



Universidad de Oviedo



Escuela de  
Ingeniería  
Informática  
Universidad de Oviedo

Universidad de Oviedo

Máster en Ingeniería Web

---

Trabajo Fin de Máster

---

Codesnug

Herramienta social de almacenamiento de código

*Vº Bº del Director del Proyecto*

**Director:** Miguel García Rodríguez

**Subdirector:** Daniel Fernández Larvin

**Autor:** René Fernández Sánchez

# 1 Introducción

## 1.1 Agradecimientos

Gracias a la gente que me ha ayudado y apoyado, especialmente a mi novia, a mi director de proyecto, a mis amigos y mi familia. Gracias también a todos aquellos que me han ayudado a probar el funcionamiento de la aplicación.

## 1.2 Resumen

Es común entre los desarrolladores de Software recurrir a los mismos fragmentos de código una y otra vez. La reutilización es una técnica que permite mejorar la productividad aprovechando el trabajo previamente realizado.

Actualmente existen diversas herramientas que permiten almacenar y compartir código. Sin embargo, desde el punto de vista de los desarrolladores, acceder a ese código almacenado en la nube o en aplicaciones de terceros no resulta cómodo y en muchas ocasiones se pierde el control sobre el mismo debido a la utilización de herramientas cerradas, lo que termina generando una dependencia del desarrollador por las mismas.

Teniendo en cuenta estas premisas, se ha decidido crear el actual proyecto, de nombre en clave “CodeSnug”.

CodeSnug viene de las palabras en inglés “*Code*” (Código) y “*Snug*” (Cómodo), pretendiendo dejar claro desde un primer momento que el objetivo del proyecto es el de realizar una herramienta sencilla y cómoda de utilizar, puesto que la productividad de un desarrollador depende en gran medida de estos factores.

CodeSnug será una plataforma web que permitirá a sus usuarios almacenar código en distintos lenguajes de programación, clasificarlo, etiquetarlo y compartirlo con otros usuarios de la aplicación, pero sobre todo acceder a él de la manera más cómoda posible.

Para garantizar el control de los usuarios sobre su código, se permitirá la exportación del mismo en todo momento.

También se implementará un servicio web para acceder a las características del sitio web desde aplicaciones de terceros, de manera que sea un servicio extensible.

Para garantizar la seguridad en las comunicaciones se utilizará HTTPS en todo momento y también se utilizará el mayor número de medidas de seguridad posibles (validación de formularios en el servidor, campos CSFR, cifrado de sesiones, etc).

Además, se crearán cuentas en servicios de terceros y redes sociales para ampliar las funcionalidades originales del sitio web (Twitter, Blog en Wordpress, foro de sugerencias, etc).

Para la realización del proyecto se utilizará el lenguaje de programación Python y su framework Django, aunque también habrá elementos escritos en Javascript (Incluyendo JQuery y AJAX), HTML5 y CSS3. En el lado del servidor se utilizará un servidor Apache que ejecutará mediante *mod\_wsgi* la aplicación.

## 1.3 Palabras Clave

Código, repositorio, almacenamiento, software, programación, compartir, social

## 1.4 Abstract

Usually, software developers use the same fragments of code many times. Reusability is a technique that allow developers to improve their productivity by using code that they have already wrote for another projects.

Nowadays there are some tools that can help developers to store and share their code. However, from their point of view accessing that code, wich can be stored in the cloud or in 3rd party apps is not comfortable and many times developers lose the control over it due to the utilization of closed platforms, wich ends generating a dependency on that platforms by the developer.

Given these premises, I decided to build this project, codenamed "CodeSnug".

CodeSnug comes from the englishword "Code" (Code) and "Snug" (Comfortable), pretending to make clear from the outset that the objective of the project is to perform a simple and convenient tool to use because productivity is very important for a developer.

CodeSnug will be a web platform that will allow its users to store code in different programming languages, classify, label and share it with other application users, but especially access it in the most comfortable way. To ensure user control over his code, export thereof is permitted at any time. A web service will also be implemented to access the features of the website from third-party applications.

To guarantee the security on the communications, HTTPS will be used everytime, aswel as other security measures like server-side form validating, CSFR fields, session encryption, etc.

Furthermore, account on other services will be also created as complementary items like a Twitter account, Wordpress blog and a suggestions forum.

The technologies to be used will be Python and its framework Django, Javascript (jQuery and AJAX), HTML5 and CSS3. On the server-side there will be an Apache server that will run the application using mod\_wsgi.

## 1.5 Keywords

Code, repository, storage, programming, software, share, social.

## 1.6 Tabla de contenido

<b>1</b>	<b>INTRODUCCIÓN</b>	<b>2</b>
1.1	AGRADECIMIENTOS	2
1.2	RESUMEN	3
1.3	PALABRAS CLAVE	4
1.4	ABSTRACT	5
1.5	KEYWORDS	6
1.6	TABLA DE CONTENIDO	7
1.7	ÍNDICE DE ILUSTRACIONES	10
<b>2</b>	<b>MEMORIA DEL PROYECTO</b>	<b>14</b>
2.1	RESUMEN DE LA MOTIVACIÓN, OBJETIVOS Y ALCANCE DEL PROYECTO	14
2.2	RESUMEN DE TODOS LOS ASPECTOS	15
2.2.1	Introducción	15
2.2.2	Memoria del proyecto	15
2.2.3	Aspectos generales	15
2.2.4	Aspectos teóricos	15
2.2.5	Planificación del proyecto y resumen de presupuestos	15
2.2.6	Análisis	15
2.2.7	Diseño del sistema	15
2.2.8	Implementación del sistema	15
2.2.9	Manuales del sistema	16
2.2.10	Conclusiones y ampliaciones	16
2.2.11	Presupuesto	16
2.2.12	Referencias bibliográficas	16
2.2.13	Apéndices	16
2.2.14	Índice alfabético	16
<b>3</b>	<b>ASPECTOS GENERALES</b>	<b>17</b>
3.1	JUSTIFICACIÓN DEL PROYECTO	17
3.2	OBJETIVOS DEL PROYECTO	17
3.2.1	Creación, edición y almacenamiento de código	17
3.2.2	Organización y privacidad del código	17
3.2.3	Exportación del código	17
3.2.4	Gestión de grupos de usuarios	17
3.2.5	Búsqueda de código	17
3.2.6	Servicio web mediante una API REST	17
3.2.7	Seguridad	18
3.3	ESTUDIO DE LA SITUACIÓN ACTUAL	19
3.4	EVALUACIÓN DE LAS ALTERNATIVAS	24
<b>4</b>	<b>ASPECTOS TEÓRICOS</b>	<b>27</b>
4.1	CONCEPTOS RELACIONADOS CON LA APLICACIÓN	27
4.1.1	Snippet	27
4.1.2	Goal	27
4.1.3	Workspace	27
4.1.4	Tag	27
4.1.5	Usergroup	28
<b>5</b>	<b>PLANIFICACIÓN DEL PROYECTO Y RESUMEN DE PRESUPUESTOS</b>	<b>29</b>
5.1	PLANIFICACIÓN	29

5.1.1	Fase de análisis .....	29
5.1.2	Fase de diseño del sistema .....	31
5.1.3	Fase de desarrollo del sistema .....	33
5.1.4	Fase de desarrollo de elementos secundarios del sitio web .....	35
5.1.5	Creación de elementos ajenos al servicio principal .....	36
5.1.6	Despliegue del sistema (alpha 1).....	36
5.1.7	Despliegue del sistema (alpha 2).....	37
5.1.8	Despliegue del sistema (beta 1) .....	38
5.1.9	Despliegue del sistema (beta 2) .....	38
5.1.10	Configuración de un sistema de copias de seguridad .....	40
5.1.11	Documentación .....	40
5.1.12	Fase de presentación .....	41
5.1.13	Mantenimiento del sitio web (cíclico) .....	42
5.1.14	Fase de comunicación (cíclico).....	43
5.2	RESUMEN DEL PRESUPUESTO .....	44
<b>6</b>	<b>ANÁLISIS.....</b>	<b>45</b>
6.1	DEFINICIÓN DEL SISTEMA .....	45
6.1.1	Determinación del alcance del sistema .....	45
6.2	REQUISITOS DEL SISTEMA .....	46
6.2.1	Obtención de requisitos del sistema .....	46
6.2.2	Identificación de los actores del sistema .....	48
6.2.3	Identificación de los subsistemas .....	49
6.2.4	Especificación de los casos de uso .....	50
6.3	ANÁLISIS DE INTERFACES DE USUARIO.....	65
6.3.1	Diagrama de navegabilidad .....	65
6.3.2	Pantallas del sitio web .....	66
6.4	DIAGRAMA DE CLASES PRELIMINAR DEL ANÁLISIS .....	70
6.4.1	Descripción de las clases .....	70
<b>7</b>	<b>DISEÑO DEL SISTEMA .....</b>	<b>74</b>
7.1	ARQUITECTURA DEL SISTEMA.....	74
7.1.1	Estructura de la Aplicación web.....	75
7.2	ESTRUCTURA DE LA BASE DE DATOS .....	119
7.2.1	Backups de la base de datos.....	119
7.3	DISEÑO DE LA INTERFAZ DE USUARIO .....	129
7.3.1	Página de inicio.....	130
7.3.2	Página de registro.....	133
7.3.3	Página de acceso .....	134
7.3.4	Página de listado de goals.....	135
7.3.5	Página de detalles de goal.....	137
7.3.6	Página de edición de goal.....	138
7.3.7	Página de listado de etiquetas.....	140
7.3.8	Página de creación de workspaces .....	141
7.3.9	Página de listado de workspaces .....	142
7.3.10	Página de detalles de grupo de usuarios.....	144
7.3.11	Página de añadir usuario a grupo de usuarios .....	144
7.3.12	Página de listado de grupos de usuarios .....	146
7.3.13	Página de perfil de usuario .....	147
7.3.14	Página de texto estándar .....	149
7.4	FUNCIONALIDADES DE LA API.....	150
7.4.1	API Navegable .....	151
7.5	ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS .....	152
7.5.1	Pruebas unitarias .....	152
7.5.2	Pruebas de integración y del sistema .....	158
<b>8</b>	<b>IMPLEMENTACIÓN DEL SISTEMA .....</b>	<b>162</b>
8.1	LENGUAJES DE PROGRAMACIÓN Y TECNOLOGÍAS USADAS .....	162



