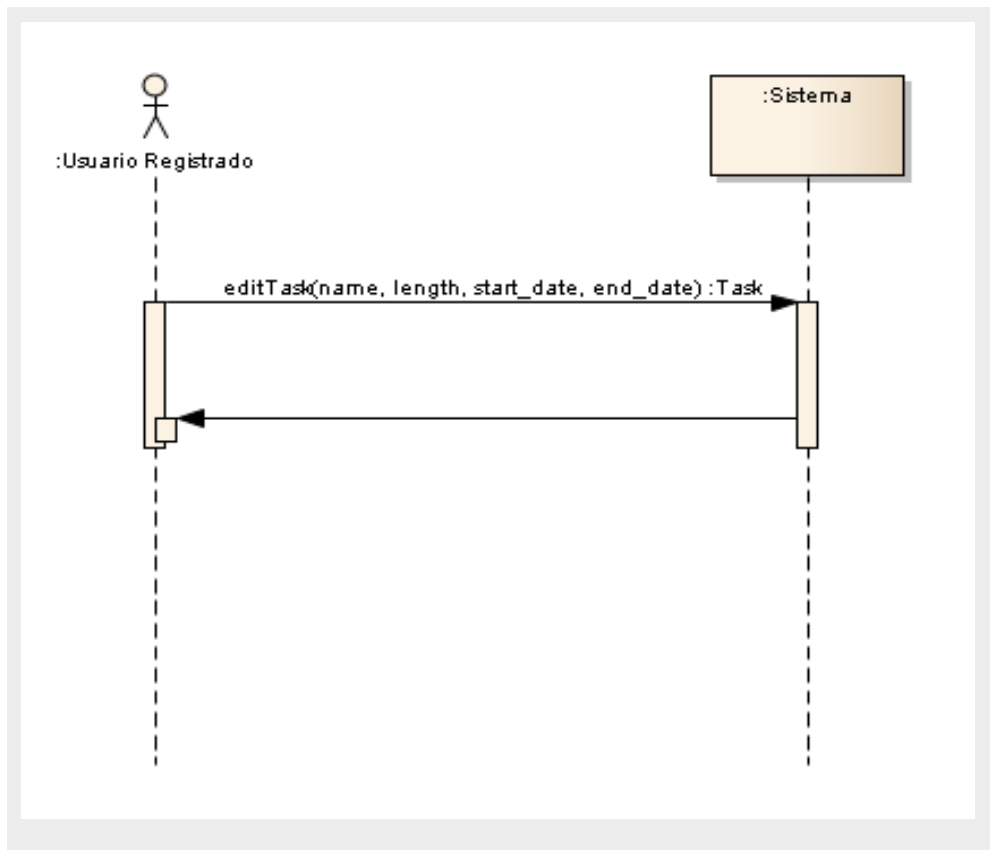
## 9. Eliminar miembro de un proyecto



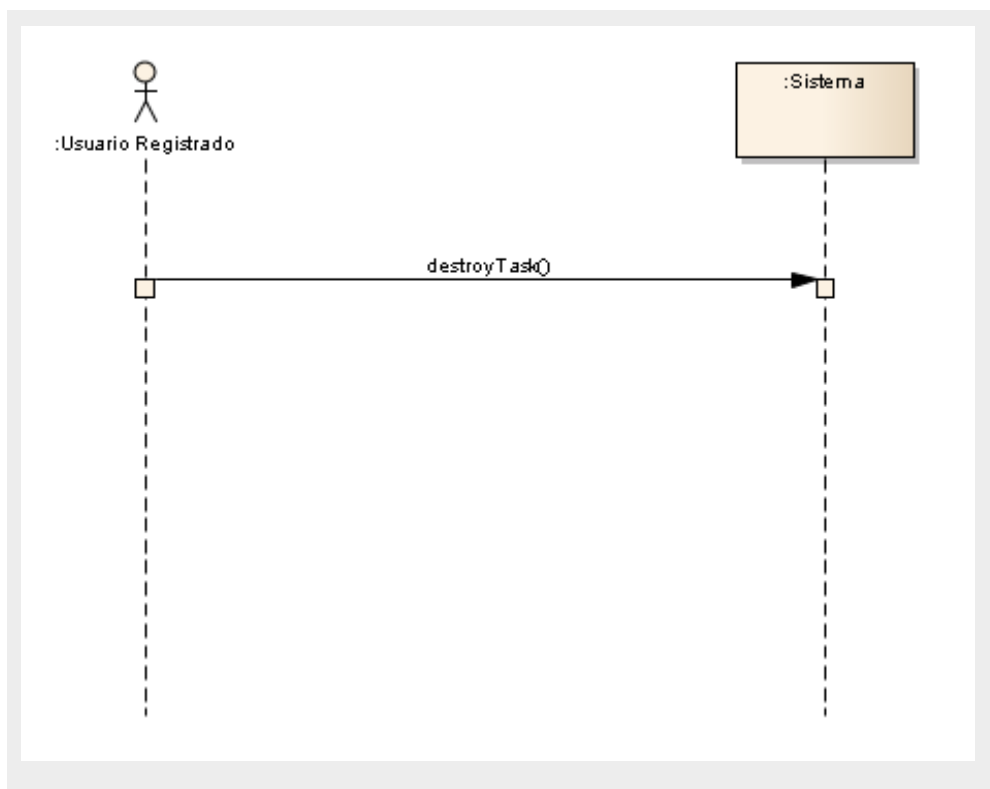
## 10. Crear nueva tarea



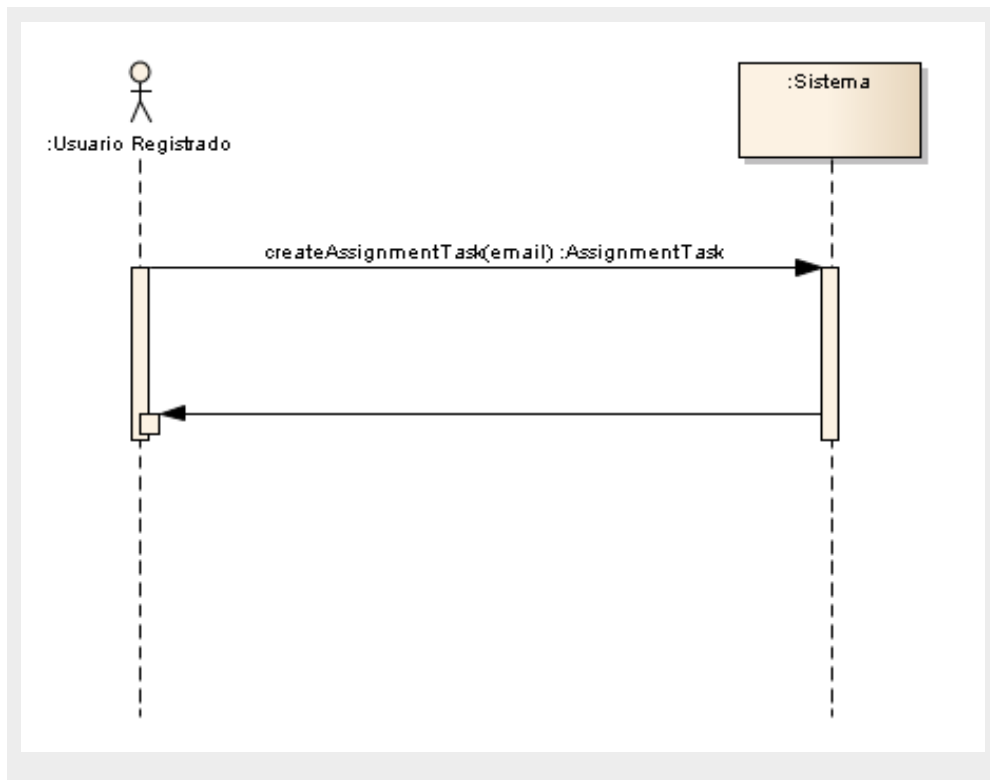
## 11. Editar tarea



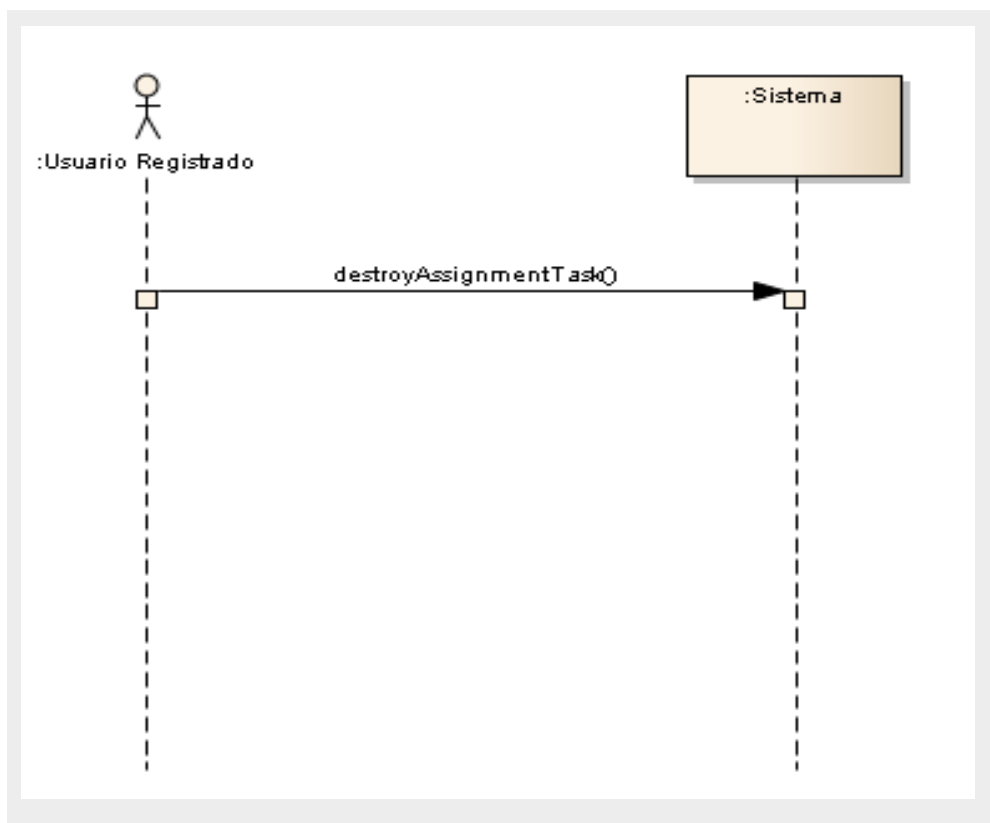
## 12. Eliminar Tarea



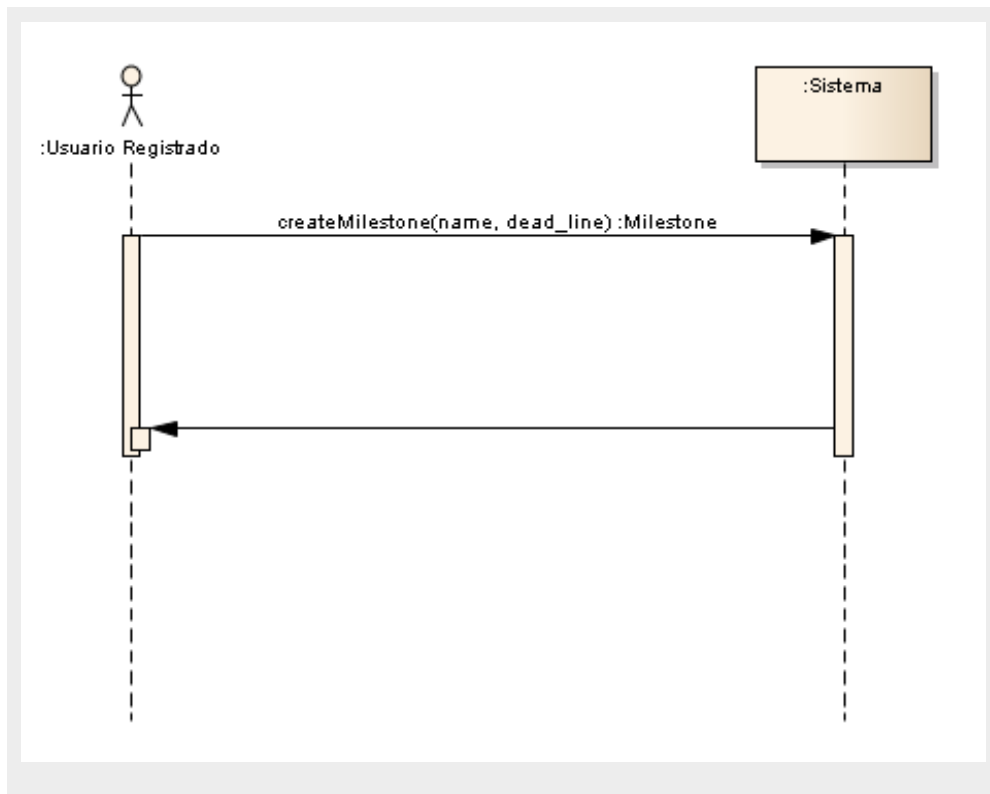
### 13. Asignar miembro a una tarea



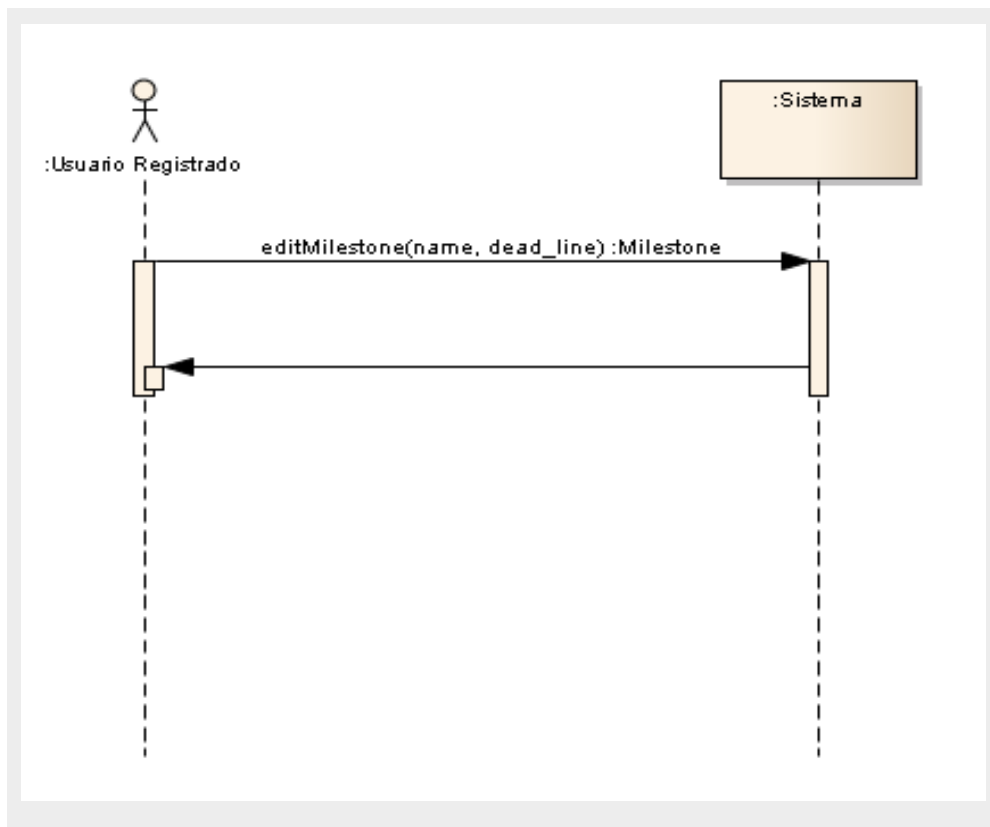
### 14. Eliminar miembro de una tarea



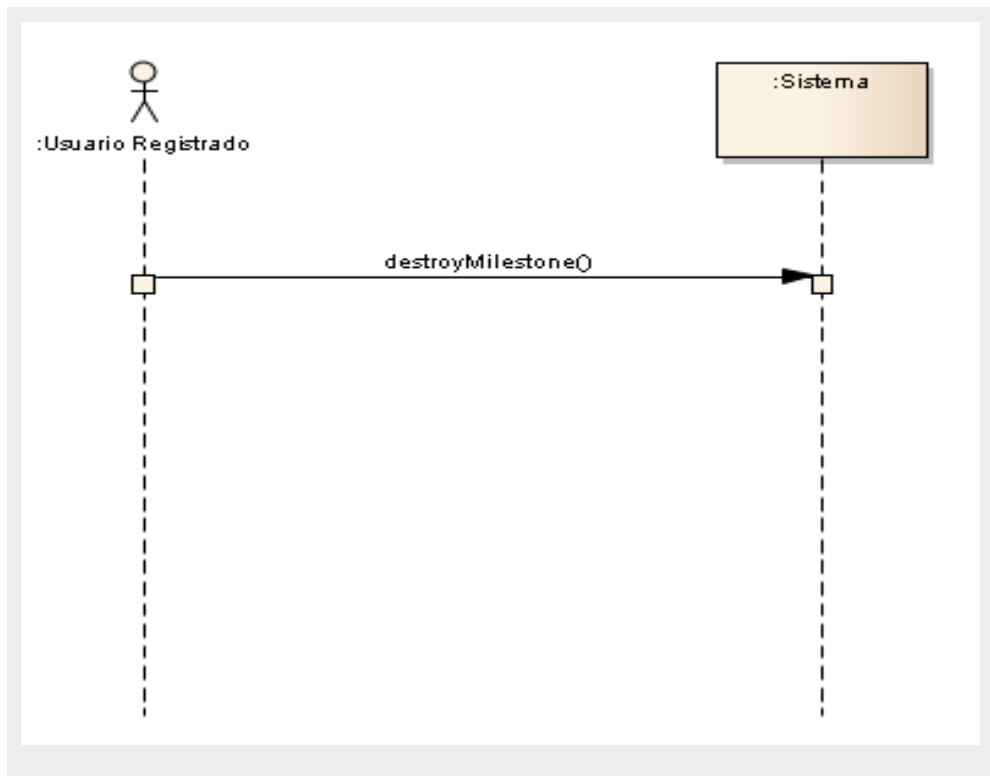
## 15. Crear nuevo hito



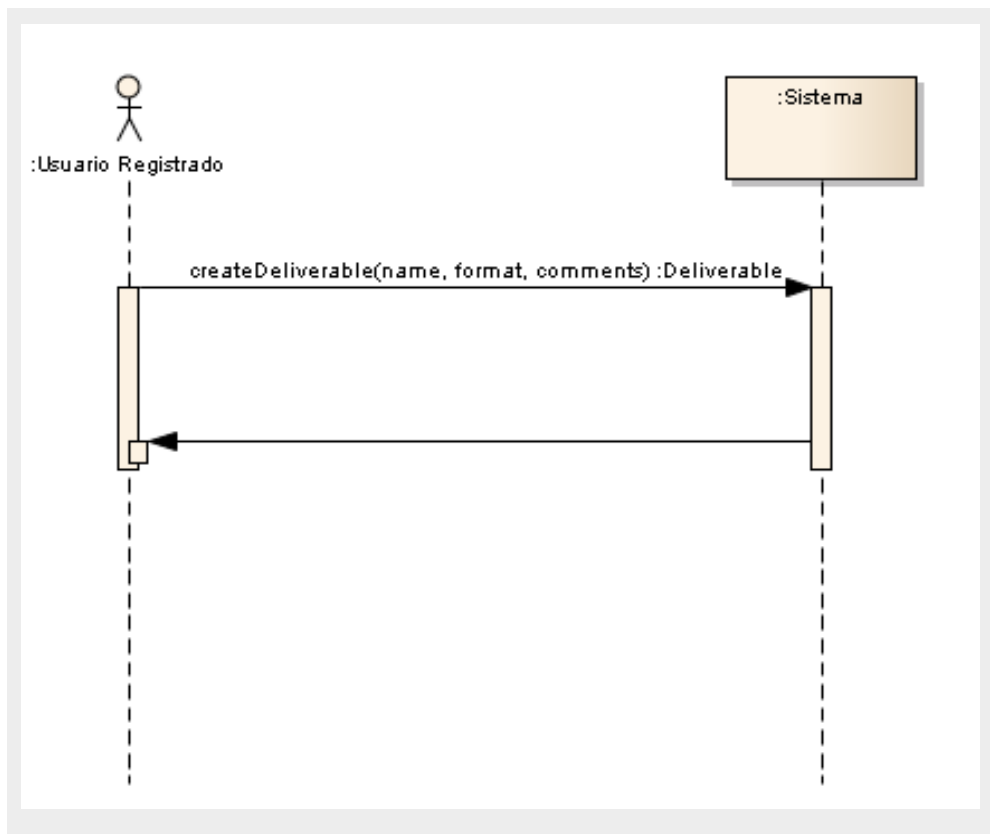
## 16. Editar hito



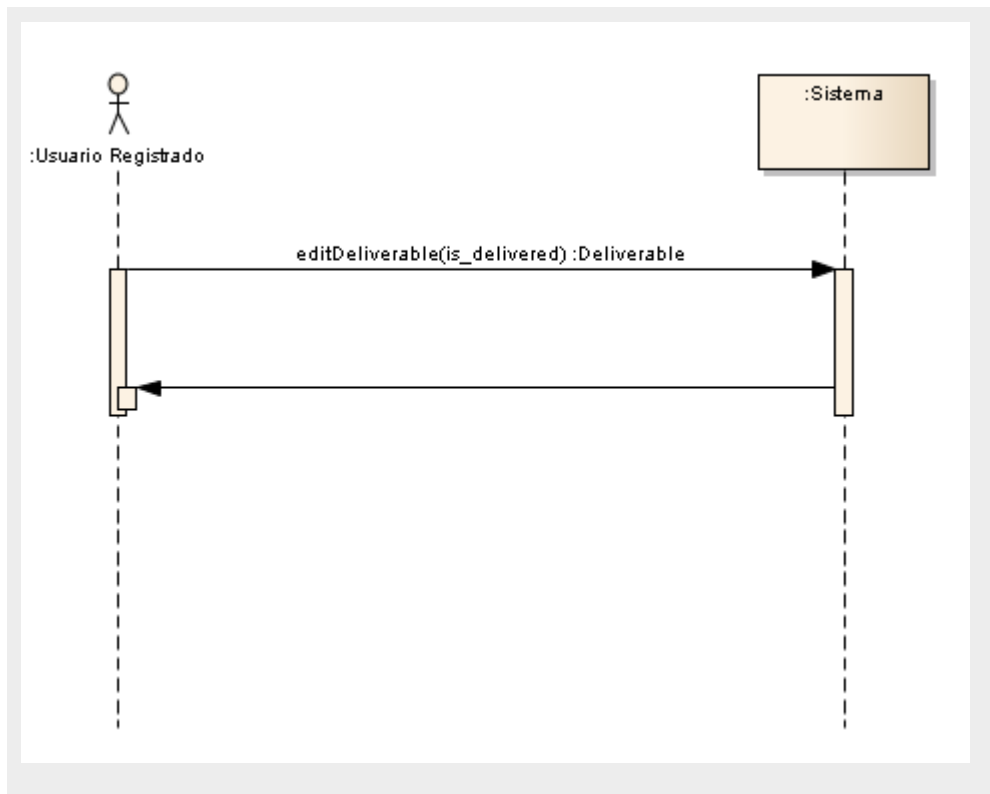
## 17. Eliminar hito



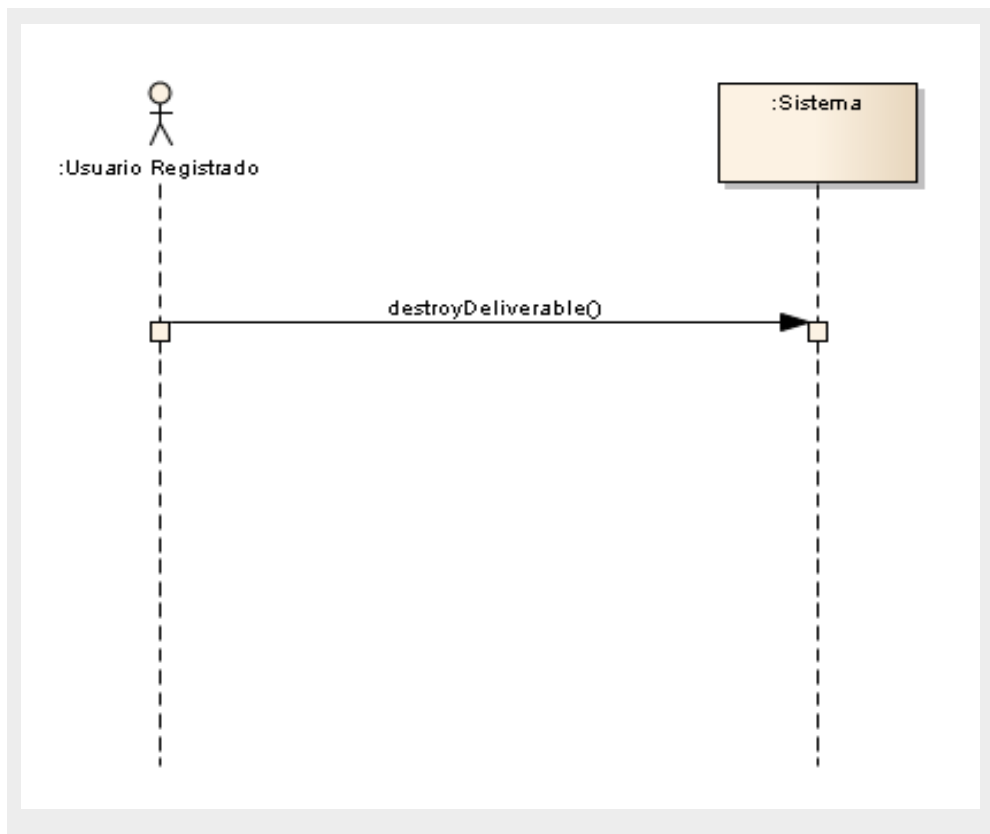
## 18. Crear nuevo entregable



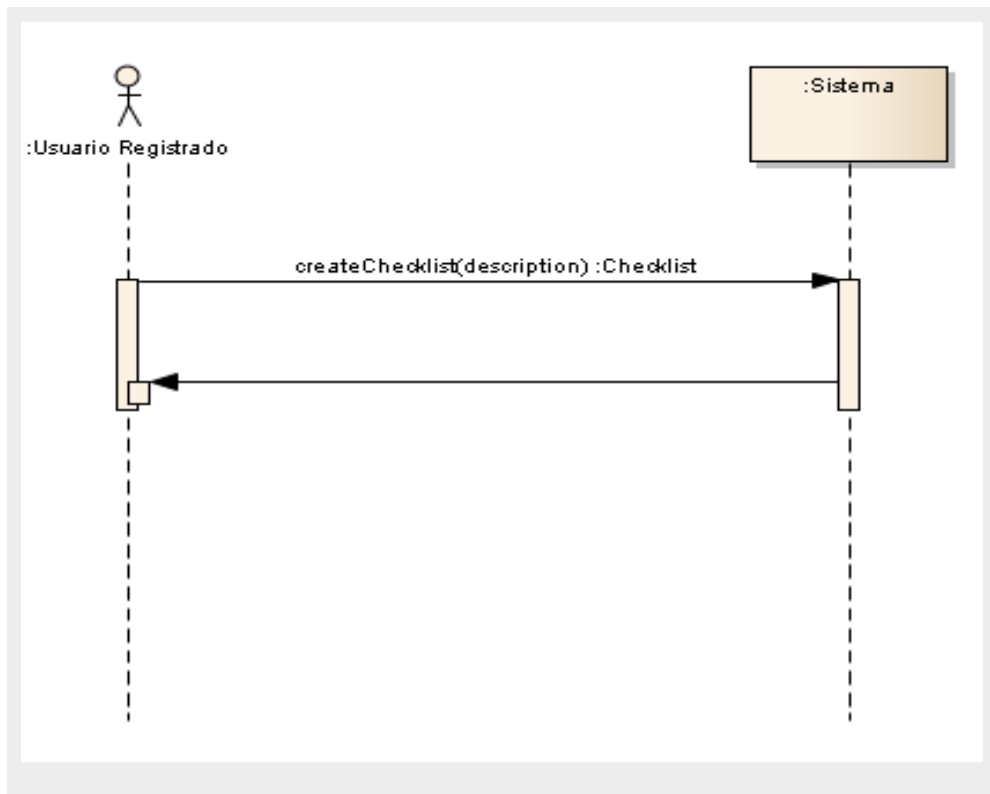
## 19. Editar estado entregable



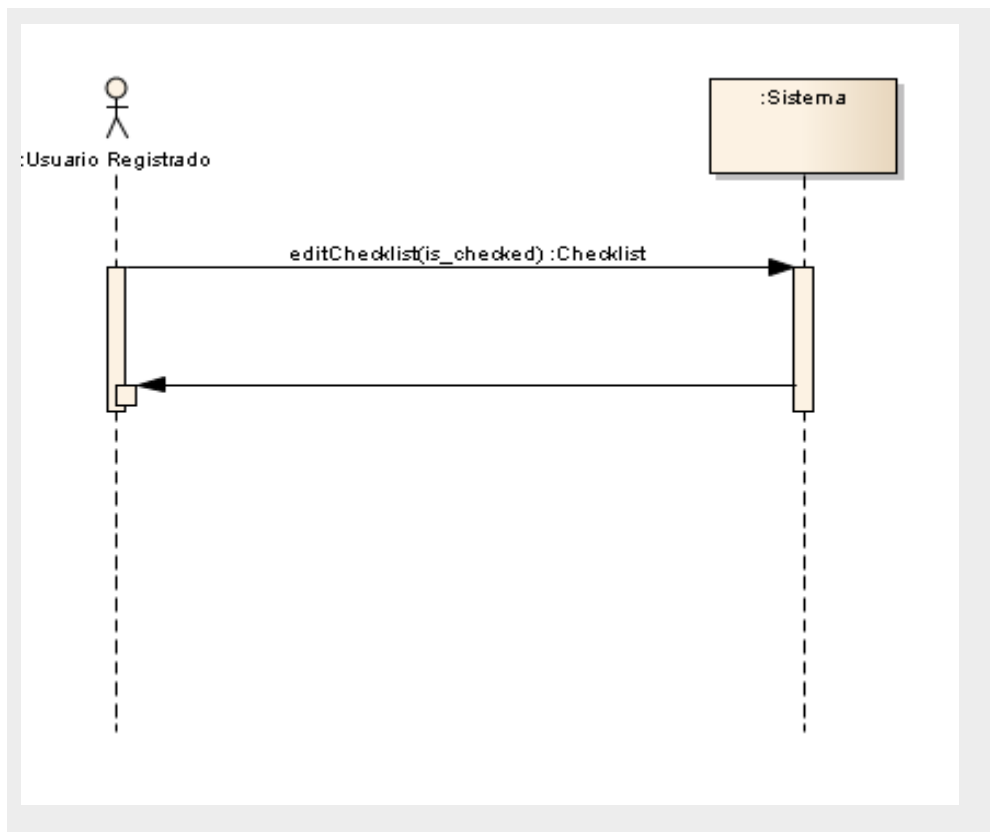
## 20. Eliminar entregable



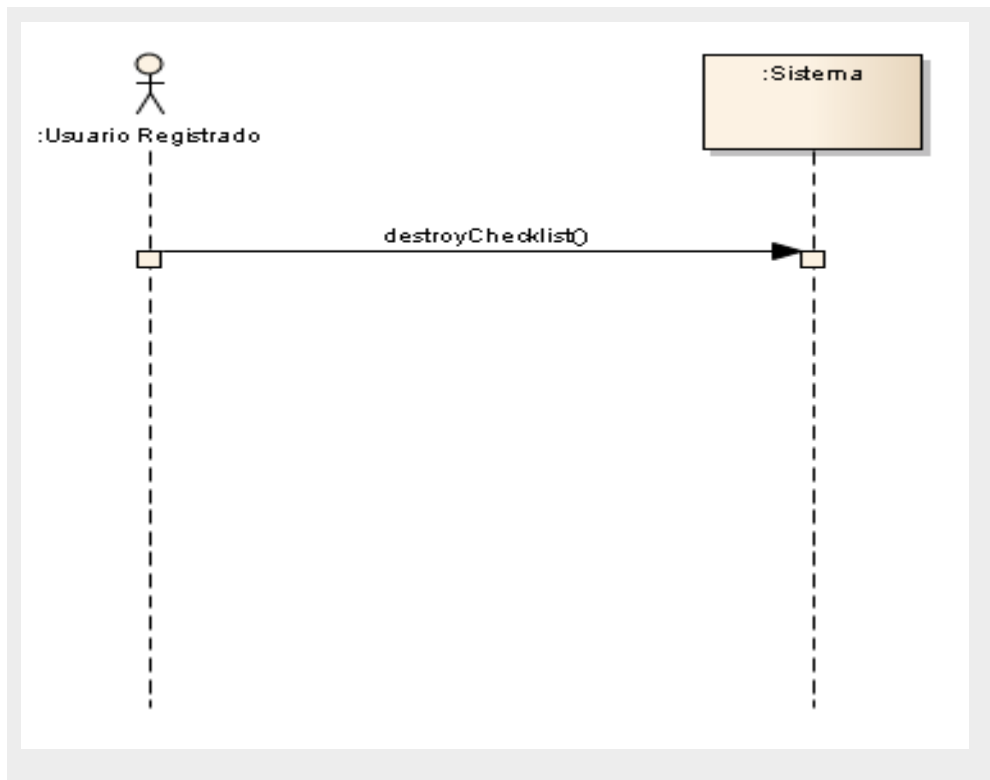
## 21. Crear nuevo checklist



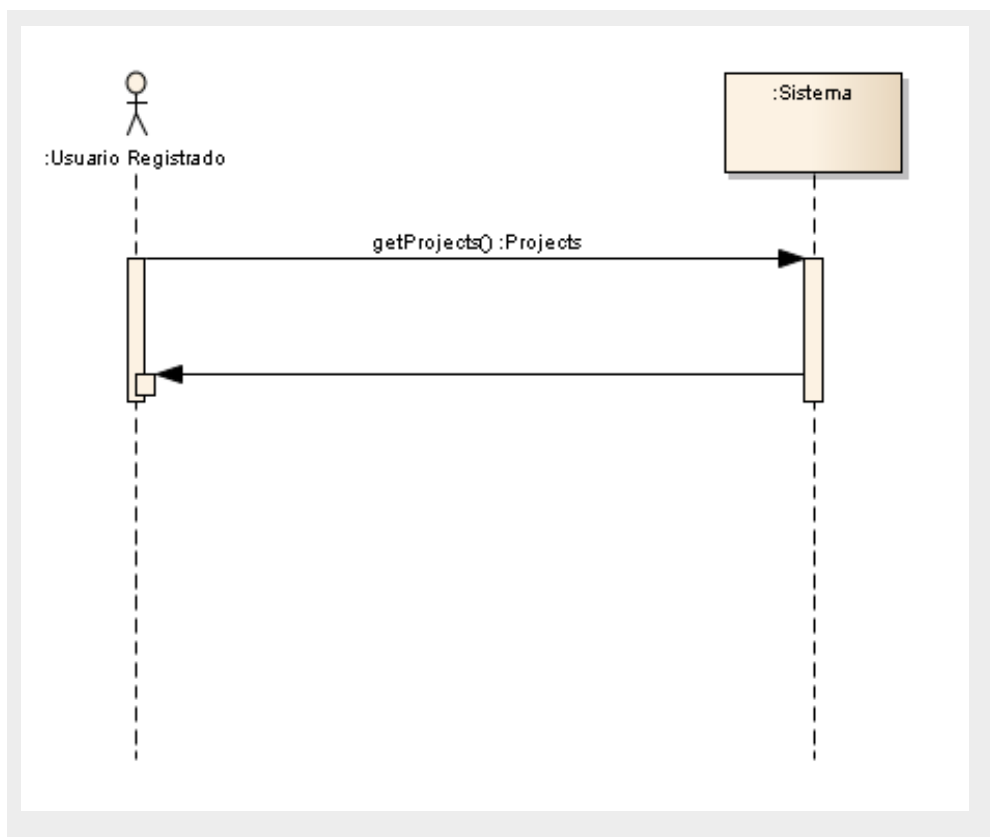
## 22. Editar estado checklist



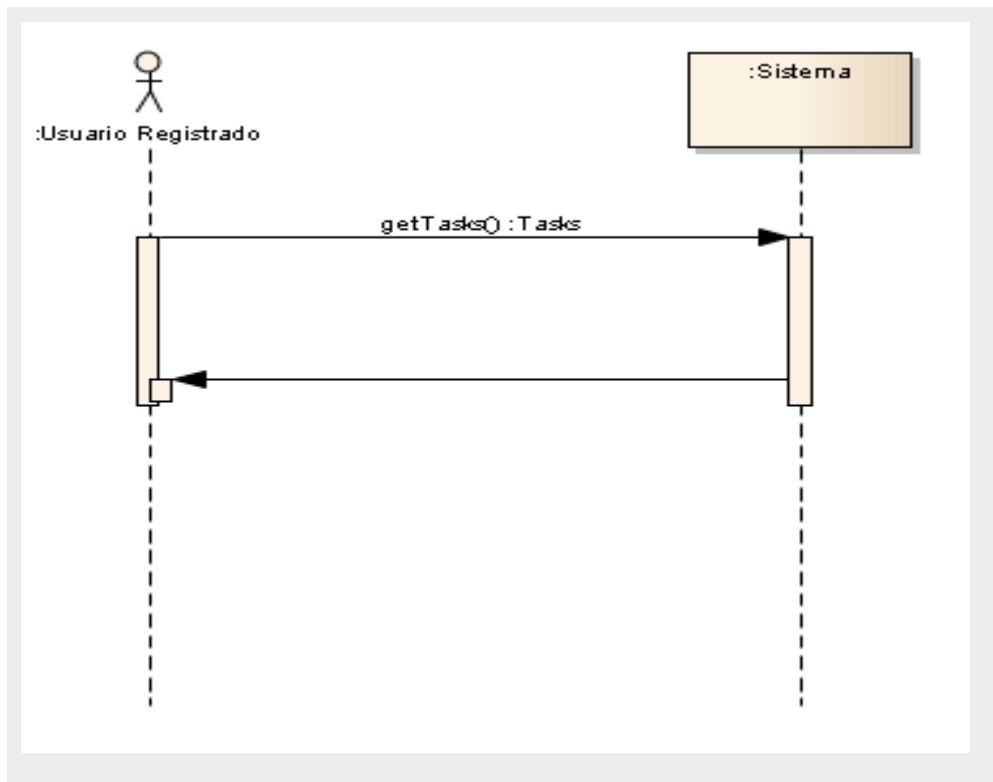
### 23. Eliminar checklist



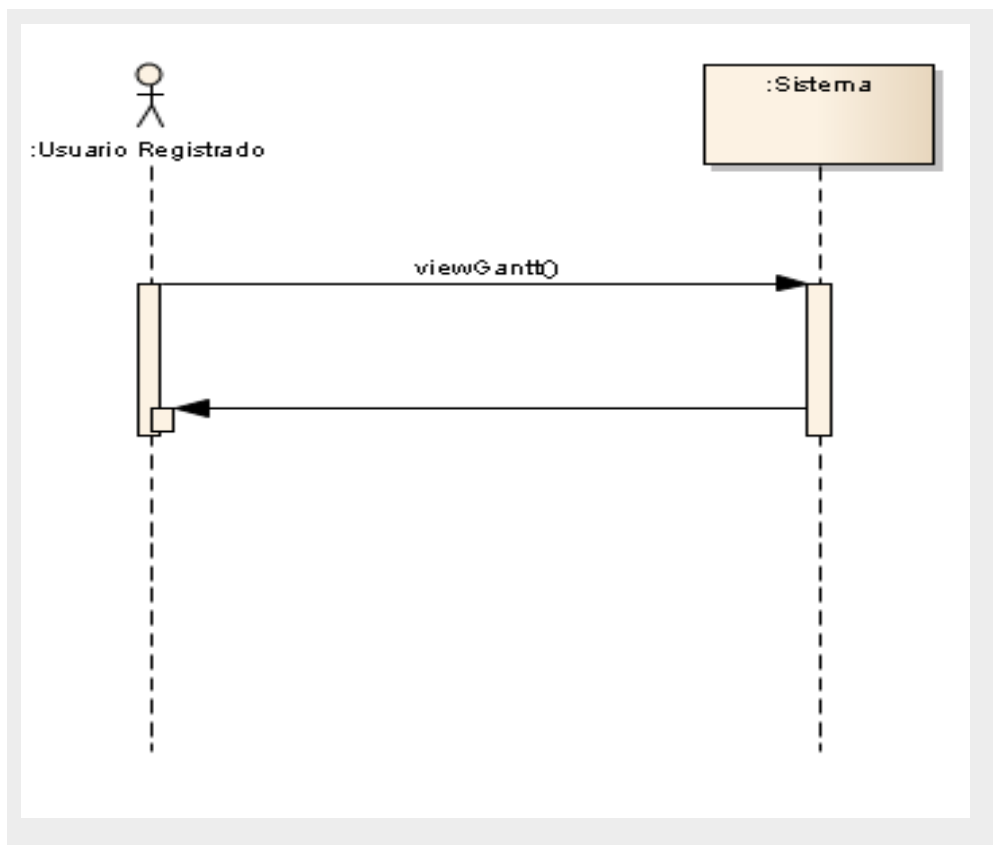
### 24. Ver proyectos



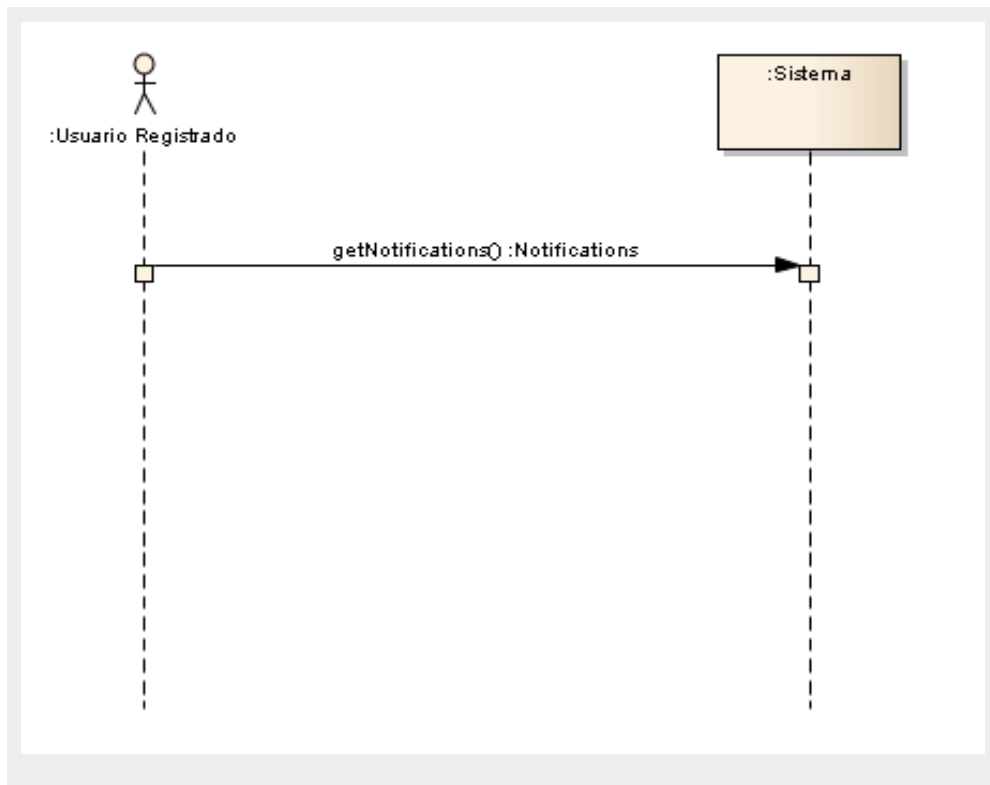
## 25. Ver tareas



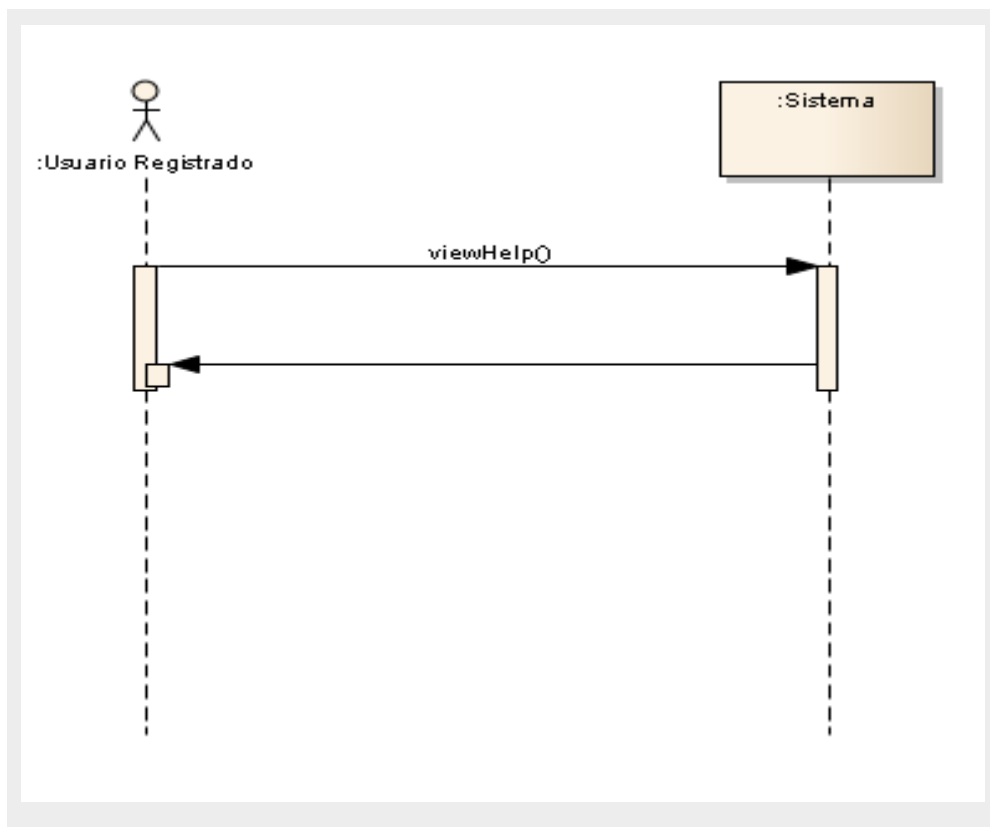
## 26. Ver diagrama de Gantt



## 27. Recibir notificaciones



## 28. Ver ayuda



## 9.2 Requisitos no funcionales

### 9.2.1 Requisitos de percepción

#### 9.2.1.1 *Requisitos de apariencia*

<b>Requisito:</b>	<b>101</b>	<b>Tipo de requisito:</b>	<b>10a</b>	<b>Acontecimiento/BUC/PUC#</b>	<b>Todos</b>
<b>Descripción:</b>	El sistema ha de ser atractivo a los usuarios.				
<b>Justificación:</b>	Los usuarios han de ver un sistema agradable para incitar a su uso.				
<b>Autor:</b>	Usuarios.				
<b>Criterio de satisfacción:</b>	Se realiza una encuesta donde más del 75% de los usuarios afirman que les gusta la apariencia.				
<b>Satisfacción del cliente:</b>	3	<b>Insatisfacción del cliente:</b>	5		
<b>Prioridad:</b>	Alta	<b>Conflictos:</b>			
<b>Materiales de soporte:</b>					
<b>Historial:</b>	07/04/2014 – Se define el requisito				

#### 9.2.1.2 *Requisitos de estilo*

<b>Requisito:</b>	<b>102</b>	<b>Tipo de requisito:</b>	<b>10b</b>	<b>Acontecimiento/BUC/PUC#</b>	<b>Todos</b>
<b>Descripción:</b>	El sistema ha de ser moderno, simple y agradable a la vista.				
<b>Justificación:</b>	Los usuarios han de ver un sistema agradable a la vista para incitar a su uso.				
<b>Autor:</b>	Usuarios.				
<b>Criterio de satisfacción:</b>	Se realiza una encuesta donde más del 75% de los usuarios afirman que el sistema es agradable a la vista, simple y moderno				

Satisfacción del cliente:	3	Insatisfacción del cliente:	5
Prioridad:	Alta	Conflictos:	
Materiales de soporte:			
Historial:	07/04/2014 – Se define el requisito		

## 9.2.2 Requisitos de capacidad de uso y humanidad

### 9.2.2.1 Requisitos de facilidad de utilización

Requisito:	<b>111</b>	Tipo de requisito:	<b>11a</b>	Acontecimiento/BUC/PUC#	<b>Todos</b>