8.1.1	Python.....	162
8.1.2	Javascript.....	162
8.1.3	HTML5.....	162
8.1.4	CSS3.....	162
8.1.5	Django.....	162
8.1.6	Twitter Bootstrap.....	163
8.1.7	pip.....	163
8.1.8	Git.....	163
8.1.9	PostgreSQL.....	163
8.2	<b>HERRAMIENTAS Y PROGRAMAS USADOS PARA EL DESARROLLO.....</b>	<b>164</b>
8.2.1	PyCharm.....	164
8.2.2	Visual Paradigm.....	164
8.2.3	GitHub.....	164
8.2.4	Microsoft Project.....	164
8.2.5	Balsamiq Mockups.....	165
8.2.6	PgCommander.....	165
8.2.7	PostgresApp.....	165
8.2.8	MAMP.....	165
8.3	<b>CREACIÓN DEL SISTEMA.....</b>	<b>166</b>
8.3.1	Librerías pip utilizadas.....	166
8.3.2	Elementos ajenos al sitio web.....	167
8.3.3	Política de privacidad y condiciones legales.....	167
8.3.4	Problemas encontrados.....	170
<b>9</b>	<b>MANUALES DEL SISTEMA.....</b>	<b>175</b>
9.1	MANUAL DE USUARIO.....	175
9.2	MANUAL DEL PROGRAMADOR.....	175
9.2.1	Entorno de desarrollo local.....	175
9.2.2	Creación y gestión de la base de datos y de sus migraciones.....	176
9.2.3	Instalación de las dependencias del proyecto.....	176
9.2.4	Generación de traducciones.....	176
9.2.5	Creación e instalación de un certificado de servidor para generar un SSL.....	177
9.2.6	Redireccionamiento de direcciones sin SSL y sin www.....	177
9.2.7	Organización de un proyecto Git.....	177
9.2.8	Configuración de la búsqueda dentro de la aplicación.....	178
9.2.9	Despliegue en producción.....	178
<b>10</b>	<b>PRESUPUESTO DETALLADO.....</b>	<b>180</b>
<b>11</b>	<b>CONCLUSIONES Y AMPLIACIONES.....</b>	<b>181</b>
11.1	CONCLUSIONES.....	181
11.2	AMPLIACIONES.....	182
11.2.1	Creación de una aplicación de escritorio.....	182
11.2.2	Crear una aplicación móvil.....	182
11.2.3	Extender las funcionalidades de la API.....	182
11.2.4	Autenticación Social.....	182
11.2.5	Modificar los colores de la página para hacerlos más accesibles.....	182
11.2.6	Permitir goals públicos.....	182
11.2.7	Permitir pantalla completa al visualizar los snippets.....	182
11.2.8	Mejorar la accesibilidad y el código HTML utilizado.....	182
<b>12</b>	<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>183</b>
12.1	PÁGINAS WEB.....	183

## 1.7 Índice de Ilustraciones

ILUSTRACIÓN 1. CAPTURA DE PANTALLA DE GITHUB GIST .....	19
ILUSTRACIÓN 2. CAPTURA DE PANTALLA DE GITBOX .....	20
ILUSTRACIÓN 3. CAPTURA DE PANTALLA DE MY CODE STOCK .....	21
ILUSTRACIÓN 4. CAPTURA DE PANTALLA DE SNIPPLR.....	22
ILUSTRACIÓN 5. CAPTURA DE BITBUCKET SNIPPET .....	23
ILUSTRACIÓN 6. PLANIFICACIÓN DEL PROCESO DE ANÁLISIS.....	29
ILUSTRACIÓN 7. PLANIFICACIÓN DEL PROCESO DE DISEÑO.....	31
ILUSTRACIÓN 8. PLANIFICACIÓN DEL DESARROLLO DEL SISTEMA.....	33
ILUSTRACIÓN 9. PLANIFICACIÓN DE LA FASE DE DESARROLLO DEL SISTEMA.....	35
ILUSTRACIÓN 10. PLANIFICACIÓN DE LA CREACIÓN DE ELEMENTOS AJENOS .....	36
ILUSTRACIÓN 11. PLANIFICACIÓN DE LA ALPHA 1 .....	36
ILUSTRACIÓN 12. PLANIFICACIÓN DE LA BETA 1 .....	38
ILUSTRACIÓN 13. PLANIFICACIÓN DE LA CONFIGURACIÓN DEL SISTEMA DE COPIAS DE SEGURIDAD.....	40
ILUSTRACIÓN 14. PLANIFICACIÓN DE LA DOCUMENTACIÓN.....	40
ILUSTRACIÓN 15. PLANIFICACIÓN DE LA PRESENTACIÓN.....	41
ILUSTRACIÓN 16. PLANIFICACIÓN DEL MANTENIMIENTO .....	42
ILUSTRACIÓN 17. PLANIFICACIÓN DE LA FASE DE COMUNICACIÓN .....	43
ILUSTRACIÓN 18. SUBSISTEMAS DE LA APLICACIÓN .....	49
ILUSTRACIÓN 20. DIAGRAMA DE CASO DE USO DE GESTIÓN DE GOALS .....	52
ILUSTRACIÓN 21. DIAGRAMA DE CASOS DE USO DE GESTIÓN DE ARCHIVOS.....	55
ILUSTRACIÓN 22. DIAGRAMA DE CASOS DE USO DE GESTIÓN DE ÉTIQUETAS .....	57
ILUSTRACIÓN 23. DIAGRAMA DE CASOS DE USO DE GESTIÓN DE EXPORTACIONES .....	59
ILUSTRACIÓN 24. DIAGRAMA DE CASOS DE USO DE GESTIÓN DE WORKSPACES .....	60
ILUSTRACIÓN 25. DIAGRAMA DE CASOS DE USO DE GESTIÓN DE GRUPOS DE USUARIOS .....	61
ILUSTRACIÓN 26. DIAGRAMA DE CASOS DE USO DE GESTIÓN DE USUARIOS .....	63
ILUSTRACIÓN 27. DIAGRAMA DE NAVEGABILIDAD.....	65
ILUSTRACIÓN 28. MUCKUP DE LA PANTALLA DE CREACIÓN DE GOAL .....	66
ILUSTRACIÓN 29. MOCKUP DE LA PANTALLA DE CREACIÓN DE WORKSPACE.....	67

ILUSTRACIÓN 30. MOCKUP DE LA PANTALLA DE CREACIÓN DE GRUPO DE USUARIOS .....	67
ILUSTRACIÓN 31. MOCKUP DE LA PANTALLA DE LISTADO DE GOALS .....	68
ILUSTRACIÓN 32. MOCKUP DE LA PANTALLA DE DETALLES DE GOAL.....	69
ILUSTRACIÓN 33. MOCKUP DE LA PANTALLA DE DETALLES DE GRUPO DE USUARIOS .....	69
ILUSTRACIÓN 34. DIAGRAMA DE CLASES PRELIMINAR .....	70
ILUSTRACIÓN 35. DIAGRAMA DE DESPLIEGUE DEL SISTEMA.....	74
ILUSTRACIÓN 36. DIAGRAMA DE MÓDULOS DE LA APLICACIÓN.....	75
ILUSTRACIÓN 37. MODELOS DE DATOS LA APLICACIÓN .....	76
ILUSTRACIÓN 38. MODELOS DEL MÓDULO DE GOALS.....	77
ILUSTRACIÓN 39. MODELOS DEL MÓDULO USERS .....	82
ILUSTRACIÓN 40. CONTROLADORES DEL MÓDULO SEARCH .....	86
ILUSTRACIÓN 41. FORMULARIO DEL MÓDULO SEARCH .....	88
ILUSTRACIÓN 42. SERIALIZADORES DE LA API .....	89
ILUSTRACIÓN 43. CONTROLADORES DE LA API.....	92
ILUSTRACIÓN 44. CONTROLADORES DE LAS VISTAS DE GOALS DEL SITIO WEB .....	93
ILUSTRACIÓN 45. CONTROLADORES DE LAS VISTAS DE WORKSPACES DEL SITIO WEB .....	99
ILUSTRACIÓN 46. CONTROLADORES DE LAS VISTAS DE TAGS DEL SITIO WEB .....	103
ILUSTRACIÓN 47. CONTROLADOR DE LA VISTA DE DETALLES DE SNIPPET .....	111
ILUSTRACIÓN 48. CONTROLADORES DE LAS VISTAS DE DESCARGA DE DATOS.....	112
ILUSTRACIÓN 49. MÓDULOS DE FORMULARIOS DEL SITIO WEB .....	115
ILUSTRACIÓN 50. MODELOS DE LOS FORMULARIOS DE USUARIOS.....	115
ILUSTRACIÓN 51. DIAGRAMA DE ACTIVIDAD DEL REGISTRO DE USUARIOS .....	116
ILUSTRACIÓN 52. MODELOS DE LOS FORMULARIOS DE GOALS .....	117
ILUSTRACIÓN 53. ESTRUCTURA DE LAS CLAVES PRIMARIAS DE LA BASE DE DATOS .....	119
ILUSTRACIÓN 54. ZONA 1 DE LA PÁGINA DE INICIO.....	130
ILUSTRACIÓN 55. MENÚ SUPERIOR DEL SITIO WEB.....	130
ILUSTRACIÓN 56. ZONA 2 DE LA PÁGINA DE INICIO.....	131
ILUSTRACIÓN 57. ZONA 3 DE LA PÁGINA DE INICIO.....	132
ILUSTRACIÓN 58. PIE DE PÁGINA DEL SITIO WEB.....	132