Descripción:	El sistema ha de ser usable para los usuarios.				
Justificación:	Los usuarios tienen que aprender fácilmente a usar el sistema.				
Autor:	Usuarios.				
Criterio de satisfacción:	Se realiza una encuesta donde más del 75% de los usuarios afirman que el sistema es fácil de utilizar.				
Satisfacción del cliente:	4	Insatisfacción del cliente:	5		
Prioridad:	Alta	Conflictos:			
Materiales de soporte:					
Historial:	07/04/2014 – Se define el requisito				

### 9.2.2.2 Requisitos de personalización e internacionalización

Requisito:	<b>112</b>	Tipo de requisito:	<b>11b</b>	Acontecimiento/BUC/PUC#	<b>Todos</b>
------------	------------	--------------------	------------	-------------------------	--------------

<b>Descripción:</b>	El sistema ha de tener varios idiomas disponibles.		
<b>Justificación:</b>	Los usuarios necesitan poder escoger un idioma en el que se sientan cómodos.		
<b>Autor:</b>	Usuarios.		
<b>Criterio de satisfacción:</b>	Todo el contenido podrá ser visto en todos los idiomas disponibles por la aplicación.		
<b>Satisfacción del cliente:</b>	4	<b>Insatisfacción del cliente:</b>	3
<b>Prioridad:</b>	Baja	<b>Conflictos:</b>	
<b>Materiales de soporte:</b>			
<b>Historial:</b>	07/04/2014 – Se define el requisito		

#### 9.2.2.3 Requisitos de aprendizaje

<b>Requisito:</b>	<b>113</b>	<b>Tipo de requisito:</b>	<b>11c</b>	<b>Acontecimiento/BUC/PUC#</b>	<b>Todos</b>
<b>Descripción:</b>	El sistema puede usarse sin recibir formación previa.				
<b>Justificación:</b>	Los usuarios no tienen que necesitar formación previa para usar el sistema, esta decisión hace que el sistema sea más atractivo.				
<b>Autor:</b>	Usuarios.				
<b>Criterio de satisfacción:</b>	Se realizaran pruebas con personas que no hayan utilizado el sistema antes y más de un 75% sabrán usarlo.				
<b>Satisfacción del cliente:</b>	1	<b>Insatisfacción del cliente:</b>	5		
<b>Prioridad:</b>	Alta	<b>Conflictos:</b>			

<b>Materiales de soporte:</b>	
<b>Historial:</b>	07/04/2014 – Se define el requisito

*9.2.2.4 Requisitos de comprensión y cortesía*

<b>Requisito:</b>	<b>114</b>	<b>Tipo de requisito:</b>	<b>11d</b>	<b>Acontecimiento/BUC/PUC#</b>	<b>Todos</b>
<b>Descripción:</b>	El sistema tiene que tener un vocabulario adecuado.				
<b>Justificación:</b>	Los usuarios deben ver un lenguaje agradable.				
<b>Autor:</b>	Usuarios.				
<b>Criterio de satisfacción:</b>					
<b>Satisfacción del cliente:</b>	2	<b>Insatisfacción del cliente:</b>	3		
<b>Prioridad:</b>	Media	<b>Conflictos:</b>			
<b>Materiales de soporte:</b>					
<b>Historial:</b>	07/04/2014 – Se define el requisito				

*9.2.2.5 Requisitos de accesibilidad*

<b>Requisito:</b>	<b>115</b>	<b>Tipo de requisito:</b>	<b>11e</b>	<b>Acontecimiento/BUC/PUC#</b>	<b>Todos</b>
<b>Descripción:</b>	El sistema se adapta correctamente a la pantalla si las fuentes se engrandecen				
<b>Justificación:</b>	Los usuarios deben poder leer cualquier texto del sistema en una fuente agradable para su visión.				
<b>Autor:</b>	Usuarios.				

<b>Criterio de satisfacción:</b>	El sistema se adapta al tamaño de pantalla.		
<b>Satisfacción del cliente:</b>	2	<b>Insatisfacción del cliente:</b>	3
<b>Prioridad:</b>	Baja	<b>Conflictos:</b>	
<b>Materiales de soporte:</b>			
<b>Historial:</b>	07/04/2014 – Se define el requisito		

### 9.2.3 Requisitos de desempeño

#### 9.2.3.1 Requisitos de velocidad y latencia

<b>Requisito:</b>	<b>121</b>	<b>Tipo de requisito:</b>	<b>12a</b>	<b>Acontecimiento/BUC/PUC#</b>	<b>Todos</b>
<b>Descripción:</b>	El sistema mostrará las modificaciones que afecten a los usuarios en el menor tiempo posible.				
<b>Justificación:</b>	Los usuarios deben poder estar al día de cualquiera de sus proyectos y tener un feedback lo más rápido posible.				
<b>Autor:</b>	Usuarios.				
<b>Criterio de satisfacción:</b>	El sistema tardara menos de 5 minutos en notificar al usuario las novedades que hay en sus proyectos.				
<b>Satisfacción del cliente:</b>	3	<b>Insatisfacción del cliente:</b>	4		
<b>Prioridad:</b>	Media	<b>Conflictos:</b>			
<b>Materiales de soporte:</b>					

<b>Historial:</b>	07/04/2014 – Se define el requisito
-------------------	-------------------------------------

*9.2.3.2 Requisitos de seguridad*

<b>Requisito:</b>	<b>122</b>	<b>Tipo de requisito:</b>	<b>12b</b>	<b>Acontecimiento/BUC/PUC#</b>	<b>Todos</b>
<b>Descripción:</b>	El sistema guardara todas las contraseñas de los usuarios encriptadas.				
<b>Justificación:</b>	Los usuarios deben sentirse seguros al usar el sistema y saber que sus datos no pueden ser consultados por cualquiera.				
<b>Autor:</b>	Usuarios.				
<b>Criterio de satisfacción:</b>	Auto explicativo				
<b>Satisfacción del cliente:</b>	2	<b>Insatisfacción del cliente:</b>	3		
<b>Prioridad:</b>	Media	<b>Conflictos:</b>			
<b>Materiales de soporte:</b>					
<b>Historial:</b>	07/04/2014 – Se define el requisito				

*9.2.3.3 Requisitos de precisión o exactitud*

<b>Requisito:</b>	<b>123</b>	<b>Tipo de requisito:</b>	<b>12c</b>	<b>Acontecimiento/BUC/PUC#</b>	<b>Todos</b>
<b>Descripción:</b>	La unidad de medida de tiempo del sistema serán siempre horas.				
<b>Justificación:</b>	La simplificación de usar horas es debido a que no tiene mucho sentido planificar tareas de menos de 1 hora.				
<b>Autor:</b>	Usuarios.				
<b>Criterio de satisfacción:</b>	Auto explicativo				

Satisfacción del cliente:	1	Insatisfacción del cliente:	2
Prioridad:	Alta	Conflictos:	
Materiales de soporte:			
Historial:	07/04/2014 – Se define el requisito		

Requisito:	<b>124</b>	Tipo de requisito:	<b>12c</b>	Acontecimiento/BUC/PUC#	<b>Todos</b>
Descripción:	El formato de las fechas será DD/MM/YYYY				
Justificación:	El sistema ha de tener una manera univoca de representar las fechas para no causar dudas a los usuarios.				
Autor:	Usuarios.				
Criterio de satisfacción:	Auto explicativo				
Satisfacción del cliente:	1	Insatisfacción del cliente:	2		
Prioridad:	Alta	Conflictos:			
Materiales de soporte:					
Historial:	07/04/2014 – Se define el requisito				

#### *9.2.3.4 Requisitos de confiabilidad y disponibilidad*

Requisito:	<b>125</b>	Tipo de requisito:	<b>12d</b>	Acontecimiento/BUC/PUC#	<b>Todos</b>
------------	------------	--------------------	------------	-------------------------	--------------

<b>Descripción:</b>	El sistema ha de estar disponible 24/7 los 365 días del año.		
<b>Justificación:</b>	Los usuarios han de poder acceder siempre al sistema.		
<b>Autor:</b>	Usuarios.		
<b>Criterio de satisfacción:</b>	Es sistema estará disponible, como mínimo, el 99% del tiempo.		
<b>Satisfacción del cliente:</b>	1	<b>Insatisfacción del cliente:</b>	5
<b>Prioridad:</b>	Alta	<b>Conflictos:</b>	
<b>Materiales de soporte:</b>			
<b>Historial:</b>	07/04/2014 – Se define el requisito		

#### *9.2.3.5 Requisitos de fortaleza o tolerancia a fallas*

El sistema no tiene ningún requisito de fortaleza o tolerancia a fallos

#### *9.2.3.6 Requisitos de capacidad*

<b>Requisito:</b>	<b>125</b>	<b>Tipo de requisito:</b>	<b>12f</b>	<b>Acontecimiento/BUC/PUC#</b>	<b>Todos</b>
<b>Descripción:</b>	El sistema ha de soportar como mínimo 1000 usuarios en un mismo momento.				
<b>Justificación:</b>	El sistema necesita de una gran comunidad de usuarios para su rentabilidad, por tanto, se den muchos momentos de trafico alto en los servidores.				
<b>Autor:</b>	Usuarios.				
<b>Criterio de satisfacción:</b>	Auto explicativo				
<b>Satisfacción del cliente:</b>	1	<b>Insatisfacción del cliente:</b>	4		

<b>Prioridad:</b>	Media	<b>Conflictos:</b>	
<b>Materiales de soporte:</b>			
<b>Historial:</b>	07/04/2014 – Se define el requisito		

#### 9.2.3.7 Requisitos de expansión o crecimiento

<b>Requisito:</b>	<b>126</b>	<b>Tipo de requisito:</b>	<b>12g</b>	<b>Acontecimiento/BUC/PUC#</b>	<b>Todos</b>
<b>Descripción:</b>	El sistema ha de ser escalable en número de usuarios y funcionalidades.				
<b>Justificación:</b>	Los usuarios necesitaran más funcionalidades y si crece la comunidad se necesitara escalabilidad.				
<b>Autor:</b>	Usuarios.				
<b>Criterio de satisfacción:</b>					
<b>Satisfacción del cliente:</b>	1	<b>Insatisfacción del cliente:</b>	2		
<b>Prioridad:</b>	Media	<b>Conflictos:</b>			
<b>Materiales de soporte:</b>					
<b>Historial:</b>	07/04/2014 – Se define el requisito				

#### 9.2.3.8 Requisitos de longevidad

<b>Requisito:</b>	<b>127</b>	<b>Tipo de requisito:</b>	<b>12g</b>	<b>Acontecimiento/BUC/PUC#</b>	<b>Todos</b>
<b>Descripción:</b>	El sistema ha de ser sostenible económicamente con la inversión inicial al menos 5 años				

Justificación:	Necesito asegurarme la viabilidad de mi proyecto por tal de recuperar la inversión inicial		
Autor:	Autor		
Criterio de satisfacción:	Auto explicativo		
Satisfacción del cliente:	1	Insatisfacción del cliente:	1
Prioridad:	Media	Conflictos:	
Materiales de soporte:			
Historial:	07/04/2014 – Se define el requisito		

#### 9.2.4 Requisitos operacionales y ambientales

##### 9.2.4.1 Ambiente físico esperado

Requisito:	<b>131</b>	Tipo de requisito:	<b>13a</b>	Acontecimiento/BUC/PUC#	<b>Todos</b>
Descripción:	El sistema se alojara físicamente en los servidores de Amazon.				
Justificación:	El sistema tiene que proporcionar los requisitos funcionales previstos a través de internet.				
Autor:	Usuario.				
Criterio de satisfacción:					
Satisfacción del cliente:	1	Insatisfacción del cliente:	1		
Prioridad:	Media	Conflictos:			

Materiales de soporte:	
Historial:	07/04/2014 – Se define el requisito

#### 9.2.4.2 Requisitos de interfaz con sistemas adyacentes

Requisito:	<b>132</b>	Tipo de requisito:	<b>13b</b>	Acontecimiento/BUC/PUC#	<b>Todos</b>
Descripción:	El sistema será crossplatform, es decir, se podrá acceder desde móviles, tablets, portátiles etc...				
Justificación:	El usuario deseará consultar sus proyectos desde todos sus dispositivos con acceso a internet, por tanto, hay que satisfacer esa necesidad que también ha sido definida como objetivo.				
Autor:	Usuario.				
Criterio de satisfacción:					
Satisfacción del cliente:	4	Insatisfacción del cliente:	4		
Prioridad:	Alta	Conflictos:			
Materiales de soporte:					
Historial:	07/04/2014 – Se define el requisito				

#### 9.2.4.3 Requisitos de producción

Requisito:	<b>133</b>	Tipo de requisito:	<b>13c</b>	Acontecimiento/BUC/PUC#	<b>Todos</b>
Descripción:	El sistema recogerá datos estadísticos sobre el uso del sistema.				

Justificación:	Los anunciantes necesitaran estadísticas para valorar el impacto de su publicidad.		
Autor:	Anunciante		
Criterio de satisfacción:			
Satisfacción del cliente:	2	Insatisfacción del cliente:	4
Prioridad:	Alta	Conflictos:	
Materiales de soporte:			
Historial:	07/04/2014 – Se define el requisito		

#### 9.2.4.4 Requisitos de lanzamiento

Requisito:	<b>134</b>	Tipo de requisito:	<b>13d</b>	Acontecimiento/BUC/PUC#	<b>Todos</b>
Descripción:	El sistema no causara fallos de características previas cuando haya actualizaciones.				
Justificación:	Los usuarios no perderán funcionalidades cuando se realice una actualización.				
Autor:	Usuarios				
Criterio de satisfacción:					
Satisfacción del cliente:	2	Insatisfacción del cliente:	4		
Prioridad:	Baja	Conflictos:			

<b>Materiales de soporte:</b>	
<b>Historial:</b>	07/04/2014 – Se define el requisito

## 9.2.5 Requisitos de preservación y soporte

### 9.2.5.1 Requisitos de mantenimiento

<b>Requisito:</b>	<b>141</b>	<b>Tipo de requisito:</b>	<b>14a</b>	<b>Acontecimiento/BUC/PUC#</b>	<b>Todos</b>
<b>Descripción:</b>	El sistema ha de ser susceptible a cambios de tecnología o a nuevas funcionalidades.				
<b>Justificación:</b>	El sistema se ha de ir actualizando a las nuevas tecnologías por tal de no quedar desfasado.				
<b>Autor:</b>	Usuarios				
<b>Criterio de satisfacción:</b>	El sistema dispone de documentación necesaria con toda la información sobre la tecnología usada para poder ser actualizado.				
<b>Satisfacción del cliente:</b>	1	<b>Insatisfacción del cliente:</b>	4		
<b>Prioridad:</b>	Media	<b>Conflictos:</b>			
<b>Materiales de soporte:</b>					
<b>Historial:</b>	07/04/2014 – Se define el requisito				

### 9.2.5.2 Requisitos de soporte

<b>Requisito:</b>	<b>142</b>	<b>Tipo de requisito:</b>	<b>14b</b>	<b>Acontecimiento/BUC/PUC#</b>	<b>Todos</b>
<b>Descripción:</b>	El sistema dispone de una página de ayuda.				

<b>Justificación:</b>	El usuario podrá consultar sus dudas acerca del funcionamiento en la página de ayuda.		
<b>Autor:</b>	Usuarios		
<b>Criterio de satisfacción:</b>	Auto explicativo		
<b>Satisfacción del cliente:</b>	1	<b>Insatisfacción del cliente:</b>	4
<b>Prioridad:</b>	Media	<b>Conflictos:</b>	
<b>Materiales de soporte:</b>			
<b>Historial:</b>	07/04/2014 – Se define el requisito		

<b>Requisito:</b>	<b>143</b>	<b>Tipo de requisito:</b>	<b>14b</b>	<b>Acontecimiento/BUC/PUC#</b>	<b>Todos</b>
<b>Descripción:</b>	El sistema dispone de un correo electrónico de ayuda.				
<b>Justificación:</b>	El usuario podrá enviar emails con sus dudas a un email del sistema que será respondido con la mayor brevedad posible.				
<b>Autor:</b>	Usuarios				
<b>Criterio de satisfacción:</b>	Auto explicativo				
<b>Satisfacción del cliente:</b>	3	<b>Insatisfacción del cliente:</b>	4		
<b>Prioridad:</b>	Media	<b>Conflictos:</b>			

Materiales de soporte:	
Historial:	07/04/2014 – Se define el requisito

### 9.2.5.3 Requisitos de adaptabilidad

Requisito:	<b>142</b>	Tipo de requisito:	<b>14b</b>	Acontecimiento/BUC/PUC#	<b>Todos</b>
Descripción:	El sistema detectará automáticamente el tamaño de pantalla del usuario y adaptara la interface				
Justificación:	El usuario podrá ver correctamente el sistema desde cualquier pantalla.				
Autor:	Usuarios				
Criterio de satisfacción:	Auto explicativo				
Satisfacción del cliente:	3	Insatisfacción del cliente:	4		
Prioridad:	Media	Conflictos:			
Materiales de soporte:					
Historial:	07/04/2014 – Se define el requisito				

### 9.2.6 Requisitos de seguridad

#### 9.2.6.1 Requisitos de acceso

Requisito:	<b>151</b>	Tipo de requisito:	<b>15a</b>	Acontecimiento/BUC/PUC#	<b>Todos</b>
Descripción:	El sistema ha de permitir el acceso a cualquier usuario a través de internet.				
Justificación:	Este criterio permite al sistema tener una presencia total.				

<b>Autor:</b>	Usuarios		
<b>Criterio de satisfacción:</b>	El sistema es accesible mediante un navegador cualquiera y conexión a internet.		
<b>Satisfacción del cliente:</b>	3	<b>Insatisfacción del cliente:</b>	4
<b>Prioridad:</b>	Media	<b>Conflictos:</b>	
<b>Materiales de soporte:</b>			
<b>Historial:</b>	07/04/2014 – Se define el requisito		

#### 9.2.6.2 Requisitos de integridad

<b>Requisito:</b>	<b>152</b>	<b>Tipo de requisito:</b>	<b>15b</b>	<b>Acontecimiento/BUC/PUC#</b>	<b>Todos</b>
<b>Descripción:</b>	El sistema ha de definir restricciones por tal de garantizar la integridad de los datos (integridad lógica), y ha de mantener también un sistema de copias de seguridad y un dietario por tal de poder re ejecutar los cambios efectuados desde la última copia de seguridad (integridad física).				
<b>Justificación:</b>	Los datos almacenados en la aplicación son la base fundamental de su funcionamiento por esa razón hay que evitar cualquier tipo de pérdida.				
<b>Autor:</b>	Usuarios				
<b>Criterio de satisfacción:</b>	El sistema realiza copias de seguridad cada día y evitará consultas de usuarios directamente a la base de datos o cualquier otro tipo de ataque.				
<b>Satisfacción del cliente:</b>	2	<b>Insatisfacción del cliente:</b>	5		
<b>Prioridad:</b>	Alta	<b>Conflictos:</b>			

<b>Materiales de soporte:</b>	
<b>Historial:</b>	07/04/2014 – Se define el requisito

#### 9.2.6.3 Requisitos de privacidad

<b>Requisito:</b>	<b>153</b>	<b>Tipo de requisito:</b>	<b>15c</b>	<b>Acontecimiento/BUC/PUC#</b>	<b>Todos</b>
<b>Descripción:</b>	El sistema informa sobre la privacidad de la aplicación y su política de datos.				
<b>Justificación:</b>	Los usuarios han de estar en todo momento al corriente de la política de datos del sistema.				
<b>Autor:</b>	Usuarios				
<b>Criterio de satisfacción:</b>	El sistema ofrece una página con toda la información necesaria en cuanto a política de privacidad y datos.				
<b>Satisfacción del cliente:</b>	3	<b>Insatisfacción del cliente:</b>	5		
<b>Prioridad:</b>	Alta	<b>Conflictos:</b>			
<b>Materiales de soporte:</b>					
<b>Historial:</b>	07/04/2014 – Se define el requisito				

#### 9.2.6.4 Requisitos de auditoria

<b>Requisito:</b>	<b>154</b>	<b>Tipo de requisito:</b>	<b>15d</b>	<b>Acontecimiento/BUC/PUC#</b>	<b>Todos</b>
<b>Descripción:</b>	El sistema guarda información acerca de los datos que se almacenan y del registro de usuarios.				
<b>Justificación:</b>	El sistema guarda todos los datos necesarios para pasar una auditoria.				
<b>Autor:</b>	Autor				

<b>Criterio de satisfacción:</b>	El sistema guarda toda la información necesaria para pasar una auditoria.		
<b>Satisfacción del cliente:</b>	5	<b>Insatisfacción del cliente:</b>	5
<b>Prioridad:</b>	Alta	<b>Conflictos:</b>	
<b>Materiales de soporte:</b>			
<b>Historial:</b>	07/04/2014 – Se define el requisito		

#### 9.2.6.5 Requisitos de inmunidad

<b>Requisito:</b>	<b>155</b>	<b>Tipo de requisito:</b>	<b>15e</b>	<b>Acontecimiento/BUC/PUC#</b>	<b>Todos</b>
<b>Descripción:</b>	El sistema evita las consultas o el código SQL de forma maligna por parte de algún usuario.				
<b>Justificación:</b>	El sistema tiene que protegerse de ataques, especialmente de los provocados por código SQL que permita leer datos a los cuales no se debería tener acceso.				
<b>Autor:</b>	Usuario				
<b>Criterio de satisfacción:</b>	El sistema no ejecuta ningún código SQL que no venga del propio sistema.				
<b>Satisfacción del cliente:</b>	2	<b>Insatisfacción del cliente:</b>	5		
<b>Prioridad:</b>	Alta	<b>Conflictos:</b>			
<b>Materiales de soporte:</b>					
<b>Historial:</b>	07/04/2014 – Se define el requisito				

## 9.2.7 Requisitos culturales y políticos

### 9.2.7.1 Requisitos culturales

<b>Requisito:</b>	<b>161</b>	<b>Tipo de requisito:</b>	<b>16a</b>	<b>Acontecimiento/BUC/PUC#</b>	<b>Todos</b>
<b>Descripción:</b>	El sistema será gratuito siempre para los usuarios que no pertenezcan a empresas.				
<b>Justificación:</b>	El sistema cumple con su compromiso social manteniéndose gratuito para aquellos usuarios cuyos proyectos son propios y no están expuestos a una empresa.				
<b>Autor:</b>	Usuario				
<b>Criterio de satisfacción:</b>	Auto explicativo				
<b>Satisfacción del cliente:</b>	5	<b>Insatisfacción del cliente:</b>	2		
<b>Prioridad:</b>	Alta	<b>Conflictos:</b>			
<b>Materiales de soporte:</b>					
<b>Historial:</b>	07/04/2014 – Se define el requisito				

### 9.2.7.2 Requisitos políticos

El sistema no tiene ningún requisito político.

## 9.2.8 Requisitos legales

### 9.2.8.1 Requisitos de cumplimiento

<b>Requisito:</b>	<b>171</b>	<b>Tipo de requisito:</b>	<b>17a</b>	<b>Acontecimiento/BUC/PUC#</b>	<b>Todos</b>
<b>Descripción:</b>	El sistema cumple con la Ley Orgánica de Protección de Datos (LOPD)				
<b>Justificación:</b>	El sistema ha de cumplir las leyes españolas.				

<b>Autor:</b>	Usuario		
<b>Criterio de satisfacción:</b>	Se consulta con un experto si el sistema cumple con la LOPD		
<b>Satisfacción del cliente:</b>	2	<b>Insatisfacción del cliente:</b>	2
<b>Prioridad:</b>	Alta	<b>Conflictos:</b>	
<b>Materiales de soporte:</b>			
<b>Historial:</b>	07/04/2014 – Se define el requisito		

#### 9.2.8.2 Requisitos de estándares

<b>Requisito:</b>	<b>172</b>	<b>Tipo de requisito:</b>	<b>17b</b>	<b>Acontecimiento/BUC/PUC#</b>	<b>Todos</b>
<b>Descripción:</b>	El sistema cumple los estándares W3C que tienen relación con las páginas web.				
<b>Justificación:</b>	El sistema se ha de poder ejecutar en cualquier otro sistema que cumpla los estándares.				
<b>Autor:</b>	Todos				
<b>Criterio de satisfacción:</b>	Se implementan las “W3C Recommendations”.				
<b>Satisfacción del cliente:</b>	1	<b>Insatisfacción del cliente:</b>	5		
<b>Prioridad:</b>	Alta	<b>Conflictos:</b>			
<b>Materiales de soporte:</b>					
<b>Historial:</b>	07/04/2014 – Se define el requisito				

## 10 Justificación de las tecnologías escogidas

Para la parte web he decidido utilizar principalmente Ruby on Rails<sup>21</sup>, los motivos que han llevado a esta decisión son los siguientes:

- ✓ Ruby on Rails tiene una integración magnífica del diseño Modelo Vista Controlador (MVC) que es el diseño que se quería utilizar desde el principio.
- ✓ Ruby on Rails tiene un gran sistema de testing basado en pruebas unitarias que facilita mucho el testing y por ello es útil en metodologías ágiles como la utilizada en este proyecto.
- ✓ La programación en Ruby on Rails es muy sencilla y rápida con poco código se pueden conseguir grandes cosas.

La base de datos escogida es PostgreSQL, esta decisión ha sido forzada ya que el servidor utilizado (Heroku) solo dispone de bases de datos de este tipo. [13]

En cuanto a la aplicación Android se valoró si hacerla nativa, híbrida o bien web, la decisión ha sido basada en mis conocimientos, me manejo con mayor destreza con las aplicaciones nativas y la implementación de notificaciones requiere que por lo menos esa parte sea nativa, por tanto, la parte Android será creada de forma nativa.

---

<sup>21</sup> Lenguaje de programación orientado al diseño web, para más información <http://rubyonrails.org/>

## 11 Diseño arquitectónico aplicación web

La aplicación web está hecha con Ruby on Rails, html5<sup>22</sup>, css3<sup>23</sup> y algún pequeño código de JQuery<sup>24</sup>, en esta sección están explicados los patrones de diseño que se usarán para la parte web, los protocolos, las capas de presentación, dominio y datos, el modelo de despliegue.

### 11.1 Patrones de diseño

#### 11.1.1 Modelo Vista Controlador

La aplicación web utiliza principalmente el patrón modelo vista controlador (MVC), este patrón está basado en separar la interface del usuario, de la lógica de la aplicación y de los datos, de esta manera el usuario pide información al controlador, el controlador pide los datos al modelo y luego la vista a las vistas y le devuelve todo al usuario. Explicado con más claridad en la figura siguiente.

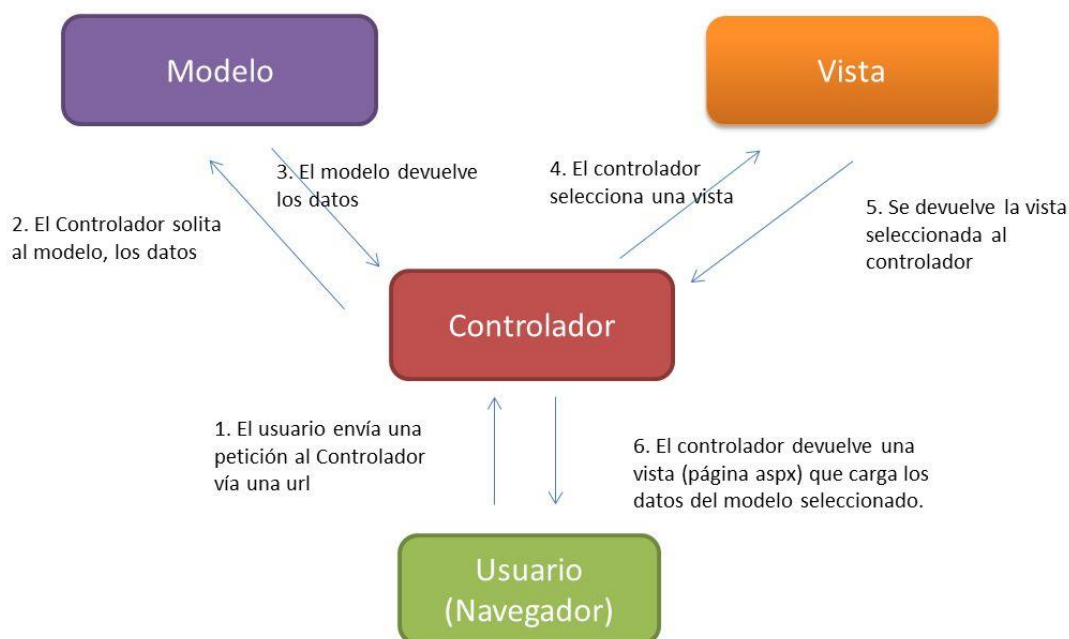


Figura 6 Diagrama funcionamiento patrón MVC [13]

<sup>22</sup> Lenguaje web por excelencia que interpretan todos los navegadores web más conocidos, para más información [http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)

<sup>23</sup> Hojas de estilo cuyo uso como el nombre indica recae en dar estilo a una página web, para más información <http://www.w3schools.com/css/>

<sup>24</sup> Biblioteca de componentes para jQuery que añade aspectos visuales, para más información <http://jqueryui.com/>

## [MVC en Ruby on Rails](#)

Ruby on Rails sigue la filosofía MVC, por lo tanto, para Rails, cada modelo, controlador y la carpeta que contiene las vistas tienen una nomenclatura predeterminada y se refieren a una clase u objeto, es decir, en nuestro caso tenemos que para nuestra clase Projects se crearan: El modelo Project.rb, el controlador projects\_controller.rb, y la carpeta de vistas projects. De esta manera Rails entiende que todas estas clases están relacionadas entre sí y con el objeto o tabla Projects, en el modelo Project.rb irán las restricciones del modelo y sus relaciones, los campos no serán necesarios porque Rails buscará la tabla Projects en la base de datos y sabrá que campos tiene, en el controlador irán las funciones típicas de index, new, show, create, edit, update, que al finalizar su código buscaran las vista con el mismo nombre que estén contenidas en la carpeta de vistas projects, en las vistas encontraremos ficheros con el formato erb.html, el formato erb es para que la vista pueda contener código ruby y de esta forma poder añadir lógica a las vistas, actúa como un server page que construye la vista. Puede entenderse mejor con el diagrama del apartado 11.3 [Diseño interno](#)

### 11.1.2 Active Record

El patrón active record, es un patrón diseñado para almacenar los datos en una base de datos relacional <sup>25</sup>como MySQL en este caso. El patrón se basa en crear una interface justo encima de la base de datos que incluya las operaciones esenciales: Insert, Update, Delete, de esta manera la aplicación hace las consultas a esta interface de manera que la interface es la encargada de recoger los datos de la base de datos y devolverlos a la aplicación de manera que esta ultima los pueda entender.

Ruby on Rails usa este patrón para comunicarse con la base de datos, pese que se puede evitar es muy útil hacer uso del porque facilita la comunicación entre la base de datos y la aplicación y elimina la dependencia de la aplicación a un solo tipo de base de datos, es decir, si más adelante se decidiera usar otra base de datos no tendría problemas con las operaciones a la base de datos ya implementadas.

---

<sup>25</sup> Base de datos que guarda relación entre sus objetos.

## 11.2 Capa de presentación

A continuación se muestra el mapa de navegación de la aplicación web.