ILUSTRACIÓN 59. PÁGINA DE REGISTRO DEL SITIO WEB .....	133
ILUSTRACIÓN 60. PÁGINA DE ACCESO DEL SITIO WEB .....	134
ILUSTRACIÓN 61. EJEMPLO DE ERROR EN UN CAMPO DE FORMULARIO .....	134
ILUSTRACIÓN 62. PÁGINA DE LISTADO DE GOALS .....	135
ILUSTRACIÓN 63. INFORMACIÓN DE LA BARRA LATERAL DEL SITIO WEB .....	135
ILUSTRACIÓN 64. ZONA IZQUIERDA DEL MENÚ SUPERIOR PARA USUARIOS REGISTRADOS .....	136
ILUSTRACIÓN 65. ZONA DERECHA DEL MENÚ SUPERIOR PARA USUARIOS REGISTRADOS .....	136
ILUSTRACIÓN 66. PÁGINA DE DETALLES DE GOAL .....	137
ILUSTRACIÓN 67. PÁGINA DE EDICIÓN DE GOALS .....	138
ILUSTRACIÓN 68. SECCIÓN DE CABECERA DE SNIPPET .....	138
ILUSTRACIÓN 69. BARRA LATERAL DERECHA DE LA PANTALLA DE EDICIÓN DE GOALS.....	139
ILUSTRACIÓN 70. ZONA INFERIOR DE LA BARRA LATERAL DERECHA DE LA PANTALLA DE EDICIÓN DE GOALS .....	139
ILUSTRACIÓN 71. PÁGINA DE LISTADO DE ETIQUETAS .....	140
ILUSTRACIÓN 72. PÁGINA DE CREACIÓN DE WORKSPACES.....	141
ILUSTRACIÓN 73. PÁGINA DE LISTADO DE WORKSPACES .....	142
ILUSTRACIÓN 74. SECCIÓN DE SELECCIÓN DE WORKSPACES .....	142
ILUSTRACIÓN 75. MENSAJE DE CONFIRMACIÓN DE BORRADO.....	143
ILUSTRACIÓN 76. PÁGINA DE DETALLES DE GRUPO DE USUARIOS .....	144
ILUSTRACIÓN 77. PÁGINA PARA AÑADIR USUARIOS A UN GRUPO DE USUARIOS .....	144
ILUSTRACIÓN 78. PÁGINA DE LISTADO DE GRUPOS DE USUARIOS.....	146
ILUSTRACIÓN 79. ZONA DE DATOS DE USUARIO DE LA PÁGINA DE PERFIL .....	147
ILUSTRACIÓN 80. EJEMPLO DE EDICIÓN DE CAMPO.....	147
ILUSTRACIÓN 81. ZONA DE DESCARGA DE DATOS DE USUARIO DE LA PÁGINA DE PERFIL.....	148
ILUSTRACIÓN 82. ZONA DE RESUMEN DE LA PÁGINA DE PERFIL.....	148
ILUSTRACIÓN 83. EJEMPLO DE PÁGINA DE TEXTO .....	149
ILUSTRACIÓN 84. PÁGINA DE LA API NAVEGABLE .....	151
ILUSTRACIÓN 85. MENSAJE DE SOLICITUD DE PARTICIPACIÓN EN LA ENCUESTA .....	158
ILUSTRACIÓN 86. CAPTURA DEL MENSAJE DEL CERTIFICADO DEL SITIO WEB .....	170
ILUSTRACIÓN 87. RESULTADOS DEL TEST DE ACCESIBILIDAD HERA.....	171

ILUSTRACIÓN 88. VISTA DE LA BARRA LATERAL IZQUIERDA EXPANDIDA.....	172
ILUSTRACIÓN 89. VISTA CON LA BARRA LATERAL IZQUIERDA CERRADA .....	172
ILUSTRACIÓN 90. VISTA DE LA BARRA LATERAL DERECHA EXPANDIDA.....	173
ILUSTRACIÓN 91. VISTA DE LA BARRA LATERAL CERRADA .....	173
ILUSTRACIÓN 92. VISTA DEL MENSAJE DE COOKIES DEL SITIO WEB.....	174

## 2 Memoria del Proyecto

### 2.1 Resumen de la motivación, objetivos y alcance del proyecto

Durante los años que lleva el alumno desarrollando software, ha utilizado diversas herramientas para almacenar y reutilizar el código que ha ido empleando en sus distintos proyectos. Sin embargo, las aplicaciones y servicios utilizados a tal fin no han estado especializadas en el almacenamiento de código, por lo que se han mostrado limitadas en cuanto a su utilización para tal fin.

En la actualidad existen proyectos como Github Gists que permiten almacenar código, aunque centrándose en los aspectos sociales y con considerables limitaciones en cuanto al acceso posterior al código y a su clasificación. Aunque este tipo de proyectos podrían considerarse la base del proyecto a realizar, están profundamente limitados y no se ajustan a lo que un desarrollador realmente necesita de cara a la productividad.

También existen numerosas aplicaciones de escritorio que permiten almacenar código de manera local, pero teniendo en cuenta que es cada vez más frecuente trabajar y escribir código en distintos dispositivos, esto resulta también una gran limitación de cara a poder acceder en cualquier momento al código que hemos escrito previamente.

Con el proyecto a desarrollar, se pretenden aunar las ventajas de estos dos sistemas. Por un lado se utilizará la nube para almacenar el código y para acceder al mismo mediante una interfaz web. Se permitirá saber en todo momento a los usuarios con quién están compartiendo dicho código, clasificarlo y etiquetarlo.

Por otro lado se permitirá que terceras personas puedan ampliar sus funcionalidades, creando aplicaciones móviles, de escritorio o incluso otros sitios web que formen parte del ecosistema CodeSnug y fomenten su crecimiento y comunidad.

## 2.2 Resumen de todos los aspectos

A continuación se enumerarán y explicarán brevemente todos los apartados del presente documento para dotar al lector de unas nociones básicas sobre su estructura y contenido.

### 2.2.1 Introducción

Este apartado estará formado por las secciones estándar que sirven para explicar brevemente el proyecto y para clasificarlo.

### 2.2.2 Memoria del proyecto

En la memoria se resumirá el contenido del proyecto para todas aquellas personas que no dispongan de unos grandes conocimientos de informática o de la materia que en él se trata.

### 2.2.3 Aspectos generales

Una descripción de carácter no técnico de todas las funcionalidades del proyecto, incluyendo los motivos por los que se ha decidido su creación, comparándolo con otras opciones y evaluando las alternativas que hayan surgido durante la fase inicial.

### 2.2.4 Aspectos teóricos

En este apartado se explican y se detallan todas las herramientas, tecnologías y conceptos que se nombrarán durante los siguientes apartados y que son necesarios para comprender la ejecución del proyecto.

### 2.2.5 Planificación del proyecto y resumen de presupuestos

En esta sección se especifica la planificación que se ha llevado a cabo para la ejecución del proyecto, explicada mediante un diagrama de Gantt en el que se establece el tiempo de las tareas e hitos. Además, en este apartado también se encuentra un resumen de los presupuestos necesarios para llevar a cabo el proyecto.

### 2.2.6 Análisis

En este apartado se establecen los límites en el desarrollo del proyecto, explicando las características que se van a añadir al mismo, enumerando sus requisitos e identificando los casos de uso y actores del mismo, así como realizando análisis preliminares de su arquitectura. Se utilizan tablas y diagramas para facilitar la comprensión.