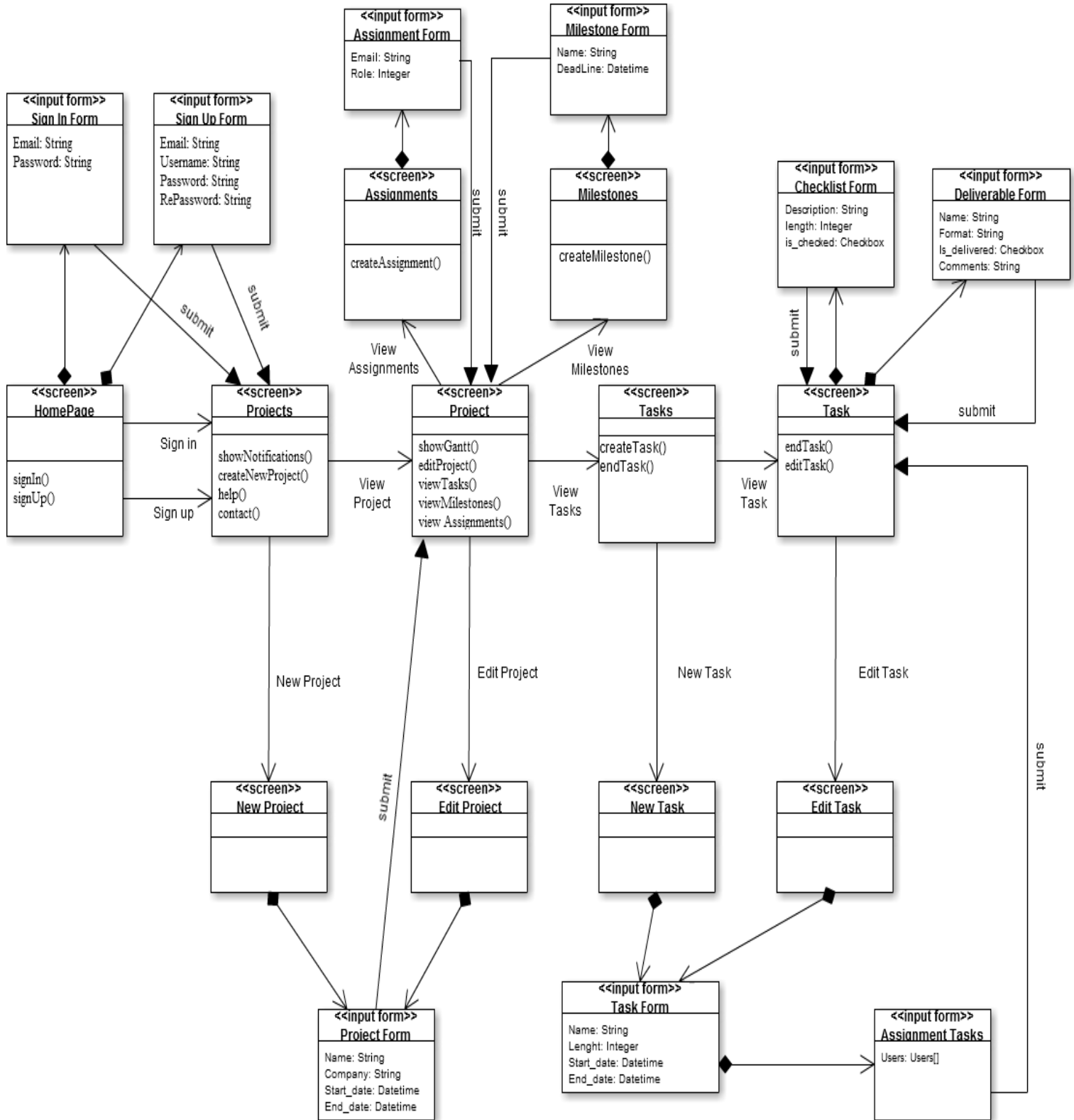


Figura 7 Mapa navegación aplicación web

A continuación se muestran capturas de pantalla de la aplicación para poder ver mejor su funcionamiento y visualización, se han añadido solo algunas de las capturas para entender el mapa navegacional.

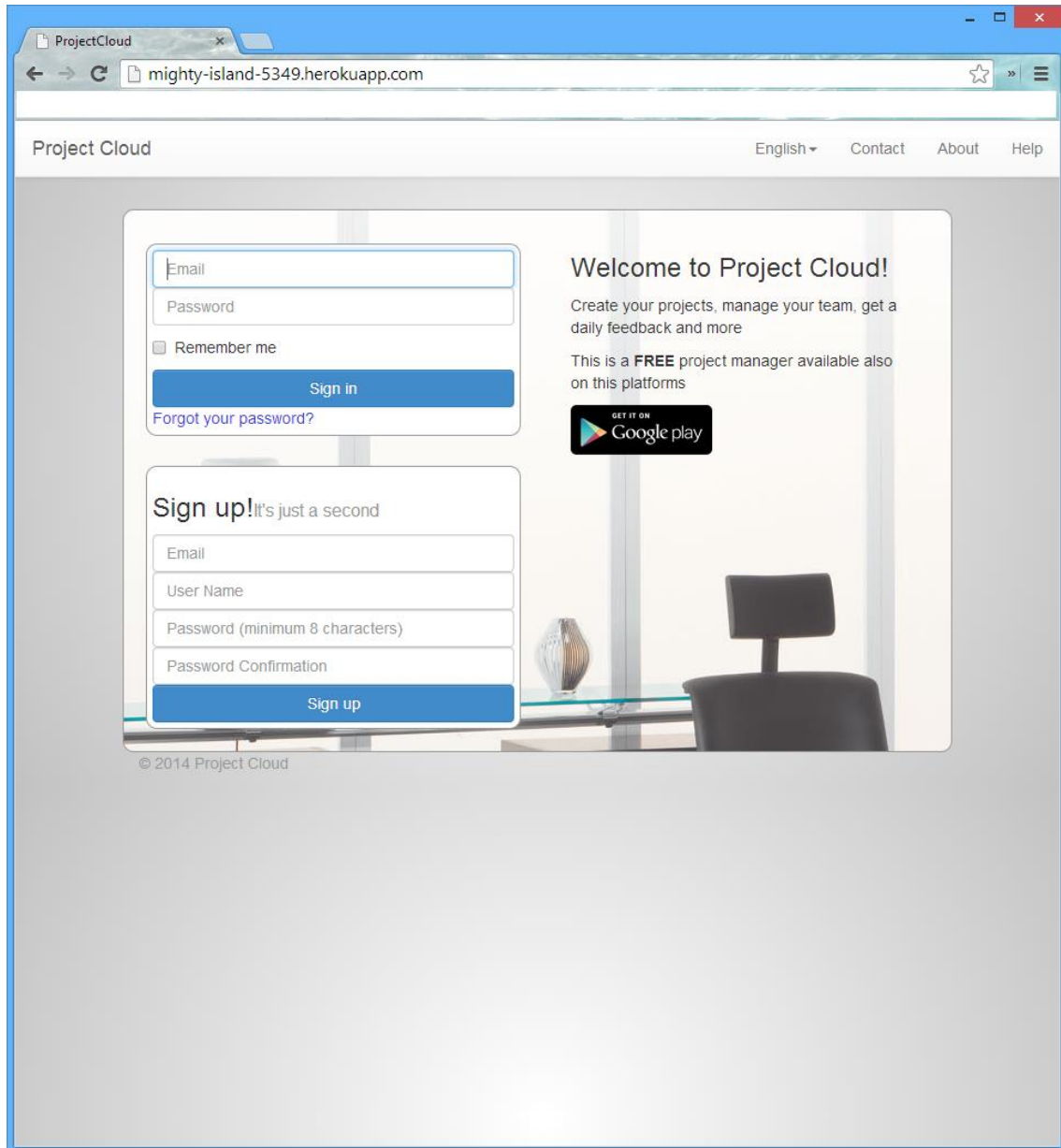


Figura 8 Pantalla principal (screenshot)

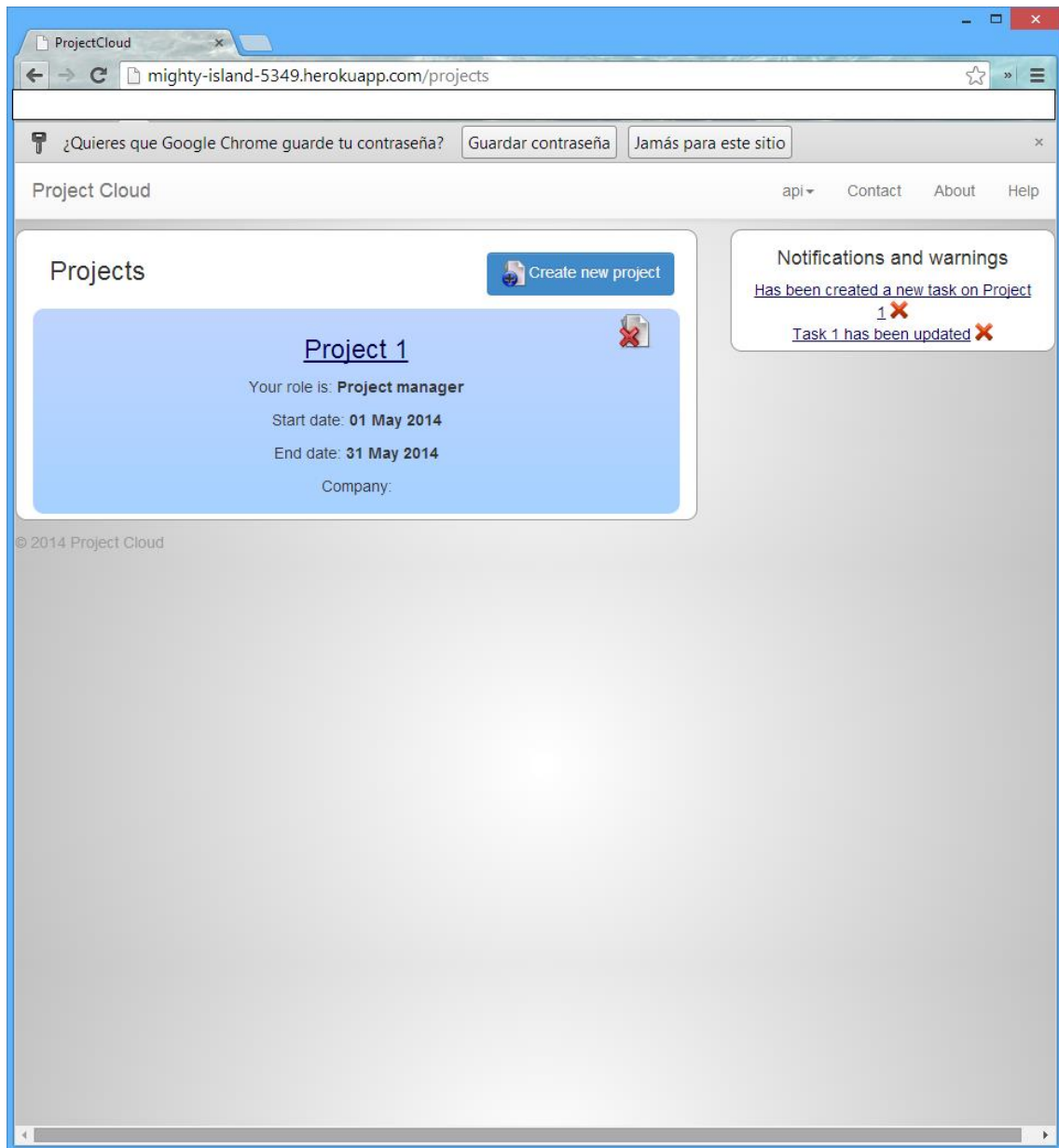


Figura 9 Pantalla indice de proyectos (screenshot)

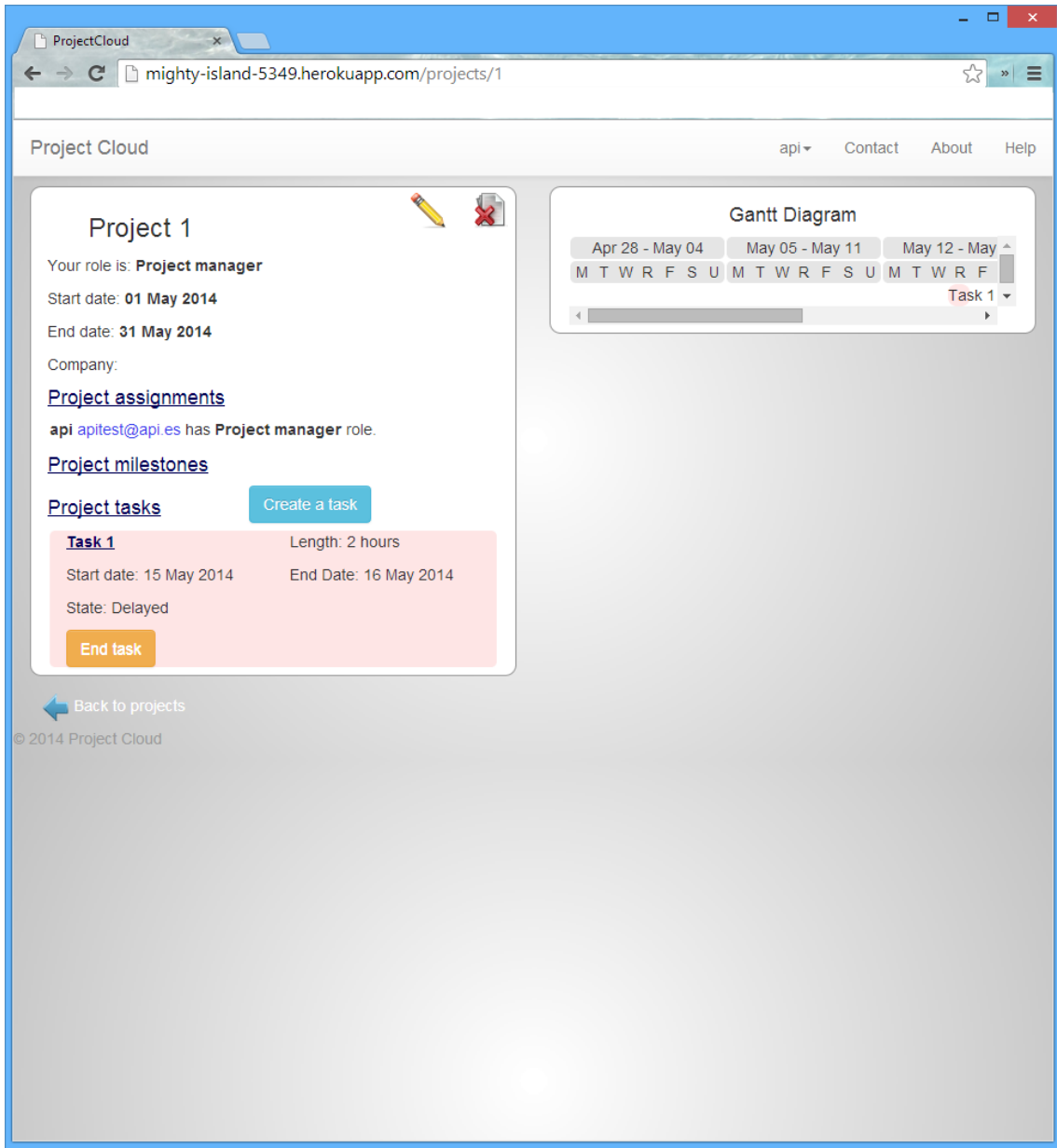


Figura 10 Pantalla proyecto (screenshot)

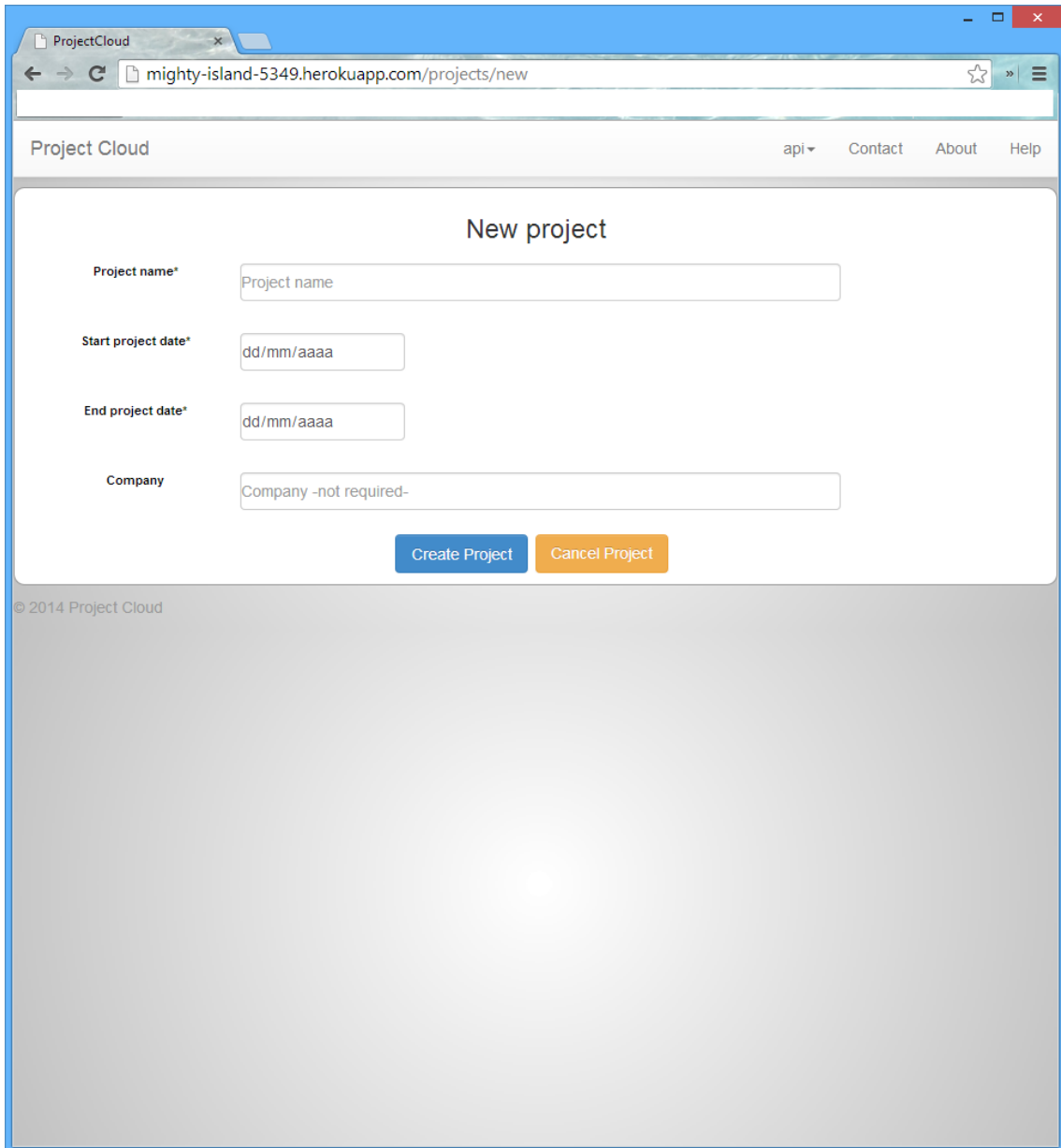


Figura 11 Pantalla formulario proyecto (screenshot)

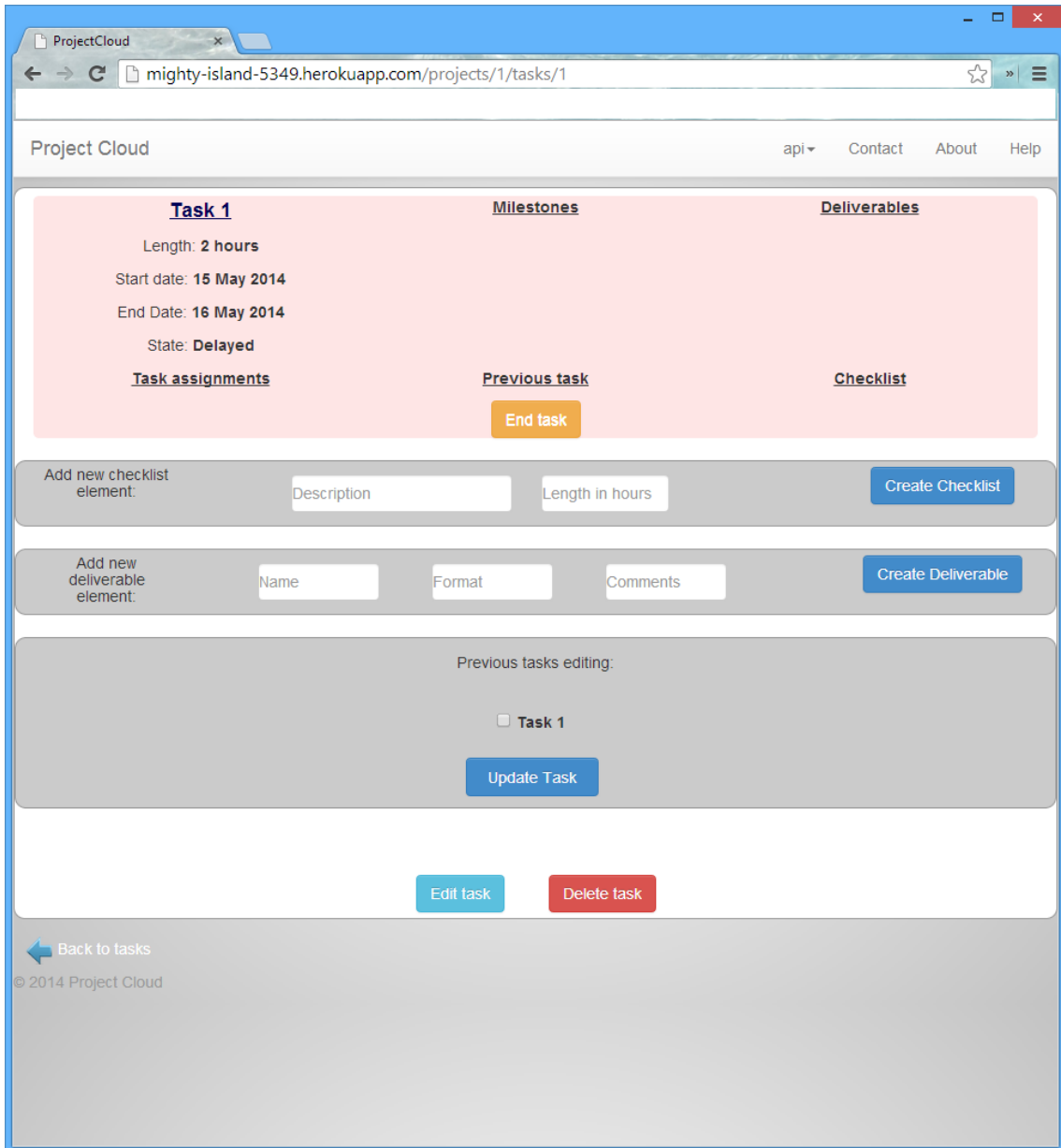


Figura 12 Pantalla tarea (screenshot)

### 11.3 Diseño interno

Este primer esquema muestra el diseño interno estático de la aplicación, como se puede ver solo lo he realizado para la parte de Project, sin embargo, el diseño es idéntico para los demás modelos controladores y vistas.

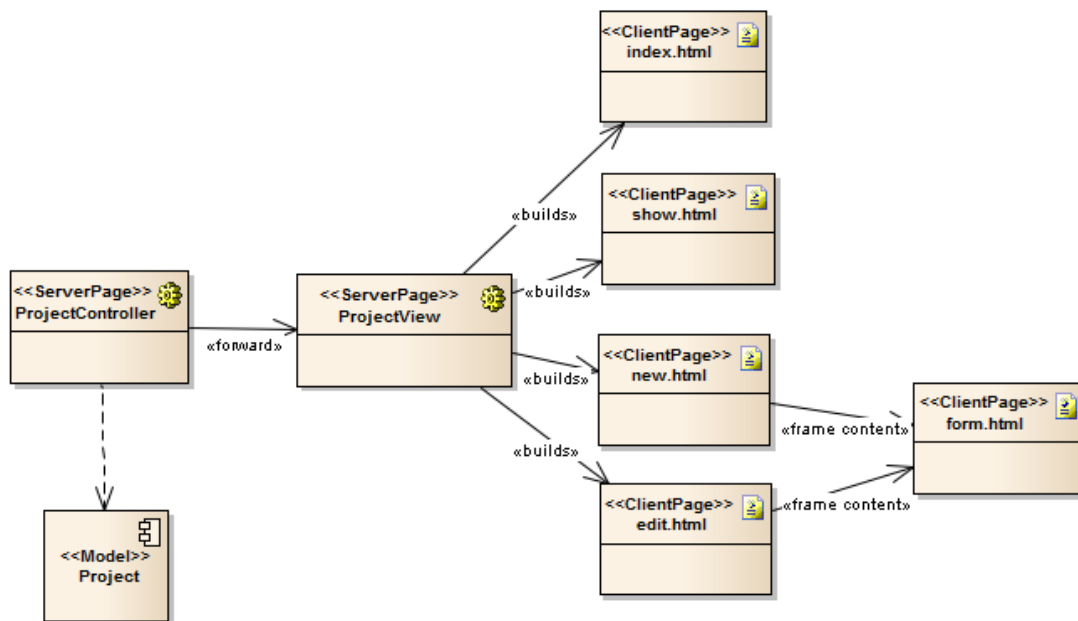
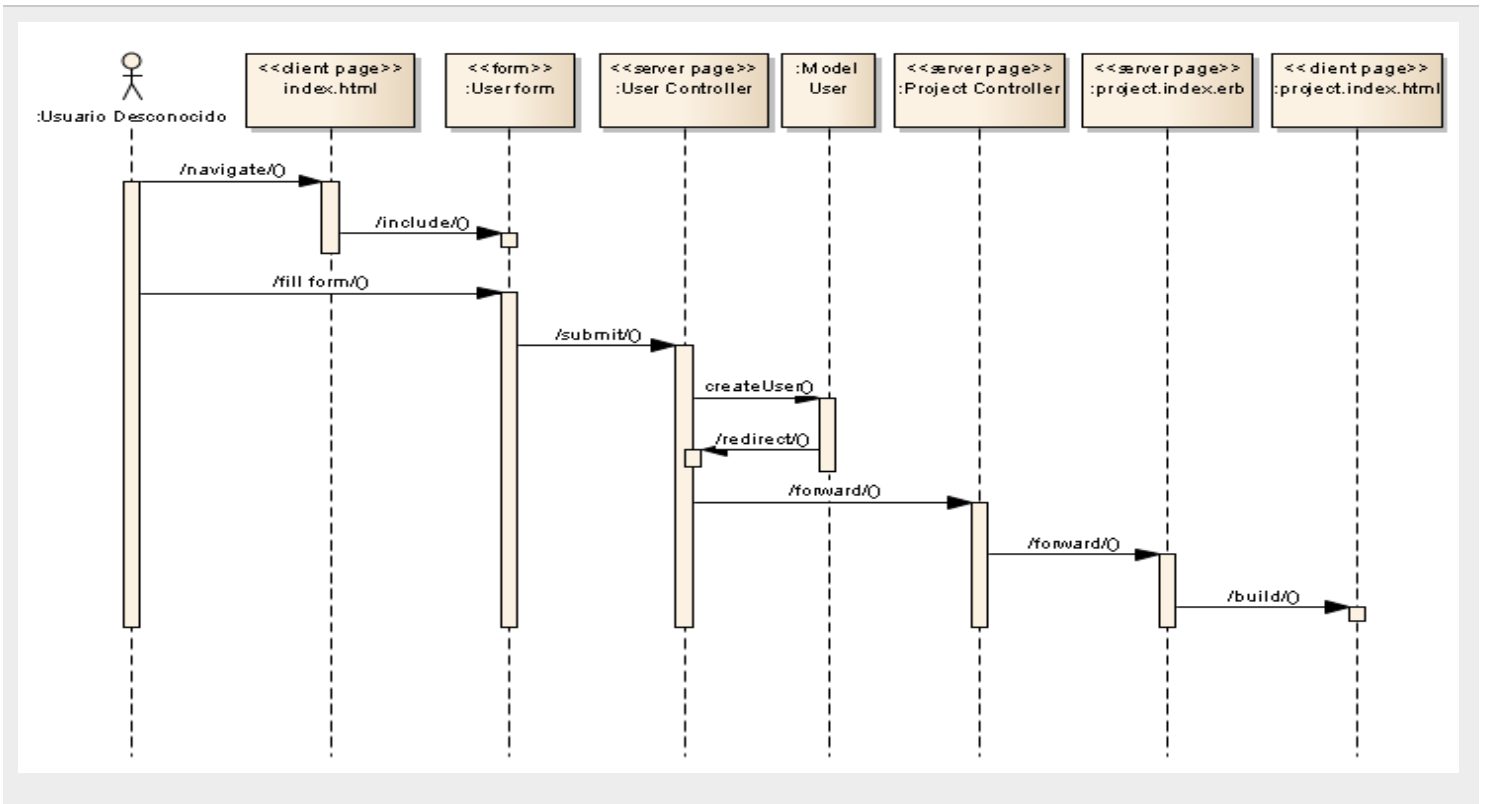