### 2.2.7 Diseño del sistema

Aquí se explica la arquitectura final que tendrá el sistema y todos sus componentes por medio de distintos diagramas que explicarán la interacción entre ellos y su funcionamiento. También se detalla el funcionamiento de la interfaz de usuario y se explican las pruebas que se han realizado en el sistema.

### 2.2.8 Implementación del sistema

En este apartado se enumeran las tecnologías, programas y lenguajes de programación describiéndolos brevemente y explicando su utilización dentro del proyecto. También se comentan brevemente algunos de los problemas que han encontrado para llevar a cabo el proyecto y las soluciones encontradas para ellos.

## 2.2.9 Manuales del sistema

En este apartado se encuentran los manuales que explican cómo un usuario puede utilizar la aplicación explicando las distintas partes de la misma y también la utilización y metodologías utilizadas para desarrollar ciertas partes del sistema.

### 2.2.10 Conclusiones y ampliaciones

En este apartado se incluyen las conclusiones finales a las que se ha llegado tras terminar el desarrollo del proyecto y algunas de las posibles ampliaciones que se le podrían realizar para convertirlo en una aplicación más completa.

### 2.2.11 Presupuesto

Aquí se desglosa el presupuesto mencionado en el apartado 5.2. Resumen del Presupuesto.

### 2.2.12 Referencias bibliográficas

Listado de libros y recursos web que se han utilizado para la elaboración del proyecto.

### 2.2.13 Apéndices

En este capítulo se enumeran los contenidos del disco que se ha entregado.

### 2.2.14 Índice alfabético

Un índice con los términos más relevantes del proyecto y las páginas en las que aparecen.



## 3 Aspectos generales

### 3.1 Justificación del proyecto

El presente proyecto se trata de una aplicación web creada en Python 2.7 utilizando el framework web Django y cuyo principal objetivo es el de almacenar código reutilizable por los desarrolladores de software. El código se podrá clasificar, etiquetar y compartir con otros usuarios de la aplicación. Además dispondrá de un sistema de control de versiones que permitirá ver la evolución del mismo a lo largo del tiempo.

El sistema incluirá la posibilidad de establecer y organizar grupos de usuarios, a los que se les podrán establecer permisos de acceso y edición a un código determinado, de tal forma que permitan contribuir en el desarrollo del mismo.

También se creará una API REST para acceder a los servicios ofrecidos por el sistema y permitir a aplicaciones de terceros interactuar con ellos.

El sitio web se alojará en un servidor web y será accesible a través del dominio [codesnug.com](http://codesnug.com). Toda la información se almacenará por tanto online y se requerirá de una conexión a Internet para poder acceder al sitio.

### 3.2 Objetivos del proyecto

#### 3.2.1 Creación, edición y almacenamiento de código

El objetivo principal del proyecto es el de almacenar código, por lo que será necesario permitir la creación, eliminación y edición del mismo.

#### 3.2.2 Organización y privacidad del código

El código se podrá organizar de varias formas, de cara al propio usuario y también de cara a compartirlo con otros usuarios.

#### 3.2.3 Exportación del código

Deberá existir la posibilidad de exportar el código que un usuario ha almacenado en la aplicación para que pueda gestionarlo a su manera desde fuera del sistema.

#### 3.2.4 Gestión de grupos de usuarios

Para compartir el código con otros usuarios, se podrán crear grupos de usuarios, editarlos y eliminarlos. A estos grupos de usuarios se les podrán asignar usuarios y establecer determinados permisos de edición o visualización del código sobre el código compartido.

#### 3.2.5 Búsqueda de código

Los usuarios podrán acceder a su código organizado previamente por etiquetas y también realizar búsquedas de código entre todo aquel al que tengan acceso, no solamente entre el creado por ellos.

#### 3.2.6 Servicio web mediante una API REST

Para que servicios de terceros (como por ejemplo aplicaciones móviles o de escritorio) puedan interactuar con el sitio web y con la información almacenada en el mismo, se dispondrá de una API REST.

### 3.2.7 Seguridad

Como medida transversal a todas las funcionalidades de la aplicación, se tendrán en cuenta también todos los aspectos posibles para que la aplicación sea segura tanto en lo referente a ataques, vulnerabilidades o comunicaciones.

## 3.3 Estudio de la situación actual

A continuación se detallan una serie de aplicaciones que disponen de funciones similares a las del presente proyecto, incluyendo sus puntos fuertes y débiles.

### 3.3.1.1 Github Gist

The screenshot shows a Github Gist page for a user named 'renefs87'. The gist is titled 'backup\_cron.sh' and was created on May 26, 2014. The main content is a shell script for database backup:

```

1 DBNAME="<DB_NAME>"
2 DBPASS="<PASSWORD>"
3 DBUSER="<USER>"
4 EMAIL="<EMAIL>"
5
6 mysqldump --opt -u $DBUSER -p$DBPASS $DBNAME > backup.sql
7 gzip backup.sql
8 DATE=`date +%Y%m%d` ; mv backup.sql.gz $DBNAME-backupFULL-$DATE.sql.gz
9 uuencode $DBNAME-backupFULL-$DATE.sql.gz $DBNAME-backupFULL-$DATE.sql.gz | mail -s "<SUBJECT>" $EMAIL
10 rm $DBNAME-backupFULL-$DATE.sql.gz
  
```

Below the script is a crontab entry:

```

1 0 4 * * * ~/<PATH_TO>/backup_cron.sh
  
```

The right sidebar contains options for 'Code', 'Revisions', 'Embed URL', 'HTTPS clone URL', and 'Download Gist'.

Ilustración 1. Captura de pantalla de Github Gist

Github Gist <sup>1</sup> es una herramienta creada por Github que permite compartir snippets con otros usuarios (gists). Todos los Gists son repositorios Git que pueden ser versionados y reutilizables desde Git.

#### 3.3.1.1.1 Ventajas

- Cada gist funciona como un repositorio Git.
- Interfaz sencilla de utilizar.
- Dispone de API.
- Permite comentarios en los gists.
- Permite embeber los snippets en sitios web.

#### 3.3.1.1.2 Inconvenientes

- Más centrada en compartir el código que en almacenarlo.
- Complicada de utilizar para organizar el código.
- No dispone de una exportación directa de todo el código almacenado.

### 3.3.1.2 Conclusiones

Gist es una herramienta potente para compartir fragmentos de código, que es su principal objetivo. Sin embargo, tiene carencias importantes en cuanto a la organización y la búsqueda de los gists por parte de los usuarios.

<sup>1</sup> URL de Github Gist: <https://gist.github.com/>

### 3.3.1.3 Gistbox

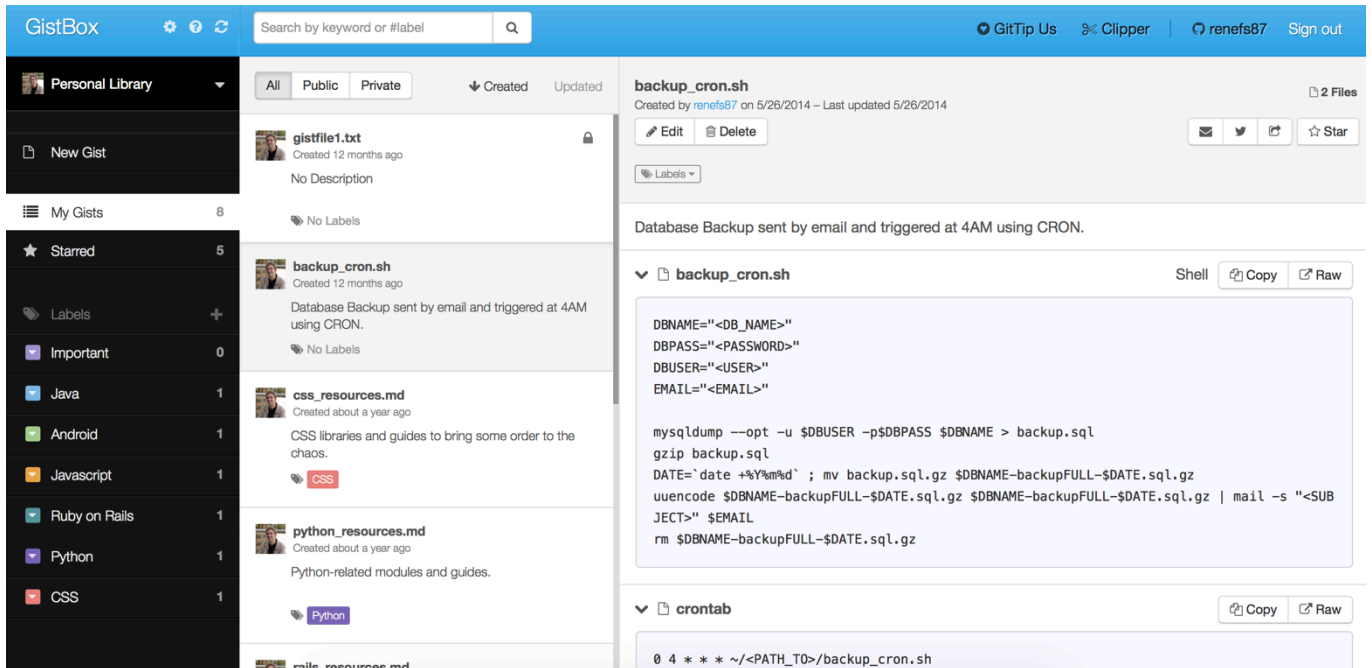


Ilustración 2. Captura de pantalla de Gistbox

Gistbox<sup>2</sup> es una herramienta creada sobre Github Gist que se define a si misma como “una manera bonita de organizar snippets de código”.

#### 3.3.1.3.1 Ventajas

- Diseño muy cuidado.
- Dispone de una extensión para Google Chrome.
- Permite clasificar el código utilizando etiquetas.
- Incluye gran parte de las funcionalidades de Github Gist.

#### 3.3.1.3.2 Inconvenientes

- Depende de Github Gist para la gestión y el almacenamiento de código.
- No permite la exportación directa de todos los datos almacenados.
- No dispone de su propia API al estar creada sobre Github Gist.
- No ofrece SSL.

#### 3.3.1.4 Conclusiones

Gistbox puede considerarse una capa extra de funcionalidades sobre Github Gist, pero estar construido sobre esta herramienta genera dependencias y limitaciones en cuanto a funcionalidades.

<sup>2</sup> URL de Gistbox: <http://www.gistboxapp.com>

### 3.3.1.5 My Code Stock

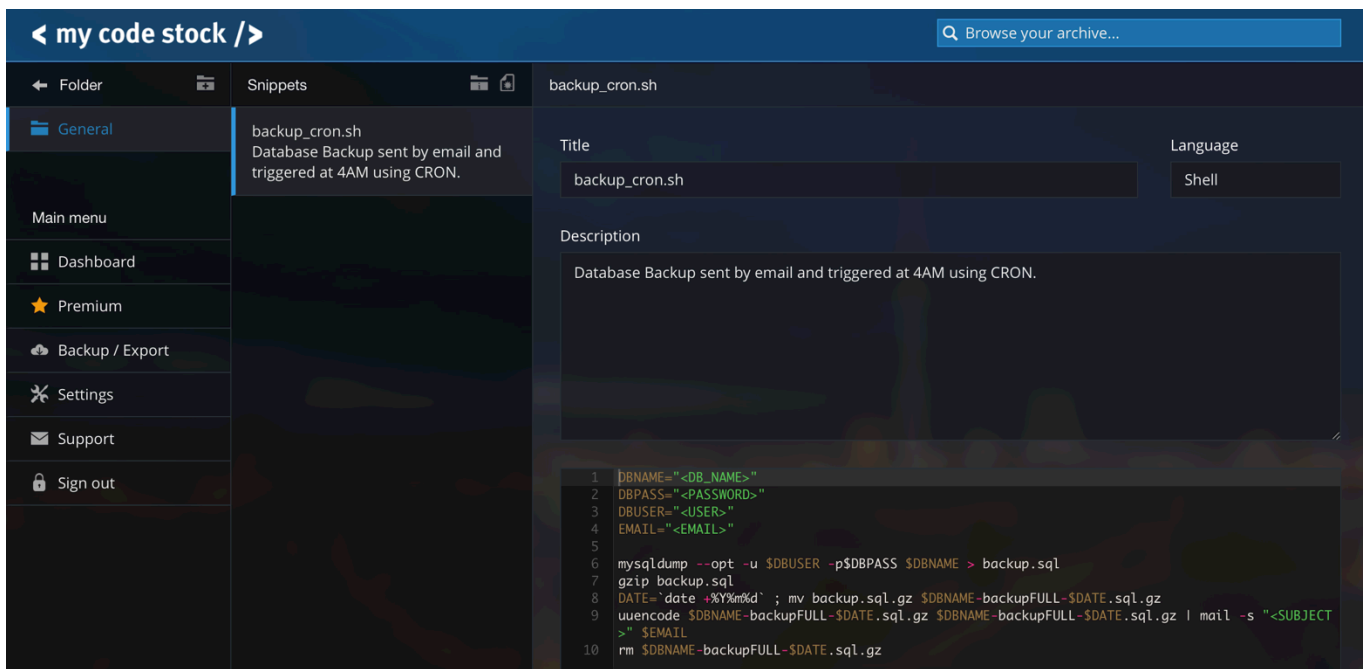


Ilustración 3. Captura de pantalla de My Code Stock

Mycodestock<sup>3</sup> es un sitio web que permite almacenar fragmentos de código y compartirlo con otros usuarios. A diferencia de las demás aplicaciones, esta dispone de una suscripción Premium para beneficiarse de ciertas características.

#### 3.3.1.5.1 Ventajas

- Dispone de gestión de versiones de los snippets.
- Permite compartir carpetas con otros usuarios.

#### 3.3.1.5.2 Inconvenientes

- Dispone de funcionalidades “Premium”.
- La exportación de los datos solamente está permitida a suscriptores.
- La interfaz resulta ciertamente confusa.
- La búsqueda avanzada solamente está disponible a suscriptores.
- No dispone de API.

#### 3.3.1.6 Conclusiones

Mycodestock es una aplicación con un diseño claramente cuidado que hace gala de las transparencias y de los colores llamativos. Sin embargo, resulta aparatosa cuando se utilizan pantallas pequeñas y la información en la pantalla queda demasiado apelotonado.

<sup>3</sup> URL de Mycodestock: <https://mycodestock.com>

### 3.3.1.7 Snipplr

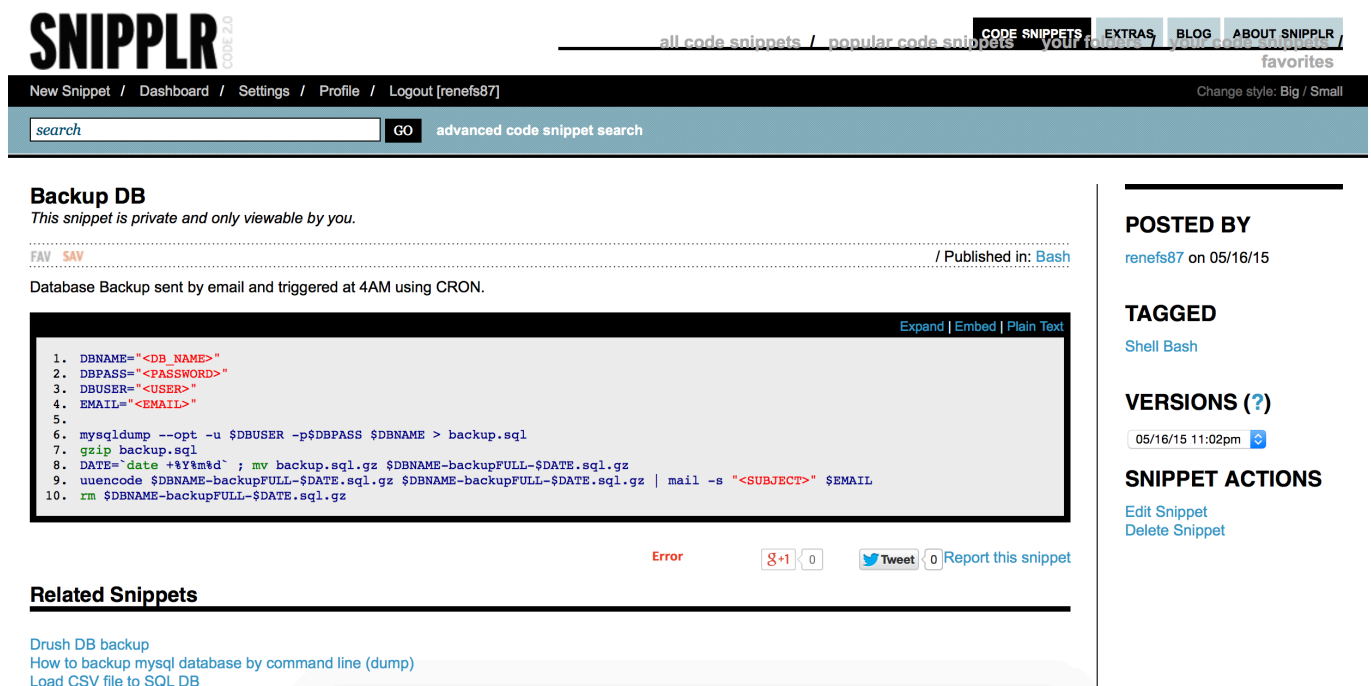


Ilustración 4. Captura de pantalla de Snipplr

Snipplr<sup>4</sup> es posiblemente uno de los gestores de código más antiguos disponibles actualmente. Sus inicios datan de 2006 y está realizado en PHP.

#### 3.3.1.7.1 Ventajas

- Gran cantidad de fragmentos de código públicos.
- Compatible con el editor de código TextMate.