Figura 13 Diseño interno aplicación web

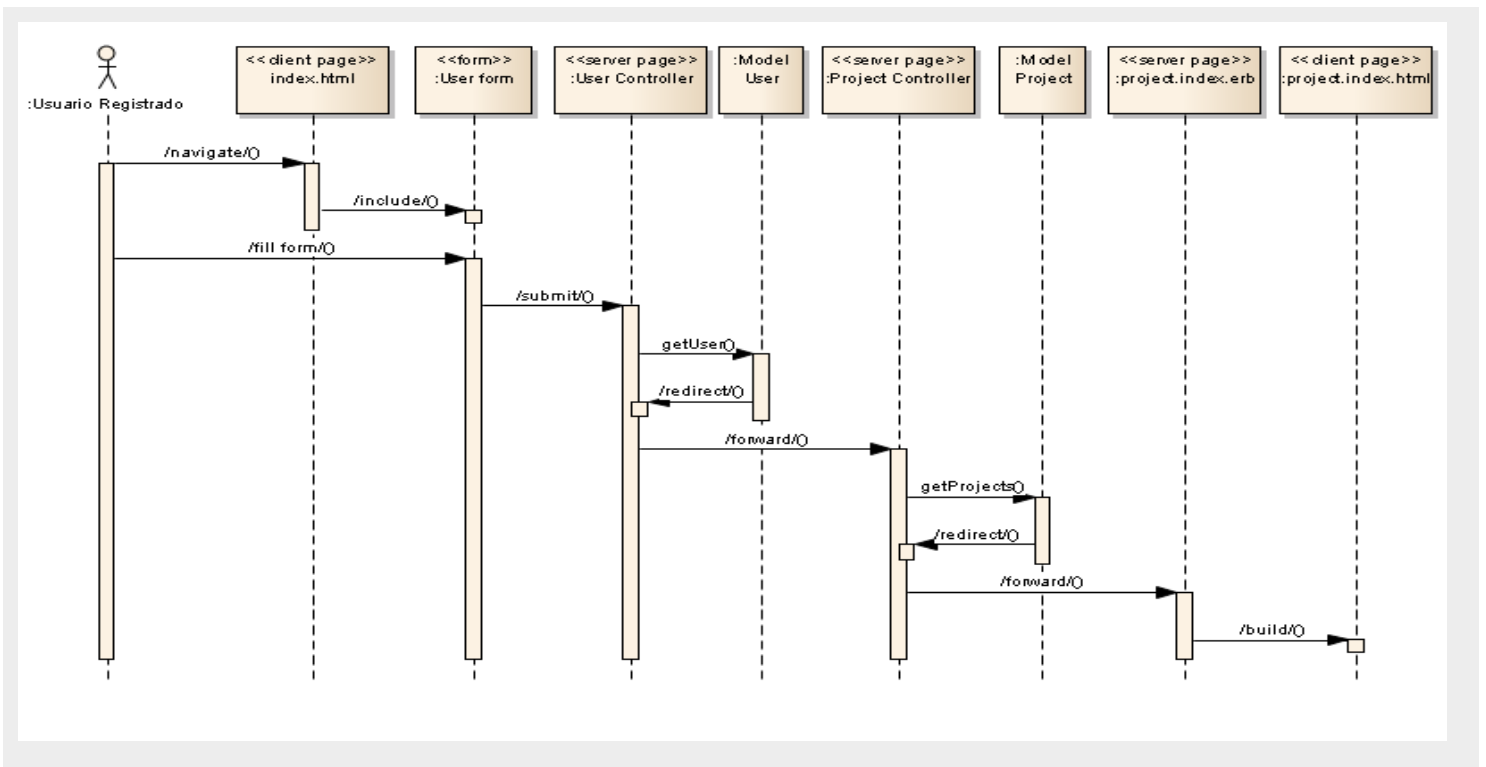
#### 11.3.1 Modelo de comportamiento

Para representar los modelos de comportamiento los he agrupado por casos de usos que tienen un diseño muy similar, de esta manera, por cada grupo mostraré el diseño interno de el primer caso de uso y se enumeraran los parecidos.

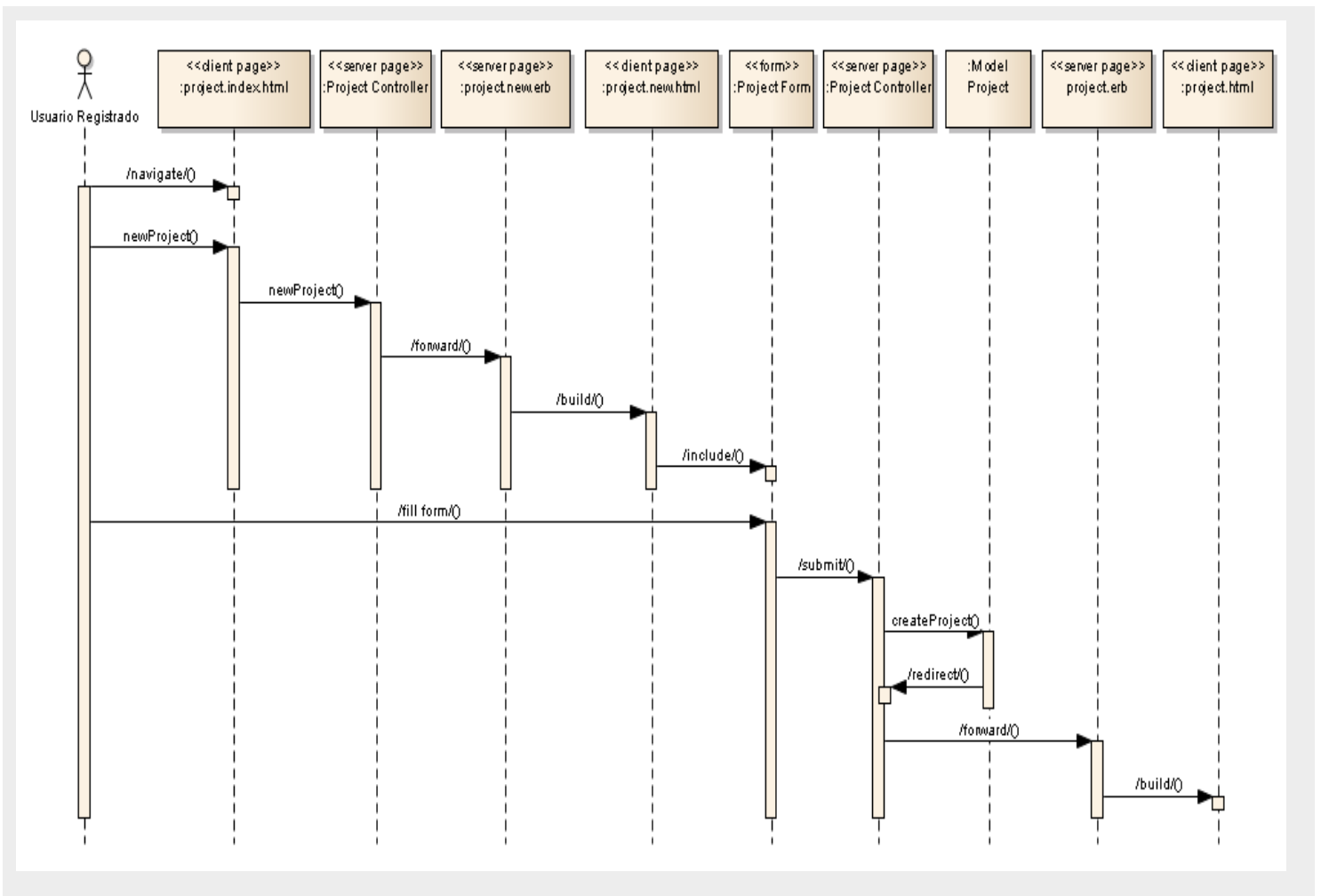
## Registrar Usuario(1)



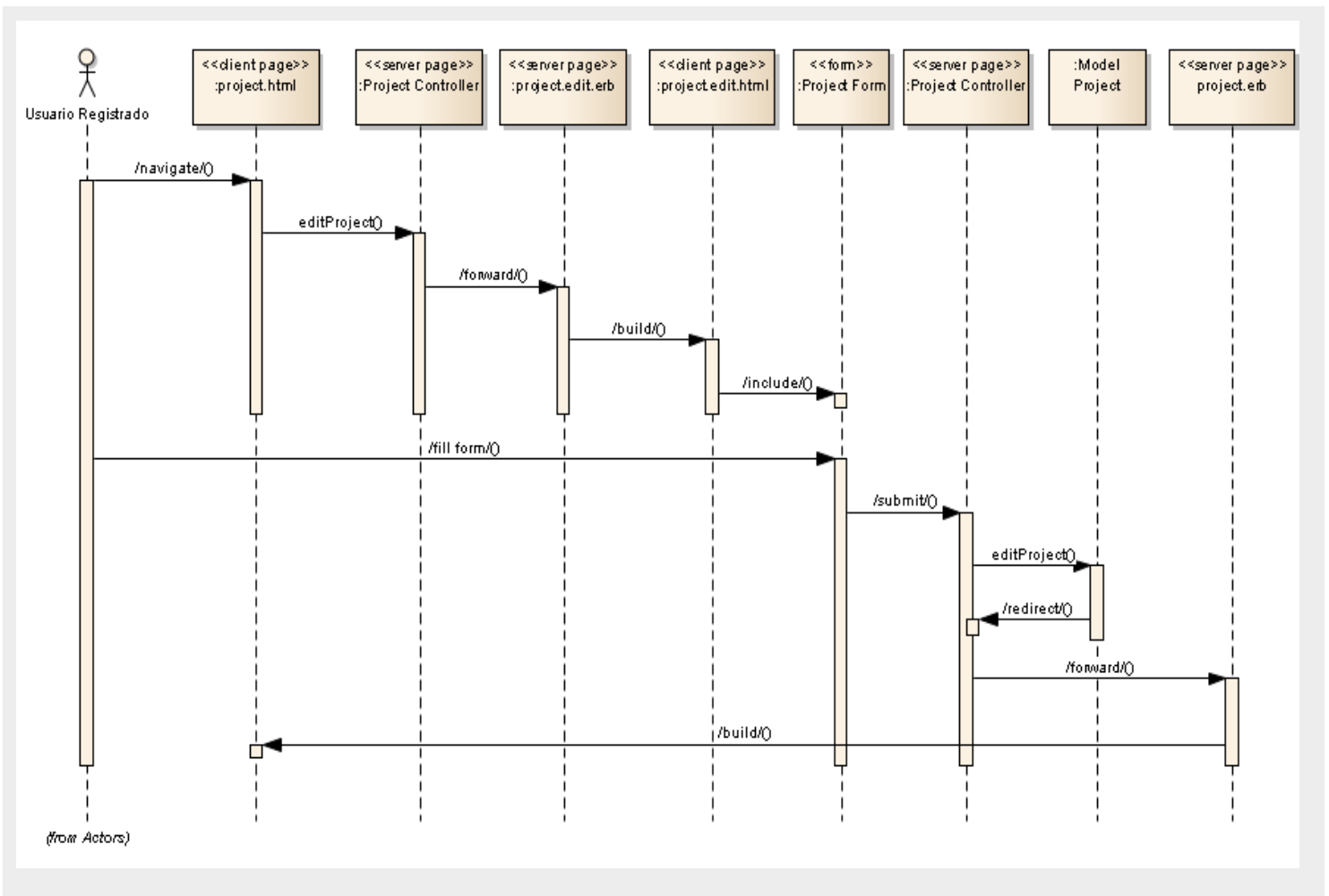
## Iniciar Sesión(2), Cerrar Sesión(3)



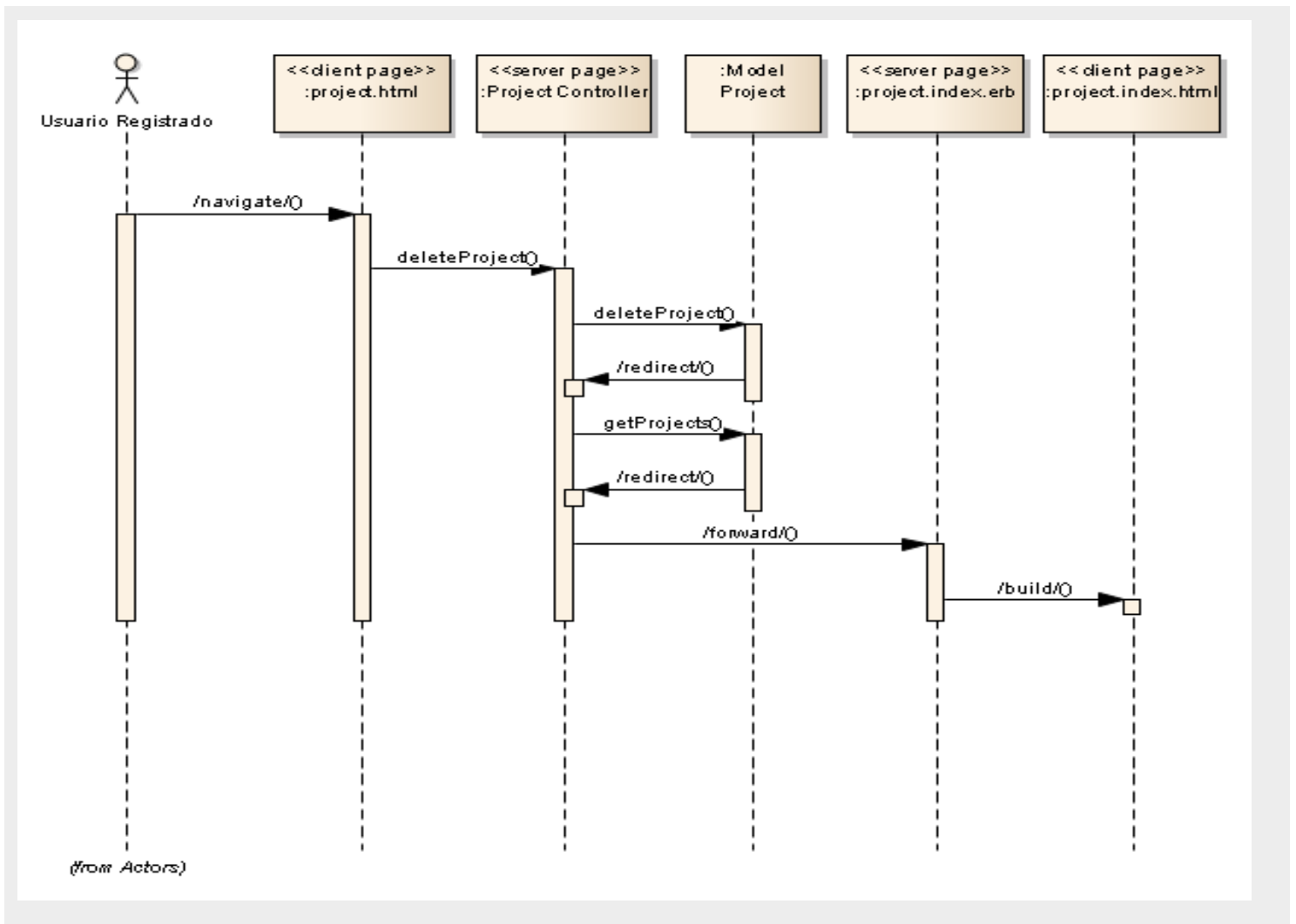
**Crear nuevo proyecto (4), Asignar miembro a un proyecto (7), Crear nueva tarea (10), Asignar miembro a una tarea (13), Crear nuevo hito (15), Crear nuevo entregable (18), Crear nuevo checklist (21)**



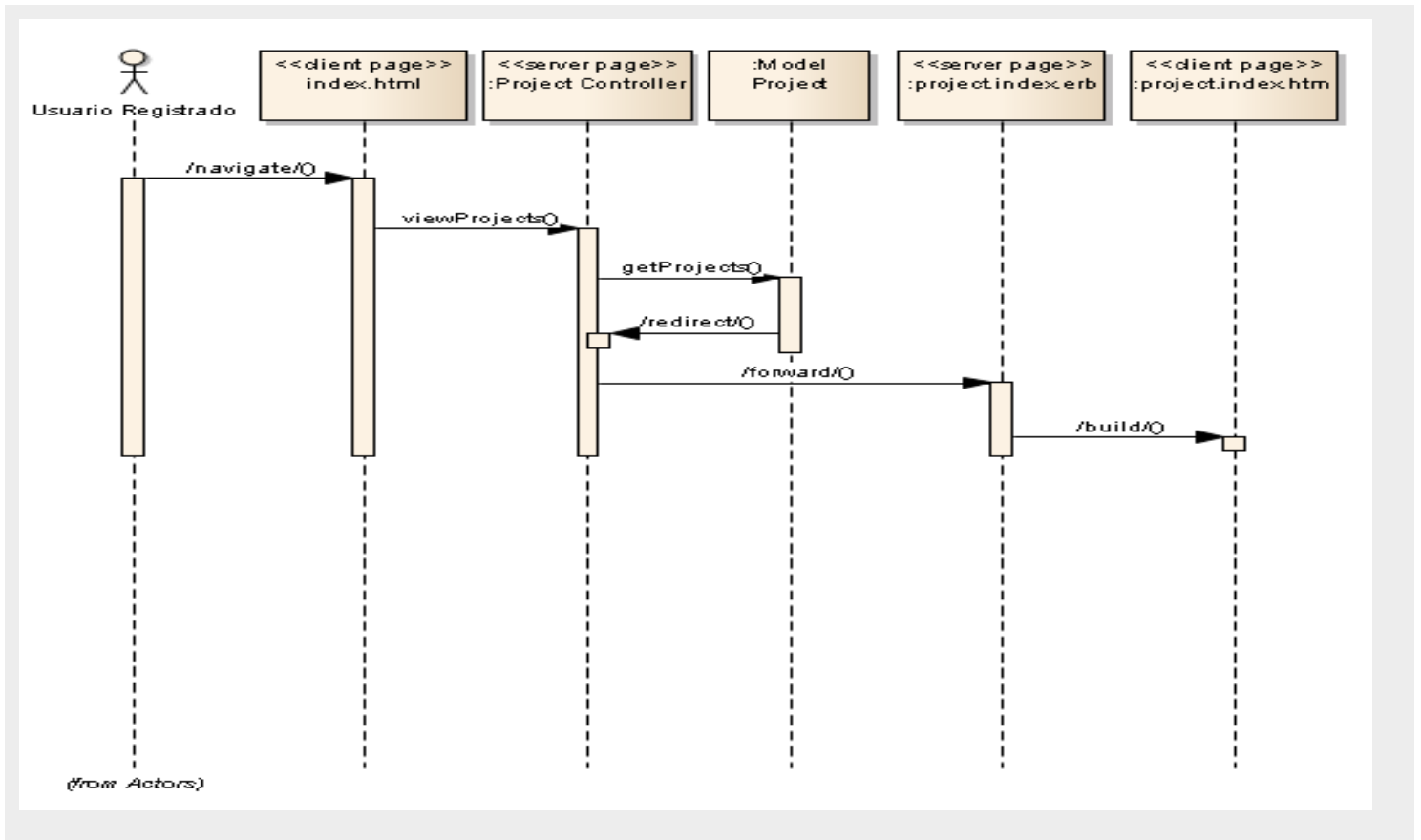
**Editar proyecto (5), Editar miembro de un proyecto (8), Editar Tarea (11), Editar hito (16), Editar estado entregable (19), Editar estado checklist (22)**



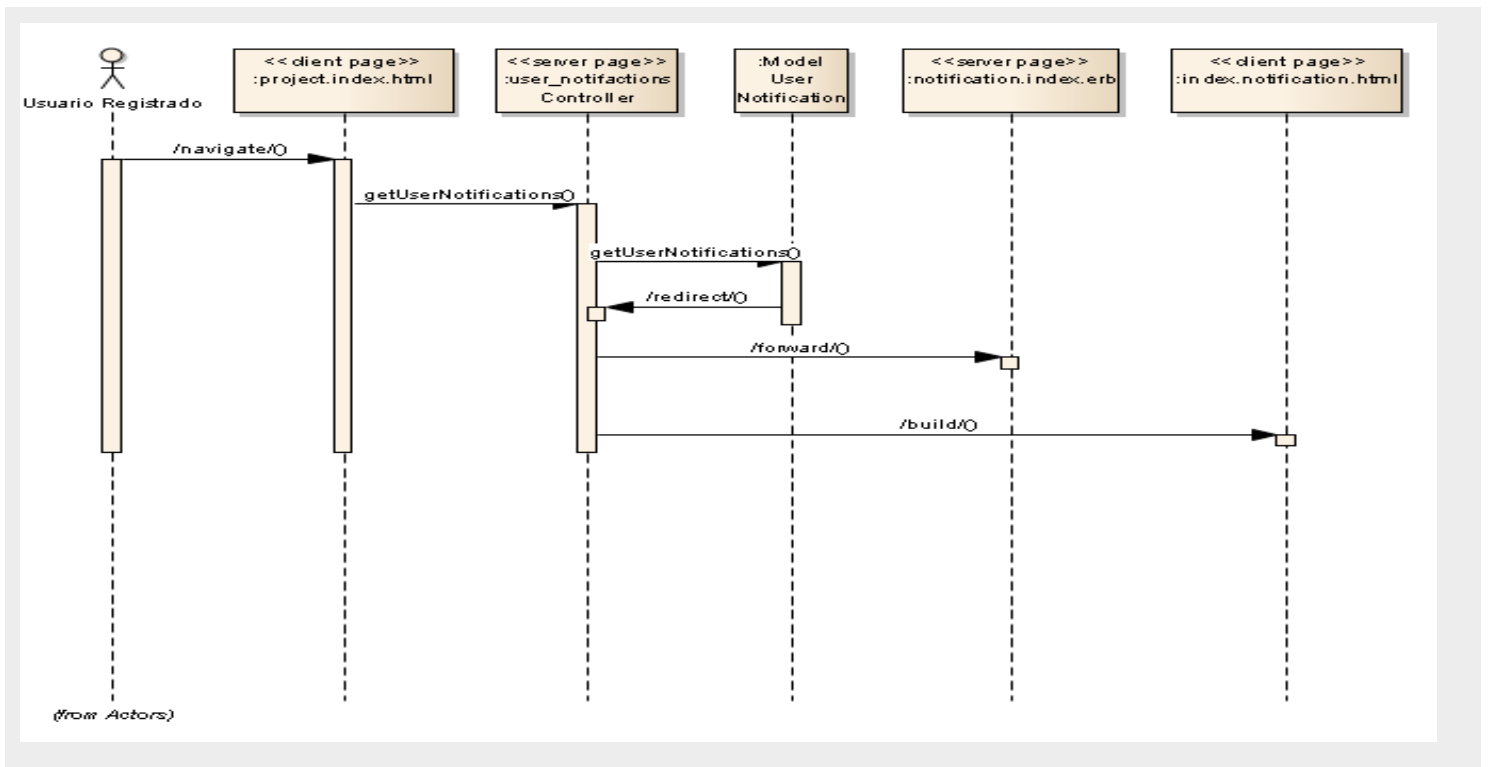
**Eliminar proyecto (6), Eliminar miembro de un proyecto(9), Eliminar Tarea (12), Eliminar miembro de una tarea (14), Eliminar hito (17), Eliminar entregable (20), Eliminar checklist (23)**



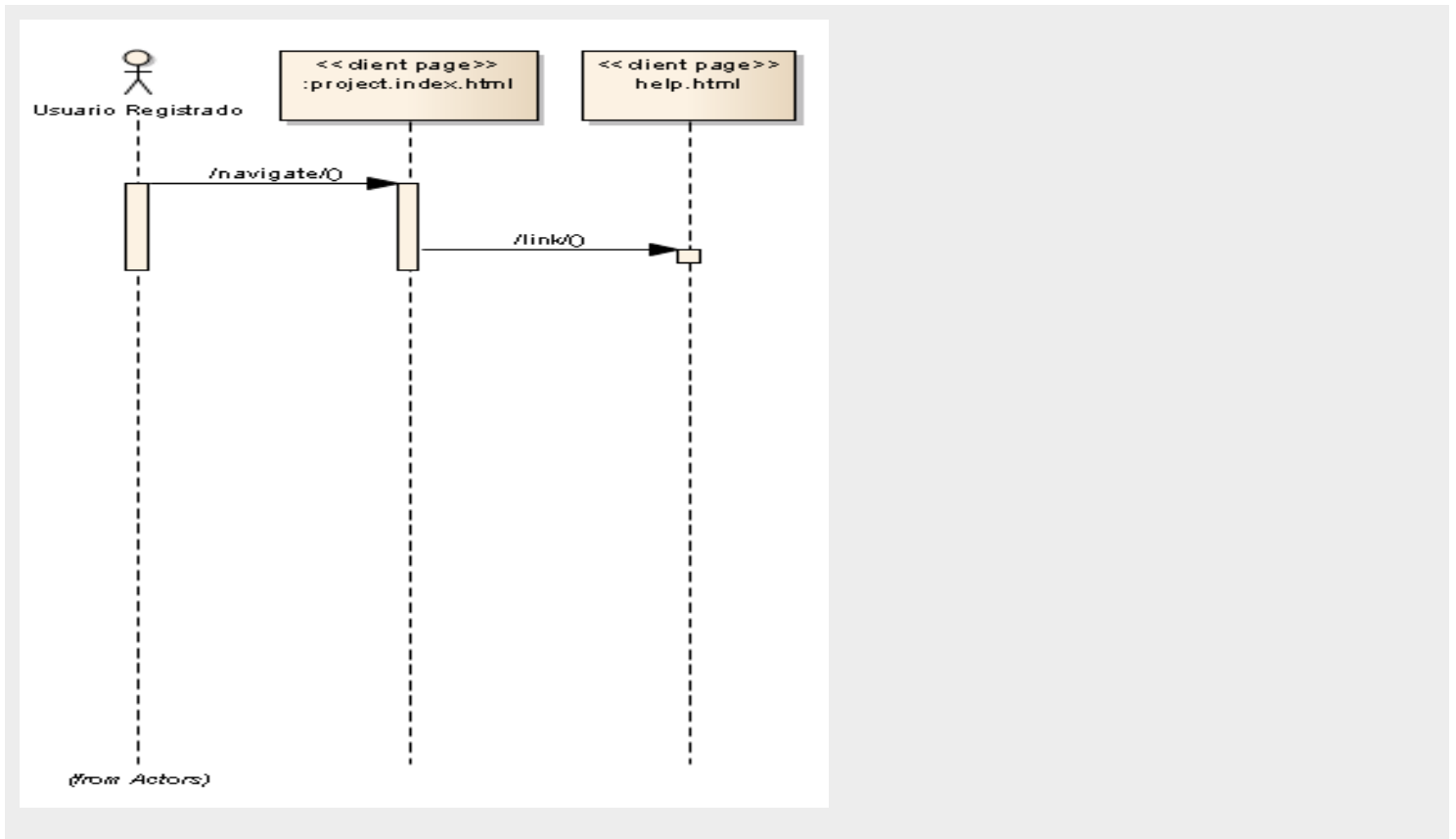
## Ver proyectos (24), Ver tareas (25), Ver diagrama de Gantt (26)



## Recibir notificaciones (27)



Ver ayuda (28)



## 11.4 Capa de datos

Teniendo en cuenta el modelo conceptual anterior la base de datos queda de la siguiente manera, esta base de datos es utilizada tanto por la aplicación web como por la aplicación Android, para hacer las pruebas ha sido construida en MySQL, sin embargo, posteriormente será construida en PostgreSQL <sup>26</sup>debido a que Heroku utiliza una base de datos PostgreSQL, los cambios necesarios son hechos automáticamente por Ruby on Rails, la principal diferencia reside en los tipos boolean que en MySQL son tratados como Tinyint.