#### 3.3.1.7.2 Inconvenientes

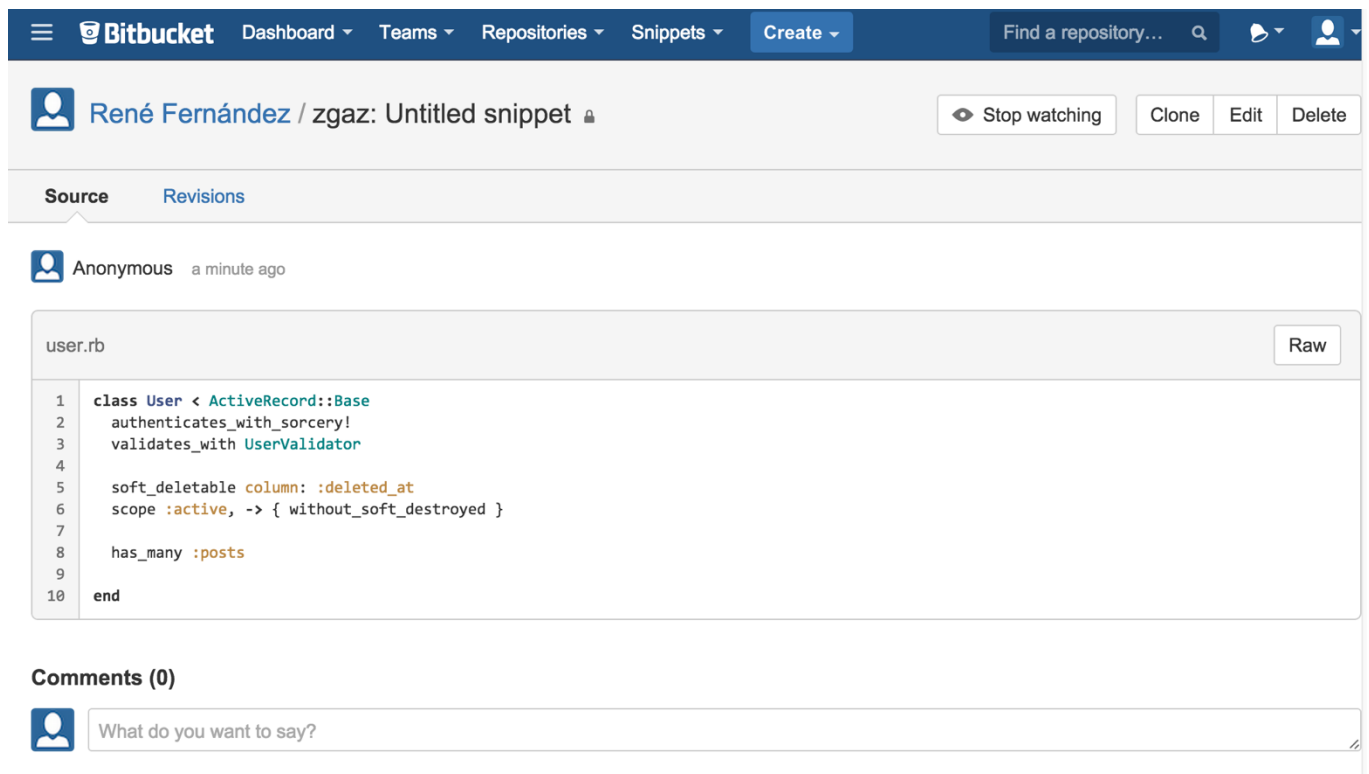
- Diseño confuso y anticuado.
- Problemas graves de visualización en Safari.
- El código compartido solamente puede ser privado o totalmente público.
- No dispone de API.
- No ofrece SSL.

#### 3.3.1.7.3 Conclusiones

Snipplr es una aplicación que posiblemente cuando se creó fue útil, pero los años le han sentado mal puesto que prácticamente no ha mejorado en cuanto a funcionalidades ni en diseño.

<sup>4</sup> URL de Snipplr: <http://snipplr.com>

### 3.3.1.8 Bitbucket Snippets



The screenshot shows the Bitbucket Snippets interface. At the top, there's a navigation bar with 'Bitbucket', 'Dashboard', 'Teams', 'Repositories', 'Snippets', and 'Create'. Below that, the user 'René Fernández' is shown with the snippet name 'zgaz: Untitled snippet'. There are buttons for 'Stop watching', 'Clone', 'Edit', and 'Delete'. The snippet is titled 'Source' and 'Revisions'. The code is for a file named 'user.rb' and is shown in a 'Raw' view. The code is as follows:

```

1 class User < ActiveRecord::Base
2   authenticates_with_sorcery!
3   validates_with UserValidator
4
5   soft_deletable column: :deleted_at
6   scope :active, -> { without_soft_destroyed }
7
8   has_many :posts
9
10 end

```

Below the code, there are 'Comments (0)' and a text input field with the placeholder 'What do you want to say?'.

Ilustración 5. Captura de Bitbucket Snippet

BitBucket Snippets es la versión equivalente a Gist de Bitbucket, mostrando muchas similitudes con la plataforma de Github.

#### 3.3.1.8.1 Ventajas

- Los snippets se encuentran bajo un control de versiones.
- Permite comentarios en los Snippets.
- Al cambiar la extensión al nombre del snippet, su lenguaje cambia automáticamente.
- Permiten clonarse los snippets.
- Modo a pantalla completa (RAW) de un snippet.

#### 3.3.1.8.2 Inconvenientes

- No permite una organización y clasificación completa.
- Más centrada en compartir el código que en almacenarlo.
- Complicada de utilizar para organizar el código.
- No dispone de una exportación directa de todo el código almacenado.

#### 3.3.1.8.3 Conclusiones

Al igual que Gists, BitBucket Snippets ofrece una herramienta potente pero con unas funcionalidades limitadas en cuanto a la organización de los mismos.

## 3.4 Evaluación de las alternativas

### 3.4.1.1 *Utilización de Java + Struts como lenguaje y framework de programación*

#### 3.4.1.1.1 *Descripción*

La elección del lenguaje de programación utilizado y de un framework es uno de los primeros puntos a tener en cuenta para creación del proyecto. En este caso, se ha planteado la utilización de Java y el framework Struts2.

#### 3.4.1.1.2 *Ventajas*

- Experiencia en Java desde hace años.
- Lenguaje de programación potente.
- Lenguaje más utilizado en el máster de ingeniería web.

#### 3.4.1.1.3 *Inconvenientes*

- Struts2 es un framework pesado en comparación con otras alternativas.
- Struts2 requiere de un servidor relativamente potente con respecto a otros frameworks.
- Está más dedicado al sector empresarial que al desarrollo de aplicaciones web.
- No dispone de un ORM propio y supondría utilizar uno externo como Hibernate.

### 3.4.1.2 *Utilización de Python + Django como lenguaje y framework de programación*

#### 3.4.1.2.1 *Descripción*

Otro planteamiento acerca del lenguaje de programación y el framework a utilizar es Python con Django.

#### 3.4.1.2.2 *Ventajas*

- Rapidez de desarrollo.
- Menos exigente con los recursos del servidor.
- Sistema de ORM y de modelos sencillo de utilizar.
- Flexible.
- Gran comunidad que proporciona multitud de módulos.
- Tecnologías en auge en diversos sectores.

#### 3.4.1.2.3 *Inconvenientes*

- Poca experiencia a la hora de comenzar el proyecto, lo que supondrá más tiempo de aprendizaje.

### 3.4.1.3 *Utilización de MongoDB como sistema de base de datos*

#### 3.4.1.3.1 *Descripción*

Una alternativa a los sistemas de bases de datos relacionales tradicionales son las bases de datos NoSQL, que basan su almacenamiento en ficheros de texto utilizando JSON como formato.

#### 3.4.1.3.2 *Ventajas*

- Escalabilidad sencilla (horizontal).
- Pueden gestionar gran cantidad de datos.



- Las estructuras de las entidades son flexibles (si por ejemplo se le añadiesen nuevos atributos a una clase, simplemente bastaría con añadir un atributo más al objeto JSON).

#### 3.4.1.3.3 *Inconvenientes*

- A largo plazo no es un sistema demasiado maduro en comparación con las bases de datos relacionales.
- Poca experiencia en este tipo de bases de datos.
- Menor orden en cuanto a una base de datos relacional.

### 3.4.1.4 *Utilización de PostgreSQL como sistema de base de datos sobre MySQL*

#### 3.4.1.4.1 *Descripción*

Tras la elección de un sistema de bases de datos relacional, se debe determinar qué sistema de base de datos utilizar. MySQL o PostgreSQL son las dos alternativas planteadas

#### 3.4.1.4.2 *Ventajas MySQL*

- Rápido
- Diseño más simple de la base de datos.
- Experiencia utilizando este tipo de bases de datos.
- Utilización de herramientas conocidas.

#### 3.4.1.4.3 *Inconvenientes MySQL*

- Peor soporte para concurrencia

#### 3.4.1.4.4 *Ventajas PostgreSQL*

- Mejor rendimiento en bases de datos complejas.
- Soporte para reglas de negocio.
- Soporte para transacciones.
- Soporte para procedimientos.
- Soporte para tipo de datos complejos

#### 3.4.1.4.5 *Inconvenientes PostgreSQL*

- Falta de experiencia en la utilización de este tipo de bases de datos que supondrán más tiempo de aprendizaje.