---

<sup>26</sup> Servidor de bases de datos relacional, para más información: <http://www.postgresql.org.es/>



Figura 14 Capa de datos aplicación web

## 12 Diseño arquitectónico Android y API

En esta parte se puede ver el diseño de la aplicación Android y de la API para conectarse con la aplicación web, hay que diferenciar por tanto de un lado el cliente Android que está implementado de forma nativa y por el otro la API que forma parte de la aplicación web y por tanto está hecho con Ruby on Rails.

La parte cliente Android solo se encarga de mostrar una interface al usuario y de enviar notificaciones al usuario, sin embargo, esta adquiere los datos de la API. En el diagrama siguiente se muestra el funcionamiento de forma visual para un mejor entendimiento.

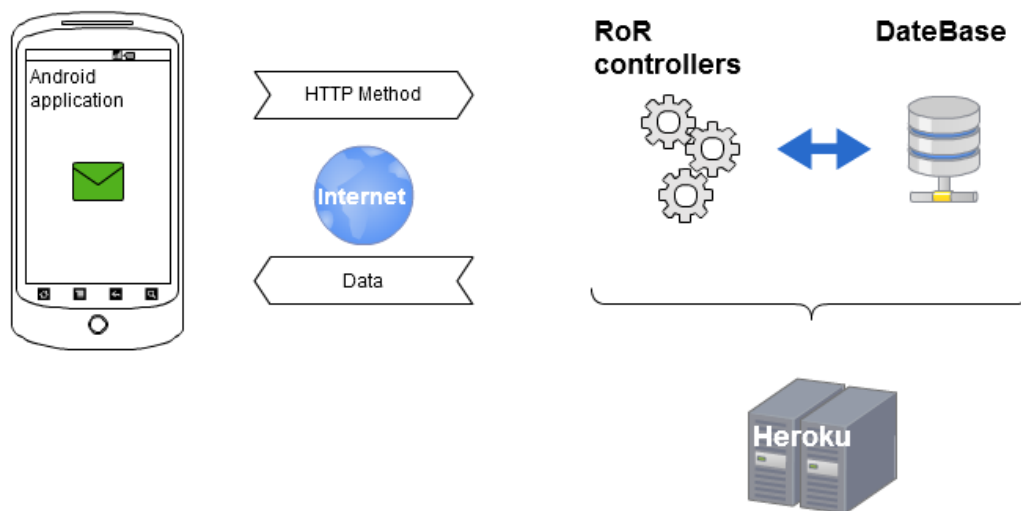


Figura 15 Diagrama AndroidClient + API

## 12.1 Patrones de diseño

### 12.1.1 REST

REST (Representational State Transfer) es un principio arquitectónico dirigido a los servicios web, la idea se basa en enviar la información necesaria de usuario a servidor siguiendo los métodos de HTTP, los principios básicos de REST son:

- Utiliza los métodos HTTP de manera explícita (POST, GET, PUT, DELETE).
- No mantiene estado<sup>27</sup>.
- Expone URIs con forma de directorios, que contienen toda la información necesaria para resolver una petición.
- Transfiere XML, JavaScript Object Notation (JSON), o ambos. [14]

Esta arquitectura es muy útil para aplicaciones web simples que no requieren de gran lógica del lado del servidor y cuando el estado no es importante. A parte se puede construir fácilmente un API (Application Programming Interface) REST para dar datos a otras aplicaciones. Ruby on Rails está muy sujeto a esta arquitectura y por eso principalmente se utilizará.

---

<sup>27</sup> El estado de una aplicación web se define como aquel sitio donde el servidor o el usuario se encuentran en un momento dado, puede ser una URL o información más compleja.

### API RESTful resultante

Una API (Application Programming Interface), es una interface que permite a los programadores ejecutar ciertas funcionalidades de un software desde un tercero.

Method	URI	Parámetros(Json)	Respuesta(Json)
<b>Post</b>	/api/users/sign_in	{ "user": { "email": "<email>", "password": "<password>" } }	{ "success": "<true or false>", "auth_token": "<auth_token>", "email": "<email>" }
<b>Get</b>	/api/user_notifications		User_Notification.json
<b>Get</b>	/api/notifications		Notification.json

Tabla 8 Métodos API

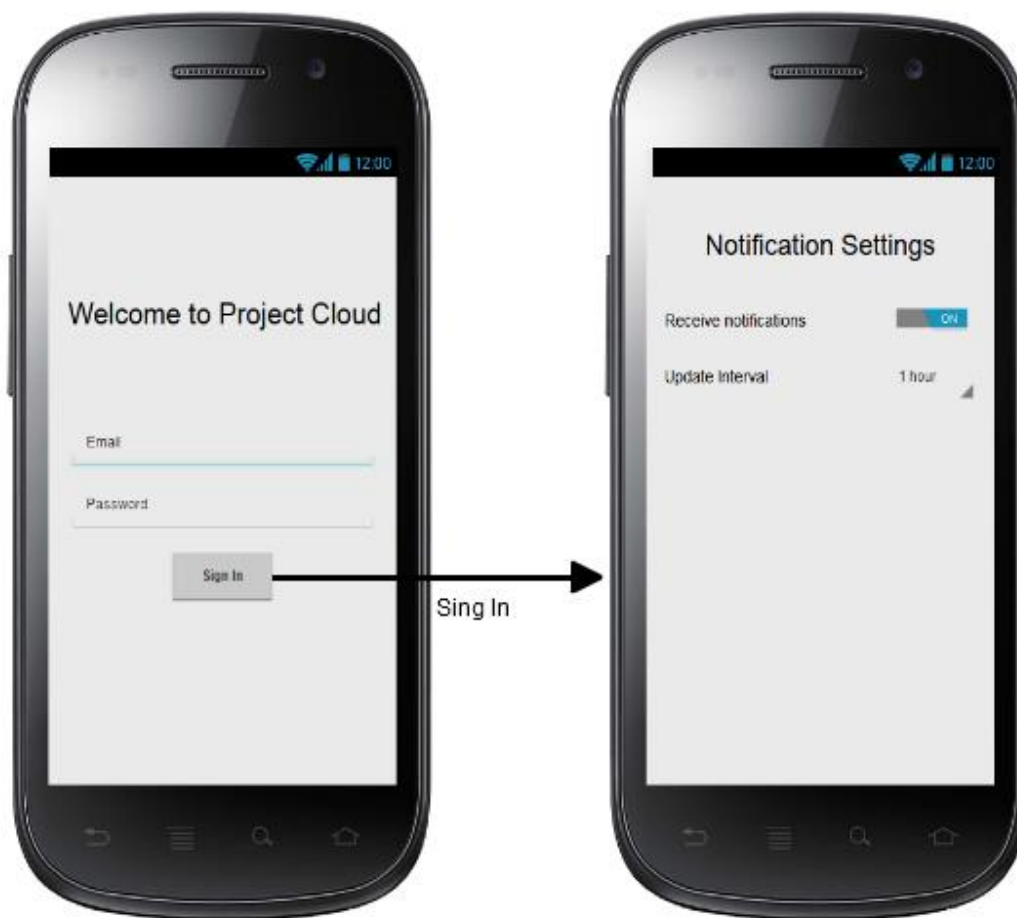
La API creada, mantendrá un cookie que servirá para mantener el estado durante toda la sesión, por tanto, el token <sup>28</sup> solo será utilizado si más adelante se generan peticiones POST, para mejorar la seguridad.

## 12.2 Capa de presentación

A continuación se muestran los StoryBoards de la aplicación Android, es un modelo bastante sencillo ya que la finalidad de la aplicación es solo recibir notificaciones, la aplicación web está construida para poderse ver sin problemas desde dispositivos móvil y por tanto la reconstrucción de la aplicación para añadir todas las funcionalidades sería reiterativa e innecesaria.

---

<sup>28</sup> Caracteres pseudoaleatorios y únicos para un usuario que lo identifican en la base de datos.



*Figura 16 Storyboards aplicación Android*

## 12.3 Diseño interno

A continuación se muestra el diseño interno de la aplicación donde se pueden apreciar dos partes, el cliente Android y la parte de API.

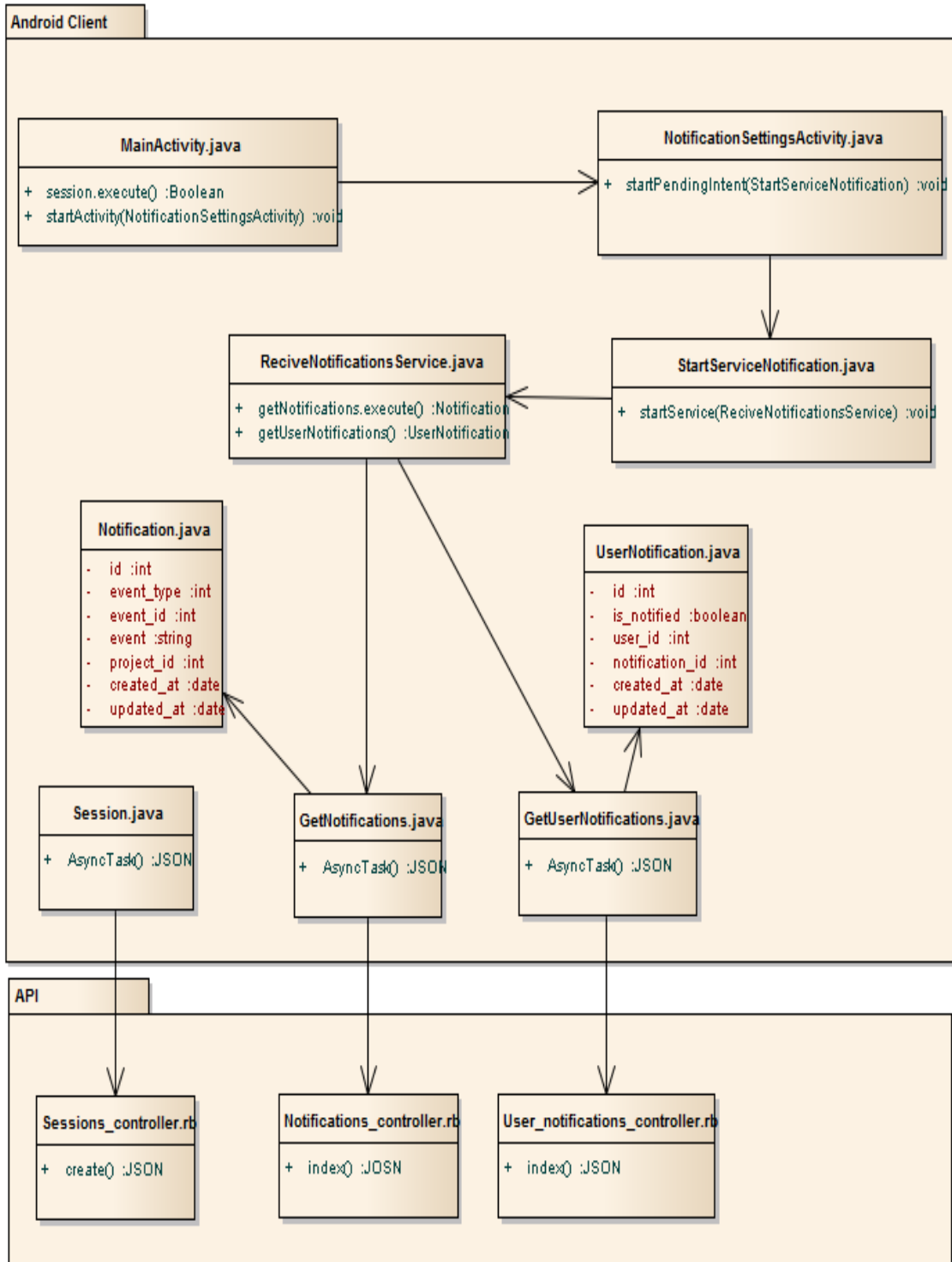
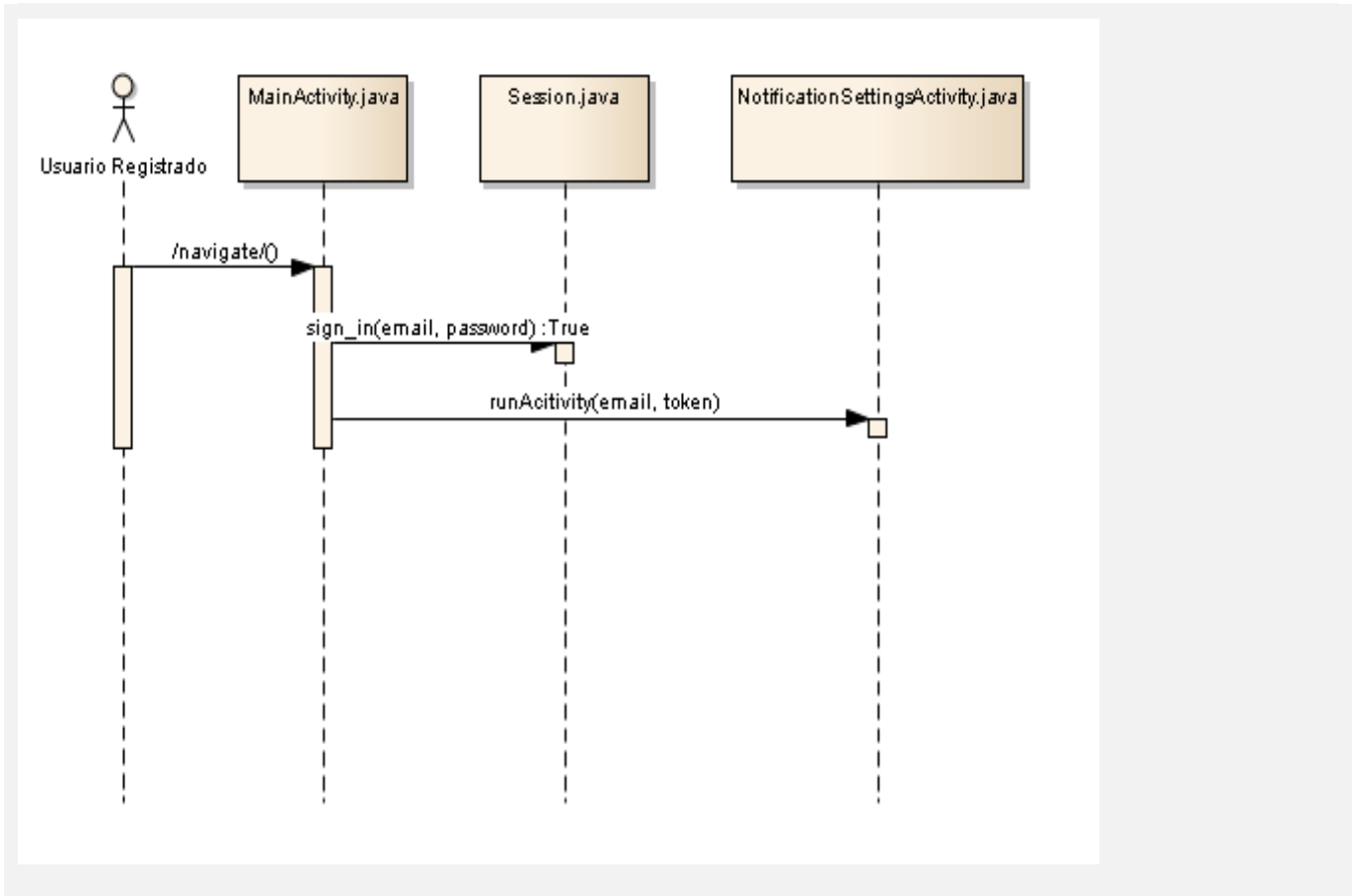


Figura 17 Diseño interno cliente android y API

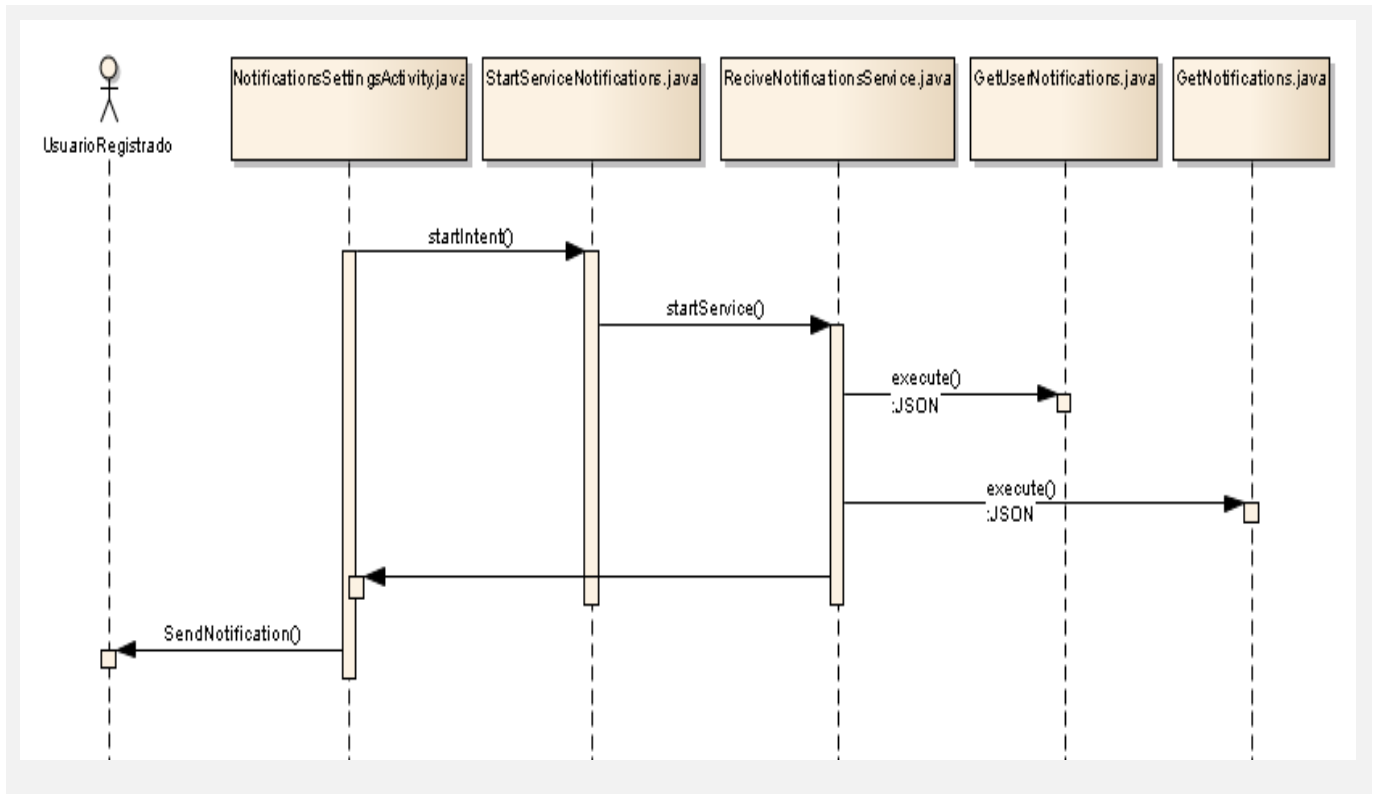
### 12.3.1 Modelo de comportamiento

A continuación se muestran todos los modelos de comportamiento para los casos de uso que implementa la aplicación Android.

#### Iniciar Sesión(2)



## Recibir Notificaciones(27)



## 12.4 Capa de datos

No existe una capa de datos, los datos son dados por la aplicación web y por tanto para la aplicación la capa de datos es transparente.

## 13 Modelo de despliegue Aplicación web y Android

El modelo de despliegue se puede ver a continuación, en él podemos ver que el código de la aplicación web está en los servidores de GitHub, el de Android también aunque el motivo es la facilidad, ese código es el utilizado para ejecutarlo en los servidores de Heroku que también contienen la base de datos implementada en Postgre SQL, como la aplicación posee una API RESTUful, la aplicación Android Nativa se conecta a la aplicación web a través de la API como se puede apreciar.

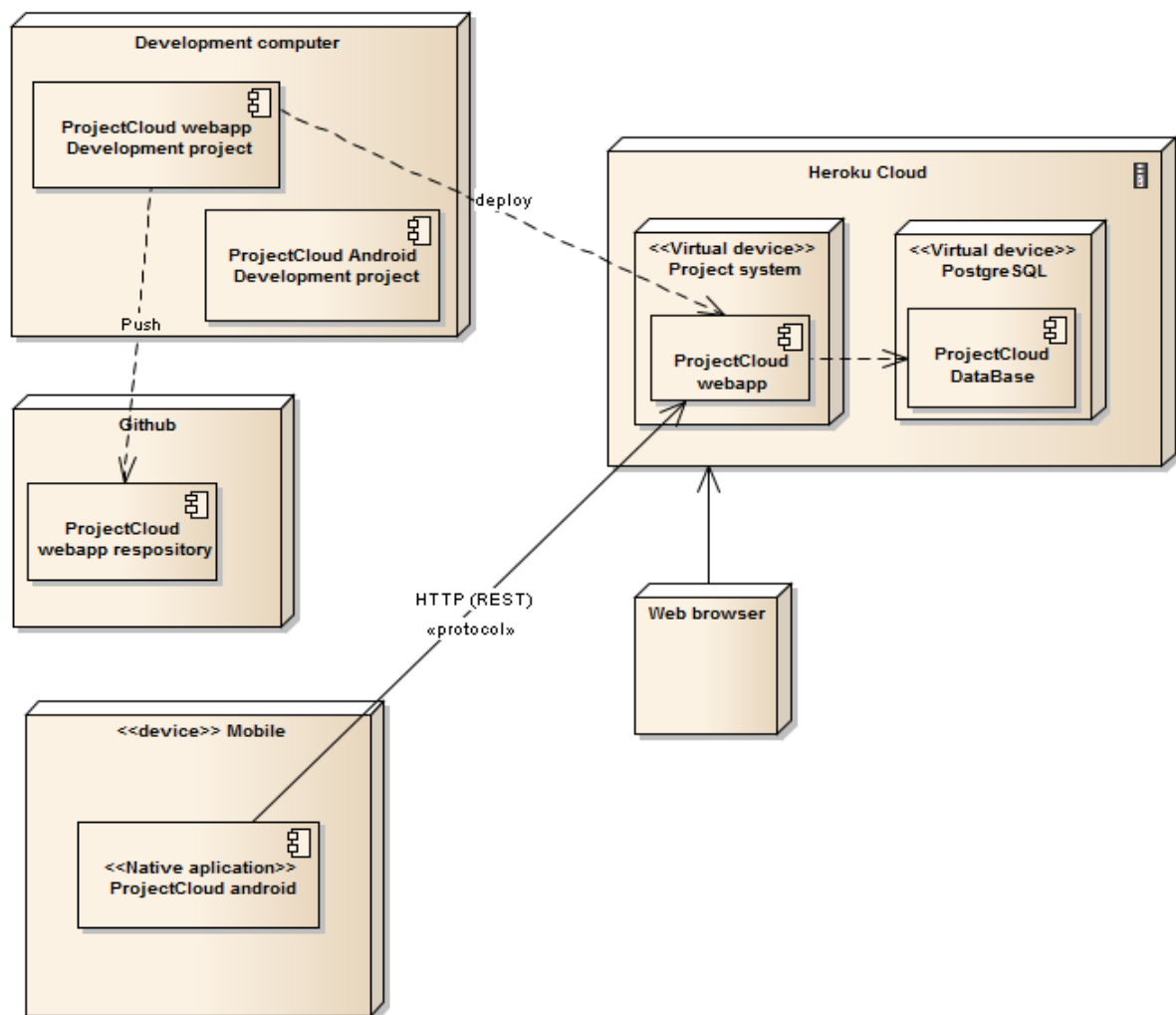


Figura 18 Modelo de despliegue

## 14 Implementación y desarrollo

### 14.1 Iteración 1: Estructura de la aplicación

#### 14.1.1 Planificación de la iteración

En esta primera planificación de la iteración he decidido que las tareas a realizar serán las necesarias para tener ya el esqueleto de la aplicación y todo lo necesario para en la siguiente iteración empezar ya con algo que el usuario pueda ver, estas tareas incluyen:

- Creación de una base de datos en MySQL.
- Descargar del editor de texto SublimeText el más apoyado por la comunidad de Ruby on Rails.
- Creación del proyectos en Ruby on Rails.
- Creación de todos los controladores (sin su código, solo cabeceras).
- Creación de todos los modelos (ídem anterior).
- Creación de todas las vistas básicas (ídem anterior, se incluyen vistas new, index, etc, pueden posteriormente ser eliminadas por su inutilización).
- Crear conexión entre la aplicación y la base de datos.
- Añadir la aplicación a GitHub.
- Crear las clases test para el testeo.

Debido a que es la primera iteración los tiempos estimados al principio no se ven modificados por tanto quedarán:

40h de diseño e implementación y 4h de testing.

#### 14.1.2 Planificación de entregas.

Hecha ya la planificación de la iteración he decidido que al final de la iteración tendré que entregar:

- Todos los modelos, controladores y vistas necesarias ya creadas aunque sin código.
- La base de datos creada correctamente en MySQL.
- La conexión con la base de datos.
- Todas las clases tipo test por cada uno de los modelos, controladores y vistas.
- Todos los archivos se subirán a GitHub.

### 14.1.3 Diseño e implementación

Para empezar el desarrollo lo primero que he hecho es descargar el software necesario, primero he descargado SublimeText2 de su página oficial <http://www.sublimetext.com/>, un editor de textos ya mencionado muy útil para programar en Ruby on Rails, a continuación he descargado también MySQL workbench también de su página oficial <http://www.mysql.com/products/workbench/>, para poder ver la base de datos en todo momento y poder por tanto comprobar su correcto funcionamiento y por último la última versión de Ruby on Rails disponible en este momento en su web <http://rubyonrails.org/>.

He empezado creando el proyecto de Ruby on Rails, ya por defecto genera unos ficheros que debo ir modificando para la construcción de la aplicación. A continuación he creado los modelos para empezar a modelar la base de datos, en esta fase me ha surgido la duda de si usar scaffold <sup>29</sup>una característica muy conocida de Ruby on Rails, sin embargo, debido a que crearía operaciones de consulta a muchas variables que no interesa que los usuarios puedan llegar y que se necesita control sobre lo que un usuario puede o no puede hacer en la aplicación en función de sus privilegios, he decidido no usar esta herramienta y crear todo desde cero.

Una vez creados todos los modelos necesarios he modificado el fichero de conexión con la base de datos para poder conectarme a ella y posteriormente crear ya la primera versión de la base de datos que utilizará la aplicación. Hecho lo anterior he creado los modelos y vistas necesarios junto a sus archivos test, Ruby on Rails hace esto automáticamente cuando se crea un controlador, además añade los ficheros de estilo necesarios para cada controlador.

Una vez acabado modifiqué el fichero de rutas para crear todas las rutas necesarias por el momento para acceder a la aplicación, es decir, los métodos GET, POST, DELETE y PUT, de manera que se pueda acceder ya a cada uno de ellos.

---

<sup>29</sup> Herramienta de Ruby on Rails que crea automáticamente controladores y vistas estándares para el modelo que se desea, para más información: <http://dummyonrails.lacoctelera.net/post/2007/06/08/que-es-scaffold-ruby-on-rails>. [19]

Posteriormente se pasa a la decisión de las gemas <sup>30</sup>que se quieren utilizar, en Ruby todos los frameworks o herramientas que se añaden son conocidos como gemas, que son software de soporte para añadir funcionalidades a nuestra aplicación, Rails es una gema de Ruby, las gemas que he decidido utilizar son:

- Mysql2: Ayuda para crear la comunicación entre Rails y MySQL.
- JQuery-rails: Sirve para ejecutar código JQuery dentro de la aplicación.
- Turbolinks: Gema muy interesante, sirve para hacer mucho más rápido el seguimiento de links por parte del servidor en una aplicación. Más información en: <https://github.com/rails/turbolinks>.
- Devise: Gema muy útil para la gestión de usuarios y sesiones dentro de una aplicación.
- Formtastic: Esta gema facilita la creación de formularios para el usuario.
- JQuery-ui-rails: Sirve para ejecutar código JQuery-UI y añadir sus estilos.
- Rails\_12factor y pg: Gemas necesarias para colgar la aplicación en los servidores Heroku.

Una vez he tenido todo hecho he pasado a la ejecución de la aplicación en la cual no ha habido grandes problemas, por tanto falta enviar todo el código a un nuevo repositorio <sup>31</sup>de Github y finalizar esta fase.

#### 14.1.4 Testing

He comprobado que la base de datos se ha creado correctamente y que la aplicación se ejecuta correctamente, en esta iteración no hay mucho más que testear.

## 14.2 Iteración 2: Registro e identificación de usuarios

### 14.2.1 Planificación de la iteración

En la planificación anterior se ha podido observar que no son necesarias 6 horas para ambas planificaciones así que el tiempo sobrante se destinara en diseño e implementación, efectivamente prácticamente se han usado las 40 horas de diseño e implementación, pero han sobrado 3 horas de testing así que podemos concluir que los

---

<sup>30</sup> Frameworks con funcionalidades más específicas que se añaden a Ruby, para más información: <http://rubygems.org/>

<sup>31</sup> Sitio web donde se almacena código de forma que este sea entendible.

tiempos que se habían establecido al principio del proyecto se pueden cumplir si añadimos el tiempo sobrante a las planificaciones a diseño y tenemos en cuenta que esta primera iteración apenas tenía testing.

En esta iteración se realizará el trabajo necesario para el registro e identificación de usuario, por tanto en esta iteración se realizarán las siguientes tareas:

- Se implementarán el modelo y controlador usuario.
- Se implementarán las vistas para todo lo relacionado con el usuario.
- Se implementará la página principal de la aplicación.
- Se implementará la posibilidad de recuperar la contraseña vía email.
- Se implementará el envío de un correo de confirmación de cuenta (no implica que la cuenta tenga que ser confirmada, simplemente es un email informativo).
- Se añadirán los atributos necesarios a la base de datos para el correcto funcionamiento del modelo usuarios.

#### 14.2.2 Planificación de entregas

En esta iteración se debe entregar:

- El modelo usuario implementado.
- El controlador usuario implementado.
- Las vistas necesarias para la interacción con el usuario.
- La tabla usuarios de la base de datos.

#### 14.2.3 Diseño e implementación

Para manejar los usuarios de la aplicación se usa la gema Devise muy útil para este tipo de funciones.

Lo primero que realicé es la creación del modelo usuario y sus atributos en la base de datos, gracias a la gema Devise estos procesos están bastante automatizados, una vez creado el modelo he creado las vistas necesarias y la página principal con los formularios tanto para el registro como para la identificación.

Para crear la primera vista he decidido que se utilizara Bootstrap<sup>32</sup>, esta herramienta es creada por los desarrolladores de Twitter y ofrece gran ayuda a la hora de realizar interface para el usuario y que además sean cross-platform <sup>33</sup>que era uno de los objetivos de la aplicación.

Por último hay que crear el servicio de mensajería para que se puedan enviar correos de confirmación y para restablecer la contraseña, por el momento se usara la dirección: [projectcloud.mailer@gmail.com](mailto:projectcloud.mailer@gmail.com), más adelante se estudiará tener un dominio propio.

#### 14.2.4 Testing

En esta fase he hecho pruebas sobre el registro e inicio de sesión su correcta creación en la base de datos, si el servicio mailing funciona y si las restricciones de contraseña (más de 8 caracteres) funcionan.

### 14.3 Iteración 3: Creación y edición de proyectos

#### 14.3.1 Planificación de la iteración

La anterior iteración ha funcionado correctamente y los tiempos han sido bastante precisos, por tanto, para esta iteración la repartición de horas queda igual a la anterior iteración. Durante esta iteración está previsto hacer las siguientes tareas:

- Se implementarán el modelo y controlador Proyecto.
- Se implementarán el modelo y controlador Assignment.
- Se implementarán las vistas para todo lo relacionado con el Proyecto.
- Se implementarán las vistas para todo lo relacionado con el Assignment.
- Se implementarán los primeros modelos de botones que se usaran durante el desarrollo de la aplicación.
- Se empezará a implementar Bootstrap para facilitar el cross-platform.
- Se añadirán a la base de datos los atributos necesarios para el correcto funcionamiento.

---

<sup>32</sup> Herramienta de ayuda al desarrollo de interface de usuario, para más información: <http://getbootstrap.com/>.

<sup>33</sup> Que sea compatible con todo tipo de dispositivos, ya sean tablets, móviles u ordenadores de sobremesa etc.

### 14.3.2 Planificación de entregas

En esta iteración se debe entregar:

- El modelo Proyecto implementado.
- El controlador Proyecto implementado.
- Las vistas necesarias para la interacción con el Proyecto.
- La tabla Proyectos de la base de datos.
- El modelo Assignment implementado.
- El controlador Assignment implementado.
- Las vistas necesarias para la interacción con el Assignment.
- La tabla Assignment de la base de datos.

### 14.3.3 Diseño e implementación

He empezado por crear el modelo proyectos y añadir todas las restricciones de integridad que ha de tener un proyecto, posteriormente he creado el controlador y de nuevo he obviado el uso de Scaffolding. Por último se crea la migración para añadir los campos necesarios de la tabla proyectos.

En esta iteración he decidido empezar a aplicar las utilidades de Bootstrap para el cross-platform, se ha dejado código CSS en un documento para poder acceder desde toda la aplicación preparado para usar posteriormente. He diseñado también todos los botones que por el momento se creen que van a ser útiles, he vigilado en los derechos de uso de los botones para evitar conflictos de derechos de autor en el futuro, para ello he utilizado la página web <http://www.creativefreedom.co.uk/free-icons/> que ofrece una colección de botones de uso gratuito.

### 14.3.4 Testing

He hecho todas las pruebas necesarias sobre el modulo proyectos y funciona correctamente. La manera de hacer testing en Rails es por pruebas unitarias y gracias a la manera de funcionar de Rails se generan automáticamente las clases donde nosotros solo tenemos que crear las pequeñas operaciones unitarias que se ejecutarán cuando las necesitamos e informan acerca de las pruebas que la aplicación no da por validas, para ello las operaciones ejecutan una operación, por ejemplo, para crear un proyecto nuevo, una vez ejecutada le decimos a Rails que compare ese valor con el esperado por

nosotros, si son iguales, Rails dará la prueba por válida, estas operaciones pueden ser tan complicadas como uno quiera, sin embargo, solo deben interactuar con una operación, de caso contrario no serían pruebas unitarias, todo el testing de la aplicación a partir de aquí se ha implementado de la misma manera, para más información Rails tiene una página dedicada a la implementación de testing: <http://guides.rubyonrails.org/testing.html>.

## 14.4 Iteración 4: Creación y edición de tareas

### 14.4.1 Planificación de la iteración

La iteración anterior ha funcionado correctamente y se ha desarrollado con el tiempo planificado, por tanto, puesto que esta iteración es bastante más sencilla que la anterior se añadirán más tareas de las previstas con anterioridad. Durante esta iteración las tareas previstas son:

- Se implementarán el modelo y el controlador Tarea.
- Se implementarán las vistas para todo lo relacionado con la Tarea.
- Se implementarán el modelo y el controlador Checklist.
- Se implementarán las vistas para todo lo relacionado con la Checklist.
- Se implementarán el modelo y el controlador Deliverable.
- Se implementarán las vistas para todo lo relacionado con la Deliverable.
- Se implementarán el modelo y el controlador Milestone.
- Se implementarán las vistas para todo lo relacionado con la Milestone.
- Se implementarán el modelo y el controlador Assignment\_task.
- Se implementarán las vistas para todo lo relacionado con la Assignment\_task.
- Se añadirán a la base de datos los atributos necesarios para el correcto funcionamiento.

### 14.4.2 Planificación de entregas

En esta iteración se debe entregar:

- El modelo Tarea implementado.
- El modelo Checklist implementado.
- El modelo Deliverable implementado.

- El modelo Milestone implementado.
- El modelo Assignment\_task implementado.
- El controlador Tarea implementado.
- El controlador Checklist implementado.
- El controlador Deliverable implementado.
- El controlador Milestone implementado.
- El controlador Assignment\_task implementado.
- Las vistas necesarias para la interacción con la Tarea.
- Las vistas necesarias para la interacción con el Checklist.
- Las vistas necesarias para la interacción con el Deliverable.
- Las vistas necesarias para la interacción con el Milestone.
- La tabla Tarea de la base de datos.
- La tabla Checklist de la base de datos.
- La tabla Deliverable de la base de datos.
- La tabla Milestone de la base de datos.
- La tabla Assignment\_task de la base de datos.

#### 14.4.3 Diseño e implementación

He empezado por crear todos los modelos mencionados y sus relaciones, también he añadido todas las restricciones de integridad necesarias, posteriormente he creado todos los modelos sin el uso de Scaffolding. Por último se crea la migración para añadir los campos necesarios de las tablas.

He implementado también funciones para determinar el rol que tiene un usuario en un proyecto y determinar los permisos que se pueden tener, estas funciones han sido implementadas en helpers para que todos los controladores puedan usarlas.

#### 14.4.4 Testing

He hecho todas las pruebas necesarias sobre los módulos y funcionan correctamente.

## 14.5 Iteración 5: Creación de deadlines e hitos

### 14.5.1 Planificación de la iteración

La iteración anterior ha cumplido con la planificación, por tanto esta iteración puesto que estaba previsto desarrollar los Milestones y relaciones que ya han sido implementadas se aprovechará para desarrollar el diagrama de Gantt y las notificaciones.

- Se implementarán el modelo y controlador Notification.
- Se implementarán el modelo y controlador User\_notification.
- Se implementaran las vistas necesarias para todo lo relacionado con las notificaciones.
- Se implementara la vista Gantt que creara los diagramas necesarios.

### 14.5.2 Planificación de entregas

En esta iteración se debe entregar:

- El modelo Notification.
- El modelo User\_notification.
- El controlador Notification.
- El controlador User\_notification.
- Las vistas necesarias para ver las notificaciones
- La tabla Notifications de la base de datos.
- La tabla User\_notification de la base de datos.
- La vista Gantt que incluye todo lo necesario para crear el diagrama de Gantt.

### 14.5.3 Diseño e implementación

En esta iteración he empezado por desarrollar el modelo Notification y User\_notifications, también he creado sus controladores, en los controladores he añadido la lógica para recibir notificaciones, la manera de funcionar es la siguiente:

- 1) Cuando cualquier controlador hace un cambio crea una fila en la tabla Notifications del mismo proyecto.
- 2) Cuando un usuario entra en /projects se crea la relación entre el User y las Notifications, buscando las que todavía no tiene y se pone con is\_notified = 0.

- 3) Se borran todas las notificaciones y relaciones que tienen más de 1 semana de antigüedad (aunque el usuario no las haya visto, se ha decidido darlas por obsoletas)
- 4) Se muestran aquellas notificaciones cuyos `is_notified = 1`.

Implementada ya la vista que muestra las notificaciones se pasa a implementar el diagrama de Gantt.

Para implementar la vista del Diagrama de Gantt, he hecho uso de un helper para no cargar la vista de código. Se ha mirado otras webs como [www.ganttter.com](http://www.ganttter.com) para ver como implementan los diagramas de Gantt, sin embargo, el usar tablas de html como es el caso de la mayoría no lo considero un método eficaz, ya que desde mi parecer el uso de las tablas html no tiene que ser extrapolado para cualquier cosa. He desarrollado, por tanto, el diagrama de Gantt usando etiquetas `<div>` calculando con los pixeles el tamaño de las barras de tareas y donde indicar las dependencias, el sistema tarda menos en cargar que el de otras webs, sin embargo, habría que comprobar si es debido al uso de los `<div>`, queda pendiente para el futuro comprobar la eficiencia del desarrollo de Gantt sin el uso de tablas.

#### 14.5.4 Testing

He hecho todas las pruebas necesarias para las notificaciones y el diagrama de Gantt y funciona correctamente.

### 14.6 Iteración 6: Mejoras en la interface

#### 14.6.1 Planificación de la iteración

La iteración anterior ha cumplido con la planificación a pesar de los cambios notables que ha sufrido, esta iteración siguiente no tiene mucho que ver con lo hecho ya que se trabajara a fondo con el front-end y de momento solo se ha hecho lo justo para que la aplicación sea más o menos visible, se buscará por tanto construir una interface lo más inteligible y fácil de usar para el usuario.

- Se implementara toda la lógica front-end

#### 14.6.2 Planificación de entregas

En esta iteración se debe entregar:

- Todas las vistas de la aplicación.

- Todos los modulos css necesarios.

### 14.6.3 Diseño e implementación

La idea de la interfaz era hacerla a partir de cajas que diferencien las diferentes partes de la vista y con una barra de navegación en la parte superior, lo primero que he hecho es definir el layout o modelo básico que seguirá la aplicación, es decir, la parte de la vista que aunque cambiemos de pantalla no varía (barra de navegación), una vez hecho, he empezado a crear los códigos CSS necesarios para los links, cajas, botones etc. Recordemos que como fue explicado en la iteración 2 y 3 para desarrollar las vistas se usan botones de licencia libre y también se está usando BootStrap. Durante el desarrollo de esta iteración he tenido la idea de usar los colores de fondo tanto de la tareas como en el diagrama de Gantt, para marcar en qué estado esta cada tarea, de esta manera el usuario, de forma muy intuitiva, puede ver en qué estado se encuentra una tarea.

### 14.6.4 Testing

El testing desarrollado es comprobar en diferentes navegadores como son: Internet Explorer, Google Chrome, Mozilla Firefox y Opera el correcto visionamiento de la interface.

Por otro lado he hecho un pequeño test de usabilidad con amigos y familiares para comprobar que realmente la aplicación es usable, he podido ver que exceptuando algunas funcionalidades en algún caso aislado, la mayoría de usuarios han sabido hacer todo aquello que se les ha pedido sin dificultad ni ayuda.

## 14.7 Iteración 7: Conectividad desde Android a la base de datos

### 14.7.1 Planificación de la iteración

Dando por finalizada la parte web, en esta iteración está previsto empezar a diseñar una interface que vía API RestFul, conecte la aplicación Android con la base aplicación web, previamente a esta decisión he valorado diferentes tipos de conexión:

1. Conectar desde la aplicación Android a la base de datos obviando la aplicación web.
2. Conectar mediante protocolos SSL y HTTPS
3. Desarrollando una API RestFul

La primera opción parecía viable, pero no la más atractiva, es decir, normalmente las aplicaciones de móviles suelen conectarse vía API por varios hechos, entre ellos los que se han valorado son:

1. La independencia a la base de datos, aunque en un futuro la base de datos cambie, la aplicación Android se mantendrá estable puesto que no mantiene una conexión directa con la base de datos
2. La necesidad de implementar una interface que conecte directamente con la base de datos deja de ser necesaria y por tanto a primera vista puede parecer menos código que programar, aunque haya que hacer la API RESTFul en la parte web.
3. La base de datos guarda las contraseñas encriptadas por la gema devise y yo no conozco el método de desencriptación, así que este es un problema eliminatorio de esta opción.
4. Más adelante y practicando un poco he descubierto que la base de datos rechaza de entrada las peticiones a la base de datos desde Android y habría que averiguar si este problema tiene solución posible.

Dicho esto la primera opción quedaba descartada, así que pasemos a la segunda.

La segunda opción quedo descartada por que requiere de certificados y otros elementos que pueden ser muy tediosos para el usuario.

Por tanto se decide construir una API RESTFul y conectar Android mediante esta API, las tareas por tanto para esta iteración son:

- Creación de una API RESTFul en la aplicación web.
- Crear una interface con Java que posteriormente sea utilizada por Android para iniciar sesión y recibir notificaciones.

#### 14.7.2 Planificación de entregas

Al final de la iteración por tanto se han de hacer las siguientes entregas:

- Los controladores necesarios para la API.
- Código Java para recibir el inicio de sesión.
- Código Java para recibir las notificaciones.

### 14.7.3 Diseño e implementación

En la implementación de la API me he encontrado con bastantes problemas, el principal ha residido en la gema devise, dicha gema implementaba un token para la autenticación vía API, sin embargo, en la versión más reciente este token ha sido eliminado porque dicen no poder asegurar la seguridad del sistema, en este blog se explica más detalladamente: <http://blog.plataformatec.com.br/2013/08/devise-3-1-now-with-more-secure-defaults/>, debido a esto he tenido que implementar mi propio token y construir un modo de autenticación hacia la API, de otro lado, para que la API funcione correctamente y mantener un control de versiones he tenido que crear nuevos controladores que heredan de un controlador API, yo creía que esto no sería necesario a primeras. Estos hechos han llevado bastante más tiempo del previsto y podían afectar a las iteraciones que faltaban para terminar el proyecto, para evitar esto he decidido no centrarme demasiado en la seguridad y simplemente hacer una parte más funcional que posteriormente si queda tiempo ya se trabajará la seguridad.

El trabajo hecho, por tanto, ha sido añadir a la base de datos un token para los usuarios, este token será enviado como respuesta del método `api/users/sign_in`, que ha de ir acompañado de los siguientes parámetros necesarios en formato json, esto devolverá si la autenticación ha sido correcta y el token que autentifica al usuario. También he implementado la petición para recibir notificaciones y el código java que genera las peticiones y obtiene los resultados. De momento el usuario queda registrado con el uso de una cookie se tendría que valorar la opción de usar el token para mantener siempre la sesión iniciada y no solo por el tiempo de vida de la cookie.

### 14.7.4 Testing

Debido a los retrasos de la iteración no he podido hacer mucho testing, simplemente he comprobado que las peticiones funcionen y envíen los resultados correspondientes para un simple usuario, más adelante, si queda tiempo comprobaré el correcto comportamiento de las peticiones y se indagará sobre temas de seguridad y cuál es la mejor manera de autenticarse vía API.

## 14.8 Iteración 8: Creación aplicación Android

### 14.8.1 Planificación de la iteración

En la iteración anterior me he encontrado bastantes problemas y he dejado, por tanto, en un estado de finalización que simplemente permita poder continuar con el desarrollo de la aplicación Android, para que el resto del proyecto no se vea afectados por estos problemas, de alguna manera, he logrado hacer lo mínimo para que el proyecto no se vea comprometido y pueda seguir con la planificación establecida, como se menciona anteriormente si sobraré tiempo de la última iteración se intentarían remediar estos problemas.

En esta iteración el objetivo es desarrollar la aplicación Android, se integrará el código generado en la anterior iteración y por tanto solo hace falta crear las pantallas, y el código necesario para enviar notificaciones al usuario, por ende, las tareas a desarrollar son las siguientes:

- Se implementaran las pantallas de la aplicación Android.
- Se implementara el código necesario para enviar notificaciones al usuario.

### 14.8.2 Planificación de entregas

En esta iteración se debe entregar:

- La aplicación Android en un estado funcional.

### 14.8.3 Diseño e implementación

Se ha empezado la implementación copiando el código de que usa la API y creando sus clases, posteriormente se ha creado el menú de opciones de las notificaciones, donde el usuario podrá decidir entre otras cosas, cada cuanto quiere que se compruebe si hay actualizaciones.

El siguiente paso ha sido crear un servicio que permita a la aplicación pedir los datos al servidor cada cierto tiempo (definido por el usuario) y mostrar notificaciones cuando sea necesario.

#### 14.8.4 Testing

En esta iteración no hay mucho testing, simplemente se ha comprobado que el servicio funciona cada cierto tiempo establecido y que las notificaciones se muestran.

## 14.9 Iteración 9: Finalización de la aplicación Android

### 14.9.1 Planificación de la iteración

La iteración anterior ha cumplido con la previsión establecida.

En esta última iteración se acabarán de modificar las notificaciones para hacerlas más agradables al usuario y añadir un poco de diseño en las pantallas para el agrado del usuario también.

### 14.9.2 Planificación de entregas

En esta iteración se debe entregar:

- Código de interfaz de pantallas.
- Código de interfaz de notificaciones.

### 14.9.3 Diseño e implementación

En esta iteración he mejorado las interfaces mencionadas anteriormente, por tanto, no hay mucho que añadir, el único hándicap ha sido el tiempo, ya que el diseño siempre requiere dedicación.

### 14.9.4 Testing

Solo he comprobado que las pantallas se muestran correctamente y la aplicación funciona bien en distintos dispositivos Android emulados.

## 15 Aspectos legales

### 15.1 Base de datos

Debido a que la aplicación usa bases de datos y además requiere del registro de usuarios hay varios factores legales que hay que tener en cuenta.

El primero es que el usuario debe aceptar un texto con la política de privacidad y el aviso legal cuando se registre en la aplicación, ambos textos se encuentran en los anexos de una forma genérica.

Por último debemos seguir los pasos indicados en esta página web: <https://www.agpd.es/portalwebAGPD/index-ides-idphp.php>, que es la web de la agencia de protección de datos española para registrar que vamos a guardar datos en nuestra base de datos y para que los utilizaremos [15]

### 15.2 Cookies

Una cookie es un pequeño fragmento de texto que los sitios Web envían al navegador y que se almacenan en el terminal del usuario, el cual puede ser un ordenador personal, un teléfono móvil, un tablet, etc. Estos archivos permiten que el sitio Web recuerde información sobre su visita, como el idioma y las opciones preferidas, lo que puede facilitar su próxima visita y hacer que el sitio resulte más útil al personalizar su contenido. Las cookies desempeñan un papel muy importante, al mejorar la experiencia de uso de la Web.

En el caso de la aplicación web se usan cookies para recordar la sesión del usuario.

Hace unos años el gobierno español obliga a las páginas web a informar al usuario sobre el uso de cookies por tanto al principio la página web avisará con un texto parecido a este:

"Utilizamos cookies propias y de terceros para mejorar nuestros servicios y mostrarle publicidad relacionada con sus preferencias mediante el análisis de sus hábitos de navegación. Si continúa navegando, consideramos que acepta su uso. Puede obtener más información, o bien conocer cómo cambiar la configuración, en nuestra Política de cookies"

Y también contendrá un texto sobre la política de cookies que se puede encontrar en el anexo.

## 16 Visión al futuro

De cara al futuro de la aplicación se han pensado diferentes funcionalidades o cambios que podrían ayudar de forma positiva al buen funcionamiento de la aplicación, se han dividido en tres categorías según su coste en desarrollo.

### Coste bajo:

#### **Modificar formulario crear tareas:**

Actualmente las tareas se crean en un formulario en una pantalla aparte y cuando se da la tarea por creada el sistema redirige al usuario a una pantalla donde se muestran todas las tareas. Lo más habitual en un usuario que crea un proyecto es que ya tenga las tareas pensadas y por ende las quiera crear seguidas por tanto el cambio estaría en hacer que desde la misma pantalla se puedan crear las tareas y visualizar las ya creadas de manera que solo se refresque la misma pantalla en el momento que se crea la tarea para evitar tener que cargar el formulario si quisiéramos añadir otra, mejoraría la eficiencia y reduciría la fatiga del usuario a la hora de crear tareas.

#### **Añadir tareas por hacer:**

Sería interesante que el usuario al entrar en la aplicación pudiera ver las tareas que tiene pendiente de realizar, para ellos habría que implementar una función nueva en el controlador tarea que devuelva aquellas tareas que el usuario tiene asignadas y no ha terminado.

### Coste medio:

#### **Subir deliverables a la aplicación:**

Actualmente los deliverables solo indican que se necesita y en que formato. Sin embargo, al usuario le puede ser muy útil encontrar ya el deliverable en cuestión subido a la red cuando se ha entregado y no tener que recurrir a otros servicios para adquirir el deliverable, ahorraría tiempo a los usuarios, hay que tener en cuenta pero, que este cambio puede ser costoso ya que requeriría de espacio extra en el servidor.

### Coste alto:

#### **Exportar el proyector en pdf:**

Actualmente el proyecto solo se puede ver desde la web. Puede que el usuario necesite imprimir el proyecto o presentarlo email o circunstancias parecidas en las que necesite tener todos los datos sin necesidad de recurrir a la aplicación, por tanto, se diseñaría un nuevo caso de uso que sería Exportar a pdf, en el cual el usuario obtendría toda la información del proyecto en pdf para poder compartir con quien desee.

**Implementar mensajería:**

Actualmente la aplicación web no dispone de ningún tipo de mensajería, si se quiere comunicar con un usuario la aplicación web redirige al usuario a su aplicación de mensajería estándar. Puede ser más cómodo para los usuarios implementar un servicio de mensajería para tener comunicación más directa entre los miembros de un proyecto.

## 17 Conclusiones

Cuando empecé este proyecto, simplemente tenía una pequeña idea que era desarrollar un gestor de proyectos web porque consideraba que podía ser una aplicación útil que se podría utilizar en el ámbito estudiantil en el que yo mismo me veía inmerso, sin embargo, era solo eso, una idea.