### 3.4.1.5 *Utilización de CodeMirror en vez de ACE Editor*

#### 3.4.1.5.1 *Descripción*

CodeMirror<sup>5</sup> y ACE Editor<sup>6</sup> son dos editores de código escritos en Javascript que permiten ser utilizados en proyectos que requieran resaltado de código y soporte para varios idiomas. La utilización de ambos es bastante pareja aunque ACE Editor es más antiguo que CodeMirror (su primera versión data de 2005). En la practica, ambos editores funcionan y se configuran de manera muy similar.

---

<sup>5</sup> URL de CodeMirror: <https://codemirror.net/>

<sup>6</sup> URL de Ace Editor: <http://ace.c9.io/>

#### 3.4.1.5.2 *Ventajas*

- Implementación sencilla.
- Su desarrollo continúa.

#### 3.4.1.5.3 *Inconvenientes*

- ACE Editor es utilizado por muchas empresas conocidas (Entre ellas Github o Mozilla).
- Problemas de rendimiento en archivos grandes.
- Cuando se comenzó el proyecto no disponía de módulo para Django.

## 4 Aspectos teóricos

### 4.1 Conceptos relacionados con la aplicación

#### 4.1.1 Snippet

##### 4.1.1.1 Descripción

Se llama Snippet a un fragmento de código fuente que cumple una función determinada.

##### 4.1.1.2 Utilización en el proyecto

El conjunto de datos más pequeño almacenado en la aplicación serán los Snippets, que definirán un fragmento de código escrito en un lenguaje de programación determinado.

#### 4.1.2 Goal

##### 4.1.2.1 Descripción

Se llama Goal a un conjunto de Snippets relacionados entre sí.

##### 4.1.2.2 Utilización en el proyecto

Los Goals serán el conjunto de datos principal de la aplicación. Podrán ser organizados y compartidos y se podrán eliminar, editar y añadir Snippets a ellos.

#### 4.1.3 Workspace

##### 4.1.3.1 Descripción

Un Workspace es un sistema para organizar Goals con el objetivo de compartirlos. Un mismo Goal puede estar en varios Workspaces.

##### 4.1.3.2 Utilización en el proyecto

Para clasificar los Goals y poder compartirlos con otros usuarios, los Goals pueden añadirse a Workspaces.

#### 4.1.4 Tag

##### 4.1.4.1 Descripción

Etiqueta para que un usuario pueda clasificar de manera interna sus propios Goals.

##### 4.1.4.2 Utilización en el proyecto

El funcionamiento de una etiqueta es similar al de un Workspace, salvo que la etiqueta está destinada a la organización individual de los propios usuarios, no a compartir los Goals con otros.

## 4.1.5 Usergroup

### 4.1.5.1 *Descripción*

Un Usergroup es un grupo de usuarios con capacidad para ver o editar los Goals de un Workspace.

### 4.1.5.2 *Utilización en el proyecto*

Para determinar qué usuarios pueden ver o editar los Goals de un Workspace, estos se englobarán en grupos de usuarios (Usergroups).

## 5 Planificación del proyecto y resumen de presupuestos

### 5.1 Planificación

El proyecto ha tenido cerca de un año de duración, comenando en Junio de 2014. Se ha estructurado en varias fases, siendo la fase de desarrollo la más extensa de todas. Además, de las fases correspondientes al inicio de todo proyecto (análisis, diseño e implementación) también existen otro tipo de tareas recurrentes que se han realizado cada cierto tiempo, como son la corrección de errores, publicación en redes sociales (*community managment*) o puestas en producción de versiones.

#### 5.1.1 Fase de análisis



Ilustración 6. Planificación del proceso de análisis

La fase de análisis ha sido la fase inicial del proyecto y ha servido para determinar todos los requisitos, funcionalidades y alcance de la aplicación, así como para establecer un presupuesto inicial de costes y de plazos.

Desde el primer momento se pretendía realizar la presentación del proyecto en Junio de 2015, por lo que se estableció esa fecha como *deadline* para el mismo.

Nº	Tarea	Nivel	Avance (%)	Descripción
3	Determinación de los requisitos	3	100	Buscar y determinar qué funciones realizará el sistema desarrollar y qué alcance tendrá.
4	Análisis de los casos de uso y escenarios	3	100	Determinar cuáles serán los actores y qué funciones podrán realizar con el sistema.
5	Reunión con el director del proyecto	3	100	Reunión para determinar si los requisitos son suficientes
6	Inicio del Proyecto	Hito	100	Comienzo del proyecto.

7	Planificación inicial	3	100	Planificación inicial para determinar los pasos a seguir para la consecución satisfactoria del proyecto.
8	Diagrama de clases preliminar	3	100	Análisis de las clases básicas necesarias para que el sistema funcione, independientemente de las tecnologías que se vayan a utilizar.
9	Especificación del plan de pruebas	3	100	Análisis de las pruebas mínimas que se deberían realizar para que el sistema funcione sin problemas y no ocurran errores ni fallos de seguridad.
10	Estimación de plazos del proyecto	3	100	Estimación en base al tiempo disponible para la realización del proyecto de las tareas necesarias para finalizar el proyecto.
11	Estimación de costes del proyecto	3	100	Cálculo de los costes estimados para llevar a cabo el proyecto (servidor, certificados, IDE, etc).
12	Fin de la fase de análisis	Hito	100	Finalización del análisis.

### 5.1.2 Fase de diseño del sistema

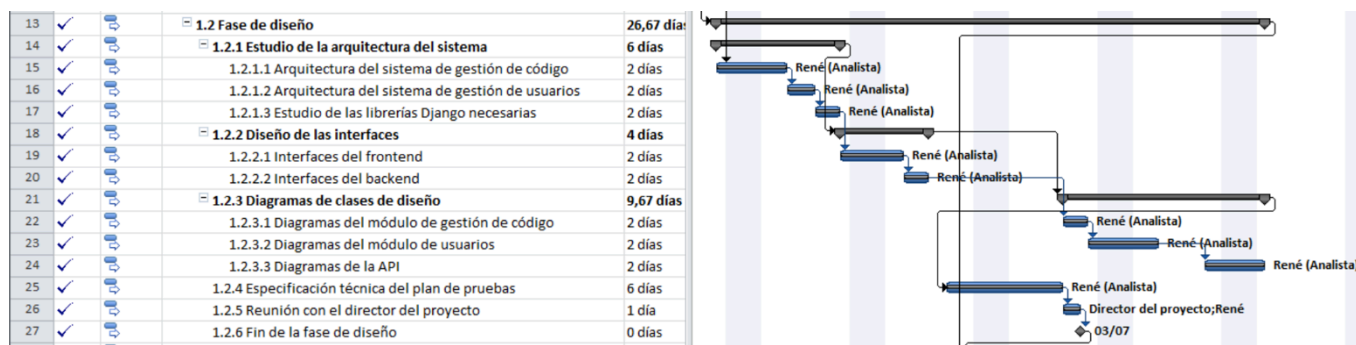


Ilustración 7. Planificación del proceso de diseño

Durante la fase de diseño del sistema se determinó la arquitectura, clases y módulos esperados y también se realizaron los primeros bocetos de los elementos más determinantes de la aplicación.

Nº	Tarea	Nivel	Avance (%)	Descripción
14	Estudio de la arquitectura del sistema	3	100	Estudio de la estructura del sistema a alto nivel.
15	Arquitectura del sistema de gestión de código	4	100	Arquitectura que deberá tener todo el módulo de gestión de código.
16	Arquitectura del sistema de gestión de usuarios	4	100	Arquitectura que deberá tener el sistema de gestión de usuarios de la aplicación
17	Estudio de las librerías Django necesarias	4	100	Creación de un plan de pruebas definitivo basándose en los componentes finales del sistema.
18	Diseño de las interfaces	3	100	Diseño teniendo en cuenta la usabilidad de las interfaces gráficas del sistema.
19	Interfaces del frontend	4	100	Diseño de las interfaces de cara al usuario.
20	Interfaces del backend	4	100	Diseño de las interfaces de cara a los administradores.

21	Diagramas de clases de diseño	3	100	Realización de los diagramas de clases definitivos teniendo en cuenta las tecnologías a utilizar.
22	Diagramas del módulo de gestión de código	4	100	Diagramas de las clases referentes a la gestión del código.
23	Diagramas del módulo de usuarios	4	100	Diagramas de las clases referentes a la gestión de usuarios.
24	Diagramas de la API	4	100	Diagramas referentes a la API.
25	Especificación técnica del plan de pruebas	3	100	Determinación de las pruebas necesarias para que la aplicación no contenga errores.
26	Reunión con el director del proyecto	3	100	Reunión con el director para validar la fase de diseño.
27	Fin de la fase de diseño	Hito	100	Finalización de la fase de diseño



### 5.1.3 Fase de desarrollo del sistema

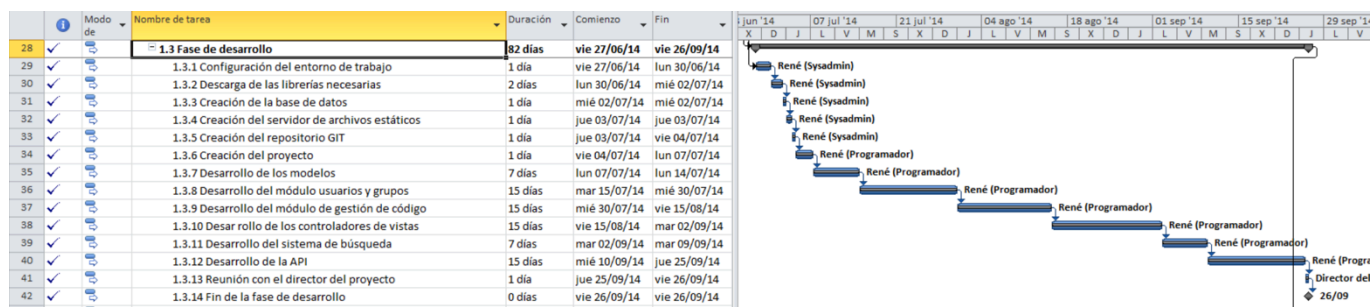


Ilustración 8. Planificación del desarrollo del sistema

La fase de desarrollo ha sido la que más se ha alargado en el tiempo en comparación con todas las demás puesto que ha sido necesario leer documentación a medida que se avanzaba con el desarrollo y el sistema es considerablemente complejo, tanto a nivel de programación como a nivel de puesta en funcionamiento.