Cuando empecé con GEP (Gestión de Proyectos), pude ir viendo como esa idea inicial cogía forma y empezaban a aparecer las primeras preguntas importantes: ¿Por qué hay necesidad de crear un gestor de proyectos web?, ¿Qué lo diferenciará de los demás?, ¿Será rentable para mí?, etc. Todas estas preguntas han hecho que aprenda a tener una vista más global sobre cómo desarrollar un proyecto y ver más allá de lo que hay entre las paredes de la informática, me ha servido para aprender cómo empezar desde una idea e ir la convirtiendo poco a poco en un proyecto.

Nunca me había visto inmerso en un proyecto de esta envergadura, por lo tanto como era de esperar en algunos momentos me he encontrado con diferentes problemas, como por ejemplo: ¿Qué lenguaje escoger?, ¿Cómo escoger servidores?, etc. Estas preguntas que durante la carrera habían aparecido, pero de alguna manera eran respondidas con facilidad, esta vez al ser expuestas en un proyecto real y con total libertad de elección se convertían en pequeños retos que asumir, sin embargo, he podido dar respuesta a todas ellas de la manera que he considera correcta y oportuna.

Durante el desarrollo del proyecto también cabe destacar la importancia de una buena planificación, el “éxito” desde mi punto de vista, ha recaído en gran parte a una correcta planificación que ha permitido llevar un control en todo momento de cómo iba avanzando el proyecto y ha permitido en algunos casos más críticos tomar decisiones más flexibles que permitiera evadir conflictos, en general me llevo conmigo la importancia de un planificación bien hecha.

Como conclusión del proyecto en general y considerando que nunca antes había hecho aplicaciones web tan grandes, aunque si tenía conocimientos sobre Android, merece decir que el resultado ha sido muy favorable, el resultado ha sido una aplicación web funcional, que excepto por algún tema legal como añadir una política de privacidad, puede ponerse en funcionamiento desde ya, por todo esto me siento muy orgulloso de

cómo ha ido evolucionando mi idea hasta convertirse en un proyecto con cara y ojos, todo esto no hubiera sido posible sin la ayuda de Carles Farré quien siempre ha estado dispuesto a dar una mano cuando ha habido problemas.

Para finalizar las conclusiones he cogido los objetivos descritos al principio y he justificado su cumplimiento.

- ✓ El gestor de proyectos será gratuito para los usuarios, de manera que los ingresos se consigan mediante otras vías como podrían ser la publicidad o cobrando solo por el uso empresarial.

**Justificación:** La aplicación es gratuita.

- ✓ Tendrá una disponibilidad 24/7 y desde cualquier lugar desde el primer momento que sea disponible para el público.

**Justificación:** La aplicación se aloja en Heroku y sus servidores tienen esa disponibilidad.

- ✓ La aplicación será accesible desde dispositivos móviles y tablets desde el primer momento que sea disponible para el público.

**Justificación:** La interface se ha hecho con Bootstrap, por tanto, puede verse correctamente desde cualquier dispositivo, además de haberse creado una aplicación Android.

- ✓ La aplicación seguirá buenos patrones de diseño en todo momento y por tanto será usable.

**Justificación:** La interface es sencilla, los botones son grandes y fácilmente se puede ver cuál es su función.

- ✓ La aplicación será sencilla de usar y su curva de aprendizaje será mínima, un nuevo usuario no debería tardar más de 30 minutos en aprender el funcionamiento completo del software después de leer la ayuda.

**Justificación:** A pesar de no poder probar la aplicación con la muestra de usuarios necesaria para desarrollar un estudio estadístico, se hizo un pequeño test de usabilidad con pocos usuarios a los cuales se les animaba a hacer algunas funcionalidades de la aplicación, en este test se pudo ver que los usuarios no tenían problemas para llevar a cabo lo que se les pedía ni necesitaban ningún

tipo de ayuda, por tanto, concluimos que la aplicación tiene una curva de aprendizaje baja.

## 18 Bibliografía

- [1] L. Talent, «Lancetalent,» 24 1 2013. [En línea]. Available: <http://www.lancetalent.com/blog/las-10-mejores-herramientas-para-la-gestion-de-proyectos-online/>. [Último acceso: 15 1 2014].
- [2] R. Suttle, «Ehowenespanol,» [En línea]. Available: [http://www.ehowenespanol.com/roles-equipo-proyecto-info\\_208138/](http://www.ehowenespanol.com/roles-equipo-proyecto-info_208138/). [Último acceso: 17 1 2014].
- [3] «Gestor de proyectos Active Collab,» [En línea]. Available: <https://www.activecollab.com/>. [Último acceso: 10 1 2014].
- [4] «Gestor de proyectos Assembla,» [En línea]. Available: <https://www.assembla.com/home>. [Último acceso: 10 1 2014].
- [5] «Gestor de proyectos Base Camp,» [En línea]. Available: <https://basecamp.com/>. [Último acceso: 10 1 2014].
- [6] «Gestor de proyectos Central Desktop,» [En línea]. Available: <http://www.centraldesktop.com/>. [Último acceso: 10 1 2014].
- [7] «Gestor de proyectos Atlassian,» [En línea]. Available: <https://www.atlassian.com/software/confluence>. [Último acceso: 10 1 2014].
- [8] «Gestor de proyectos Kapost,» [En línea]. Available: <http://kapost.com/>. [Último acceso: 10 1 2014].
- [9] «Gestor de proyectos Producteev,» [En línea]. Available: <https://www.producteev.com/>. [Último acceso: 10 1 2014].
- [10] «Gestor de proyectos TeamBox,» [En línea]. Available: <http://teambox.com/>. [Último acceso: 10 1 2014].

- [11] «Gestor de proyectos TeamLab,» [En línea]. Available: <http://www.teamlab.com/es/>. [Último acceso: 10 1 2014].
- [12] «Gestor de proyectos Asana,» [En línea]. Available: <https://asana.com/>. [Último acceso: 10 1 2014].
- [13] miblogtecnico, «miblogtecnico.wordpress,» 2012 07 16. [En línea]. Available: <http://miblogtecnico.wordpress.com/tag/asp-net-mvc-2/>. [Último acceso: 25 02 2014].
- [14] Dosideas, «Dosideas,» 13 10 2008. [En línea]. Available: <http://www.dosideas.com/noticias/java/314-introduccion-a-los-servicios-web-restful.html>. [Último acceso: 9 3 2014].
- [15] «Gestor de proyectos Active Collab,» [En línea]. Available: <https://www.activecollab.com/>. [Último acceso: 10 1 2014].
- [16] «Amazon web services,» [En línea]. Available: <http://aws.amazon.com/es/web-mobile-social/>. [Último acceso: 25 1 2014].
- [17] S. P. Moreno, Ruby on Rails Desarrollo práctico de aplicaciones web, Madrid: RC Libros, 2013, p. 453.
- [18] J. D. Cauchat, El gran libro de HTML5, CSS3 y Javascript, Barcelona: Marcombo, 2012, p. 354.
- [19] Dummyonrail, «Dummyonrail,» 08 06 2007. [En línea]. Available: <http://dummyonrails.lacoctelera.net/post/2007/06/08/que-es-scaffold-ruby-on-rails>. [Último acceso: 3 3 2014].
- [20] «RubyGems,» [En línea]. Available: <http://rubygems.org/>. [Último acceso: 1 3 2014].

[21] R. o. Rails, «Ruby on Rails Guides,» [En línea]. Available: <http://guides.rubyonrails.org/>. [Último acceso: 30 1 2014].

[22] «Agencia Española de Protección de Datos,» [En línea]. Available: <https://www.agpd.es/portalwebAGPD/index-ides-idphp.php>. [Último acceso: 1 5 2014].

## 19 Glosario

**24/7:** Las 24 horas del día, los 7 días de la semana.

**Backend:** Lógica de una aplicación web que recae del lado del servidor.

**Base de datos relacional:** Base de datos que guarda relación entre sus objetos.

**Cloud Computing:** Nuevo paradigma que permite ofrecer servicios de computación a través de internet, todo aquello que puede ofrecer un sistema informático se ofrece como servicio de manera que todos los usuarios puedan acceder a él.

**Cookie:** Pequeño archivo almacenado en el ordenador del cliente que contiene información necesaria para el correcto funcionamiento de la aplicación. En este caso contendrá los credenciales del usuario para recordarlo en posteriores entradas a la aplicación.

**CSS3:** Hojas de estilo cuyo uso como el nombre indica recae en dar estilo a una página web, para más información <http://www.w3schools.com/css/>

**Datos sensibles:** Datos sobre una persona o empresa que no desean que sean vistos por otros sin previa autorización.

**Deadline:** Un “deadline” es un término anglosajón que simboliza el tiempo límite, en el caso de los proyectos suele referirse a la fecha de finalización de una tarea o del proyecto en sí.

**Diagrama de Gantt:** Herramienta gráfica cuyo objetivo es mostrar el tiempo de dedicación previsto a diferentes tareas en un plazo de tiempo total determinado.

**Dropbox:** Servicio de almacenamiento en la nube, cada usuario almacena sus datos en la nube y puede acceder a ellos desde cualquier lugar en cualquier momento. Para más información: [www.dropbox.com](http://www.dropbox.com)

**EXtreme Programming (XP):** eXtreme Programming es una metodología de desarrollo orientada a la ingeniería de software, es una metodología ágil y se diferencia principalmente por dar más énfasis a la adaptabilidad que a la previsibilidad.

**Feedback:** Retro alimentación, en nuestro caso el usuario debe ir introduciendo de manera continua nueva información al software.

**Framework:** Herramienta de ayuda para el desarrollo de software.

**Google AdSense:** Servicio de Google que añade publicidad automáticamente a cualquier sitio web siempre y cuando cumpla sus condiciones de uso.

**GoogleDocs:** Servicio de Google usado para crear archivos tipo Word, Excel etc con la capacidad de ser vistos y modificados por varios usuarios al mismo tiempo.

**GitHub:** Servicio en la nube para almacenar código y hacerlo accesible a más usuarios, para más información: [www.github.com](http://www.github.com)

**Hitos:** Dentro de un proyecto, un hito se considera un conjunto de tareas cuya finalización forman parte de una parte muy importante del mismo.

**HTML5:** Lenguaje web por excelencia que interpretan todos los navegadores web más conocidos, para más información [http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)

**Interface:** Parte visual de un software, es decir, donde el usuario interactúa y ve la información.

**JavaScript:** Lenguaje web muy utilizado para la lógica del cliente por su eficiencia, para más información <http://www.w3schools.com/js/>

**jQuery UI:** Biblioteca de componentes para jQuery que añade aspectos visuales, para más información <http://jqueryui.com/>

**LOPD:** Ley orgánica de protección de datos, contiene todas las leyes que las bases de datos han de cumplir para estar acorde a la ley española.

**MVC:** Patrón modelo vista controlador, es un patrón para la programación orientado a la programación de aplicaciones web.

**MySQL:** Sistema de gestión de bases de datos relacionales, para más información <http://www.mysql.com/>

**Nube:** Representa el concepto de tener documentos o servicios en internet y de manera remota en vez de en un propio ordenador o servidor.

**PostgreSQL:** Servidor de bases de datos relacional, para más información <http://www.postgresql.org.es/>.

**Requerimientos:** Se entiende por requerimientos aquello que el software debe ser capaz de hacer.

**Ruby on Rails:** Lenguaje de programación orientado al diseño web, para más información <http://rubyonrails.org/>

**Tester:** Encargado de hacer las pruebas a un software y valorar su correcto funcionamiento.

**Token:** Caracteres pseudoaleatorios y únicos para un usuario que lo identifican en la base de datos.

## 20 Anexos

Damos fe de que este documento utiliza material de la Plantilla de Especificación de Requisitos Volere, derecho de propiedad © 1995 – 2006 The Atlantic Systems Guild Limited.

### 20.1 INFORMACION CORPORATIVA - Política de cookies

Este Web utiliza cookies propias y de terceros para ofrecerle una mejor experiencia y servicio. Al navegar o utilizar nuestros servicios el usuario acepta el uso que hacemos de las cookies. Sin embargo, el usuario tiene la opción de impedir la generación de cookies y la eliminación de las mismas mediante la selección de la correspondiente opción en su Navegador. En caso de bloquear el uso de cookies en su navegador es posible que algunos servicios o funcionalidades de la página Web no estén disponibles.

¿Qué es una cookie?

Una cookie es un pequeño fragmento de texto que los sitios Web envían al navegador y que se almacenan en el terminal del usuario, el cual puede ser un ordenador personal, un teléfono móvil, un tablet, etc. Estos archivos permiten que el sitio Web recuerde información sobre su visita, como el idioma y las opciones preferidas, lo que puede facilitar su próxima visita y hacer que el sitio resulte más útil al personalizar su contenido. Las cookies desempeñan un papel muy importante, al mejorar la experiencia de uso de la Web.

¿Cómo se utilizan las cookies?

Al navegar por este portal Web el usuario está aceptando que se puedan instalar cookies en su terminal y que nos permiten conocer la siguiente información:

- Información estadística del uso de la web.
- El 'login' del usuario para mantener la sesión activa en la web.
- El formato de la web preferente en el acceso desde dispositivos móviles.
- Las últimas búsquedas realizadas en los servicios de la web y datos de personalización de estos servicios.
- Información sobre los anuncios que se muestran al usuario.
- Información de las encuestas en las que ha participado el usuario.
- Datos de la conexión con las redes sociales para los usuarios que acceden con su usuario de Facebook o Twitter.

Tipos de cookies utilizadas

Este Web utiliza tanto cookies temporales de sesión como cookies permanentes. Las cookies de sesión almacenan datos únicamente mientras el usuario accede al Web y las cookies

permanentes almacenan los datos en el terminal para que sean accedidos y utilizados en más de una sesión.

Según la finalidad para la que se traten los datos obtenidos a través de las cookies, el Web puede utilizar:

#### Cookies técnicas

Son aquéllas que permiten al usuario la navegación a través de la página Web o aplicación y la utilización de las diferentes opciones o servicios que en ella existen. Por ejemplo, controlar el tráfico y la comunicación de datos, identificar la sesión, acceder a las partes Web de acceso restringido, recordar los elementos que integran un pedido, realizar la solicitud de inscripción o participación en un evento, utilizar elementos de seguridad durante la navegación y almacenar contenidos para la difusión de videos o sonido.

#### Cookies de personalización

Son aquéllas que permiten al usuario acceder al servicio con algunas características de carácter general predefinidas en su terminal o que el propio usuario defina. Por ejemplo, el idioma, el tipo de navegador a través del cual accede al servicio, el diseño de contenidos seleccionado, geolocalización del terminal y la configuración regional desde donde se accede al servicio.

#### Cookies publicitarias

Son aquellas que permiten la gestión eficaz de los espacios publicitarios que se han incluido en la página Web o aplicación desde la que se presta el servicio. Permiten adecuar el contenido de la publicidad para que esta sea relevante para el usuario y para evitar mostrar anuncios que el usuario ya haya visto.

#### Cookies de análisis estadístico

Son aquéllas que permiten realizar el seguimiento y análisis del comportamiento de los usuarios en los sitios Web. La información recogida mediante este tipo de cookies se utiliza en la medición de la actividad de los sitios Web, aplicación o plataforma y para la elaboración de perfiles de navegación de los usuarios de dichos sitios, con el fin de introducir mejoras en el servicio en función de los datos de uso que hacen los usuarios.

#### Cookies de terceros

En algunas páginas Web se pueden instalar cookies de terceros que permiten gestionar y mejorar los servicios ofrecidos. Como por ejemplo, servicios estadísticos de Google Analytics.

#### ¿Cómo administrar cookies en el navegador?

El usuario tiene la opción de permitir, bloquear o eliminar las cookies instaladas en su equipo mediante la configuración de las opciones del navegador instalado en su terminal:

Para ajustar los permisos relacionados con las cookies en el Navegador Google Chrome:

- Hacer clic en el menú situado en la barra de herramientas.
- Seleccionar Configuración.
- Hacer clic en Mostrar opciones avanzadas.
- En la sección "Privacidad" hacer clic en el botón Configuración de contenido.
- En la sección de "Cookies" se pueden configurar las opciones.

Para ajustar los permisos relacionados con las cookies en el Navegador Mozilla Firefox:

- En la parte superior de la ventana de Firefox hacer clic en el menú Herramientas.
- Seleccionar Opciones.
- Seleccionar el panel Privacidad.
- En la opción Firefox podrá elegir Usar una configuración personalizada para el historial para configurar las opciones.

\* Más información sobre Mozilla Firefox.

Para ajustar los permisos relacionados con las cookies en el Navegador Internet Explorer 9:

- En la parte superior de la ventana de Internet Explorer hacer clic en el menú Herramientas.
- Seleccionar la pestaña de Seguridad y utilizar la opción Eliminar el historial de exploración para eliminar las cookies. Activar la casilla Cookies y, a continuación, hacer clic en Eliminar.
- Seleccionar la pestaña de Seguridad y utilizar la Configuración mover el control deslizante totalmente hacia arriba para bloquear todas las cookies o totalmente hacia abajo para permitir todas las cookies. Tras ello, hacer clic en Aceptar.

\* Más información sobre Internet Explorer 9.

En caso de bloquear el uso de cookies en su navegador es posible que algunos servicios o funcionalidades de la página Web no estén disponibles.

En algunos navegadores se pueden configurar reglas específicas para administrar cookies por sitio Web, lo que ofrece un control más preciso sobre la privacidad. Esto significa que se puede inhabilitar cookies de todos los sitios salvo de aquellos en los que se confíe."

## 20.2 AVISO LEGAL

Denominación Social:

Nombre Comercial:

Domicilio Social:

CIF / NIF:

Teléfono:

Fax:

e-Mail:

Inscrita en el Registro:

Nombre de dominio:

Autorización administrativa:

## OBJETO

El prestador, responsable del sitio web, pone a disposición de los usuarios el presente documento con el que pretende dar cumplimiento a las obligaciones dispuestas en la Ley 34/2002, de Servicios de la Sociedad de la Información y del Comercio Electrónico (LSSI-CE), así como informar a todos los usuarios del sitio web respecto a cuáles son las condiciones de uso del sitio web.

Toda persona que acceda a este sitio web asume el papel de usuario, comprometiéndose a la observancia y cumplimiento riguroso de las disposiciones aquí dispuestas, así como a cualquier otra disposición legal que fuera de aplicación.

El prestador se reserva el derecho a modificar cualquier tipo de información que pudiera aparecer en el sitio web, sin que exista obligación de preavisar o poner en conocimiento de los usuarios dichas obligaciones, entendiéndose como suficiente con la publicación en el sitio web del prestador.

## TERCERO DE CONFIANZA Y Código de Conducta (OPCIONAL)

El prestador se encuentra adherido al código de conducta *(Indicar Código de Conducta y link al mismo)* que puede ser visualizado, descargado y/o impreso desde el link referenciado. Este código de conducta garantiza a los usuarios que el sitio web cumple con la legislación vigente y se encuentra sometido a todas las obligaciones dispuestas en dicho código.

El código de conducta (*nombre del código de conducta*) dispone de un sello de confianza, que es emitido de forma única y exclusiva para cada uno de los prestadores adheridos. El sello en posesión del prestador dispone del código de identificación [XXXXX], así como todos los datos identificativos del prestador.

El certificado será válido para el sitio web del prestador durante el plazo de 1 año desde la fecha de su emisión. Toda la información sobre la emisión del certificado y sobre las obligaciones de la entidad son accesibles por parte de los usuarios a través del siguiente link.

## Responsabilidad

El prestador se exime de cualquier tipo de responsabilidad derivada de la información publicada en su sitio web, siempre que esta información haya sido manipulada o introducida por un tercero ajeno al mismo.

El sitio web del prestador puede utilizar cookies (pequeños archivos de información que el servidor envía al ordenador de quien accede a la página) para llevar a cabo determinadas funciones que son consideradas imprescindibles para el correcto funcionamiento y visualización del sitio. Las cookies utilizadas en el sitio web tienen, en todo caso, carácter temporal con la única finalidad de hacer más eficaz su transmisión ulterior y desaparecen al terminar la sesión del usuario. En ningún caso se utilizarán las cookies para recoger información de carácter personal.

Desde el sitio web del cliente es posible que se redirija a contenidos de terceros sitios web. Dado que el prestador no puede controlar siempre los contenidos introducidos por los terceros en sus sitios web, éste no asume ningún tipo de responsabilidad respecto a dichos contenidos. En todo caso, el prestador manifiesta que procederá a la retirada inmediata de cualquier contenido que pudiera contravenir la legislación nacional o internacional, la moral o el orden público, procediendo a la retirada inmediata de la redirección a dicho sitio web, poniendo en conocimiento de las autoridades competentes el contenido en cuestión.

El prestador no se hace responsable de la información y contenidos almacenados, a título enunciativo pero no limitativo, en foros, chats, generadores de blogs, comentarios, redes sociales o cualesquiera otro medio que permita a terceros publicar contenidos de forma independiente en la página web del prestador. No obstante y en cumplimiento de lo dispuesto en el art. 11 y 16 de la LSSI-CE, el prestador se pone a disposición de todos los usuarios, autoridades y fuerzas de seguridad, y colaborando de forma activa en la retirada o en su caso bloqueo de todos aquellos contenidos que pudieran afectar o contravenir la legislación nacional, o internacional, derechos de terceros o la moral y el orden público. En caso de que el usuario considere que existe en el sitio web algún contenido que pudiera ser susceptible de esta clasificación, se ruega lo notifique de forma inmediata al administrador del sitio web.

Este sitio web ha sido revisado y probado para que funcione correctamente. En principio, puede garantizarse el correcto funcionamiento los 365 días del año, 24 horas al día. No obstante, el prestador no descarta la posibilidad de que existan ciertos errores de programación, o que acontezcan causas de fuerza mayor, catástrofes naturales, huelgas, o circunstancias semejantes que hagan imposible el acceso a la página web.

#### Protección de datos personales

El prestador se encuentra profundamente comprometido con el cumplimiento de la normativa española de protección de datos de carácter personal, y garantiza el cumplimiento íntegro de las obligaciones dispuestas, así como la implementación de las medidas de seguridad dispuestas en el art. 9 de la Ley 15/1999, de Protección de Datos de Carácter Personal (LOPD) y en el Reglamento de Desarrollo de la LOPD.

El prestador pone a disposición de los usuarios la *Política de Privacidad de la entidad* informando a los usuarios respecto a los siguientes aspectos:

- Datos del Responsable del tratamiento.
- Datos tratados.
- Fichero en el que se almacenan.
- Finalidad del tratamiento.
- Obligatoriedad o no de facilitarlos, así como las consecuencias en caso de no facilitarlos.
- Sobre los derechos que asisten a todo usuario y el procedimiento para ejercitarlos.

#### Propiedad Intelectual e industrial

El sitio web, incluyendo a título enunciativo pero no limitativo su programación, edición, compilación y demás elementos necesarios para su funcionamiento, los diseños, logotipos, texto y/o gráficos son propiedad del prestador o en su caso dispone de licencia o autorización expresa por parte de los autores. Todos los contenidos del sitio web se encuentran debidamente protegidos por la normativa de propiedad intelectual e industrial, así como inscritos en los registros públicos correspondientes.

Independientemente de la finalidad para la que fueran destinados, la reproducción total o parcial, uso, explotación, distribución y comercialización, requiere en todo caso de la autorización escrita previa por parte del prestador. Cualquier uso no autorizado previamente por parte del prestador será considerado un incumplimiento grave de los derechos de propiedad intelectual o industrial del autor.

Los diseños, logotipos, texto y/o gráficos ajenos al prestador y que pudieran aparecer en el sitio web, pertenecen a sus respectivos propietarios, siendo ellos mismos responsables de cualquier posible controversia que pudiera suscitarse respecto a los mismos. En todo caso, el prestador cuenta con la autorización expresa y previa por parte de los mismos.

*(Opcional)* El prestador *autoriza/no autoriza* expresamente a que terceros puedan redirigir directamente a los contenidos concretos del sitio web, debiendo en todo caso redirigir al sitio web principal del prestador.

El prestador reconoce a favor de sus titulares los correspondientes derechos de propiedad industrial e intelectual, no implicando su sola mención o aparición en el sitio web la existencia de derechos o responsabilidad alguna del prestador sobre los mismos, como tampoco respaldo, patrocinio o recomendación por parte del mismo.

Para realizar cualquier tipo de observación respecto a posibles incumplimientos de los derechos de propiedad intelectual o industrial, así como sobre cualquiera de los contenidos del sitio web, puede hacerlo a través del siguiente correo electrónico.

#### [Ley Aplicable y Jurisdicción](#)

Para la resolución de todas las controversias o cuestiones relacionadas con el presente sitio web o de las actividades en él desarrolladas, será de aplicación la legislación española, a la que se someten expresamente las partes, siendo competentes para la resolución de todos los conflictos derivados o relacionados con su uso los Juzgados y Tribunales de *(Indicar lugar)*.

## 20.3 Código fuente

Todo el código fuente se encuentra en GitHub en repositorios privados, para solicitar acceso enviar un email a: [pablo.bcn.16@gmail.com](mailto:pablo.bcn.16@gmail.com).

## 20.4 Manual de usuario

### Aplicación web

**Barra de navegación:** La barra de navegación aparece siempre en la parte superior de la pantalla, en esta barra desde la página principal podemos ver el idioma, una página de contacto, la página sobre y el manual de usuario, de otro lado una vez hayamos iniciado sesión podemos ver nuestro nombre de usuario que nos permite ver el perfil o bien cerrar la sesión en lugar del idioma.

**Página principal:** En esta página el usuario puede ver por un lado el formulario de inicio de sesión (superior) y el formulario para registrarse (inferior).

**Página proyectos:** Es la primera página que aparece tras iniciar sesión por defecto, en ella se puede ver los proyectos que tenemos a la izquierda y las **notificaciones y avisos** a la derecha, de otro lado el botón “create new project” nos lleva al formulario para crear un nuevo proyecto.

**Página proyecto:** Esta es la página que se muestra cuando entramos en un proyecto, a la izquierda se pueden ver los datos del proyecto y al lado del nombre dos iconos que sirven para editar y borrar respectivamente y a la parte derecha se muestra el **diagrama de Gantt**, veamos a continuación que podemos hacer en esta página:

**Crear nueva asignación al proyecto:** Si hacemos click en “Project assignments” podremos acceder a las asignaciones y añadir o quitar cualquiera de ellas (siempre y cuando tengamos permisos).

**Crear nuevo hito:** Si hacemos click en “Project milestones” podremos acceder a los hitos y crear nuevos hitos o eliminarlos (siempre y cuando tengamos permisos).

**Crear nuevas tareas:** Si se dispone de permisos aparecerá un botón azul “create a task” este botón nos llevara al formulario de tareas, donde además de la tarea añadiremos a los usuarios que queramos que estén asignados a la tarea.

**Ver tareas:** Si hacemos click en “Project task” podremos las tareas más detalladas.

**Página tareas:** Aquí podemos ver la lista de todas las tareas del proyecto, desde esta pantalla se puede editar la tarea haciendo click en su nombre, y finalizarla con el botón “end task”.

**Añadir checklist, deliverables o previous task:** Para hacer cualquiera de estas funciones es necesario ir al formulario de edición de tareas.

### [Aplicación Android](#)

En la aplicación Android solo hay que iniciar sesión y automáticamente se recibirán las notificaciones. Al iniciar sesión pueden modificarse algunas opciones para recibir las notificaciones.