Nº	Tarea	Nivel	Avance (%)	Descripción
29	Configuración del entorno de trabajo	3	100	Se configurará todo el entorno de trabajo antes de comenzar con el desarrollo del sistema.
30	Descarga de las librerías necesarias	3	100	Se descargarán e instalarán todas las librerías y otras dependencias necesarias para el funcionamiento del sistema.
31	Creación de la base de datos	3	100	Se creará la base de datos de pruebas sobre la que funcionará todo el sistema.
32	Creación del servidor de archivos estáticos	3	100	Los archivos estáticos se almacenarán en un servidor aparte, independientes de la lógica de la aplicación.
33	Creación del repositorio GIT	3	100	Se creará un repositorio GIT como sistema de gestión de cambios del sistema.
34	Creación del proyecto	3	100	El proyecto debe ser creado utilizando el sistema proporcionado por el framework que se vaya a utilizar.
35	Desarrollo de los modelos	3	100	Desarrollo e implementación de las clases que componen el modelo.
36	Desarrollo del módulo usuarios y grupos	3	100	Desarrollo del módulo de gestión de usuarios y de grupos dentro de la aplicación.
37	Desarrollo del módulo de gestión de código	3	100	Desarrollo del módulo principal, correspondiente a la gestión y organización del código almacenado en la aplicación.

38	Desarrollo de los controladores de vistas	3	100	Desarrollo de los controladores de las vistas de la aplicación e implementación de las plantillas.
39	Desarrollo del sistema de búsqueda	3	100	El sistema incluirá un sistema de búsqueda de código que deberá ser desarrollado.
40	Desarrollo de la API	3	100	Desarrollo e integración en el sistema de la API de acceso desde aplicaciones de terceros.
41	Reunión con el director del proyecto	3	100	Validación por parte del director de la fase de desarrollo principal.
42	Fin de la fase de desarrollo	Hito	100	Finalización de la fase de desarrollo.

### 5.1.4 Fase de desarrollo de elementos secundarios del sitio web

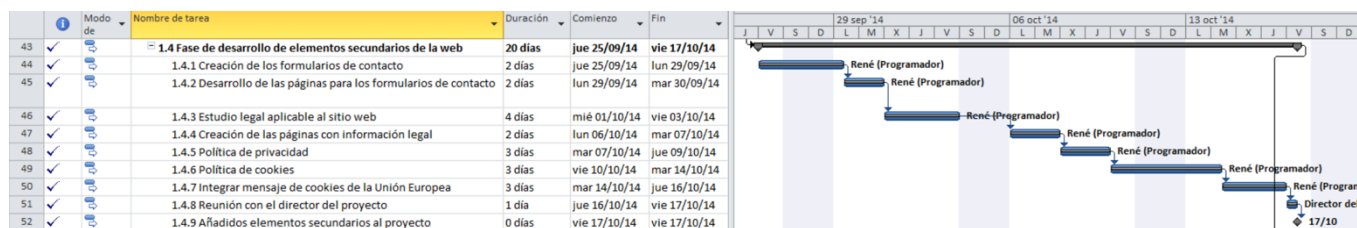


Ilustración 9. Planificación de la fase de desarrollo del sistema

Además de los elementos principales del sitio web, también ha sido necesario planificar la realización de otros elementos, tales como formularios de contacto, estudios legales aplicables, etc. Estos elementos, pese a no estar incluidos en los requisitos iniciales son elementos que deben existir en prácticamente cualquier proyecto web.

Nº	Tarea	Nivel	Avance (%)	Descripción
44	Creación de los formularios de contacto	3	100	El sistema debe incluir diversos formularios de contacto y de notificación de errores.
45	Desarrollo de las páginas para los formularios de contacto	3	100	Deben diseñarse e integrarse en el sitio las páginas que incluyan estos formularios.
46	Estudio legal aplicable al sitio web	3	100	Al tratarse de un sitio que albergará información de usuarios, será necesario realizar un estudio sobre qué legislación afecta al mismo y cómo debe mostrarse esta información en el sitio.
47	Creación de las páginas con información legal	3	100	Deben integrarse las páginas con la información antes mencionada en el sitio.
48	Política de privacidad	3	100	Debe incluirse una página con la política de privacidad del sitio.
49	Política de cookies	3	100	Debe incluirse una página con la política de cookies del sitio.
50	Integrar mensaje de cookies de la Unión Europea	3	100	Debido a la política de cookies de la Unión Europea, se deberá mostrar en el sitio un mensaje informando acerca de ellas.
51	Reunión con el director del proyecto	3	100	Validación por parte del director de los elementos secundarios del sitio web.
52	Final de la fase de desarrollo de elementos secundarios	Hito	100	Finalización de la fase de desarrollo de elementos secundarios.

### 5.1.5 Creación de elementos ajenos al servicio principal

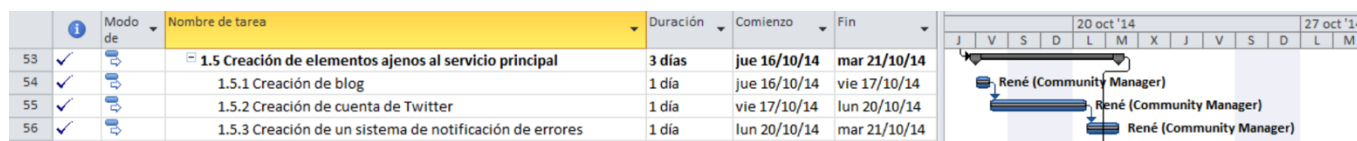


Ilustración 10. Planificación de la creación de elementos ajenos

Para dar a conocer la aplicación y mantenerse en contacto con sus usuarios también ha sido necesaria realizar la planificación de la creación de estos elementos sociales, que en este caso ha sido una de las tareas que menos tiempo ha llevado.

Nº	Tarea	Nivel	Avance (%)	Descripción
54	Creación del blog	3	100	Se creará un blog para informar a los usuarios de las novedades de la aplicación.
55	Creación de cuenta de Twitter	3	100	Cumplirá funciones de comunicación con usuarios.
56	Creación de sistema de notificación de errores	3	100	Se utilizará un sistema de tickets para que los usuarios informen sobre errores o puedan indicar qué mejoras les gustaría que incluyese la aplicación.

### 5.1.6 Despliegue del sistema (alpha 1)

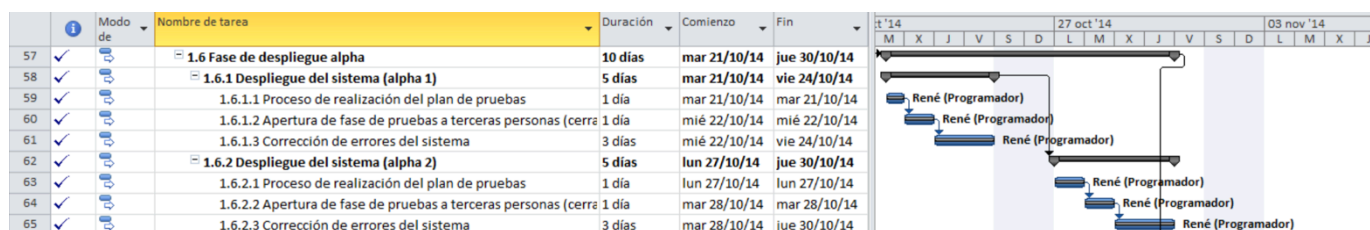


Ilustración 11. Planificación de la alpha 1

Antes de desplegar la versión final del proyecto se han publicado varias alphas y betas, cada una de las cuales corregía los errores notificados en las anteriores fases. Todas estas fases están formadas por la misma secuencia de tareas: realización de un plan de pruebas para determinar los elementos que se probarán, abrir la aplicación al público y corregir los errores de la misma.

Nº	Tarea	Nivel	Avance (%)	Descripción
59	Proceso de realización del plan de pruebas	4	100	Se realizará el paquete de pruebas (manuales y automáticas) establecido en el plan de pruebas.
60	Apertura de fase de pruebas a terceras personas (cerrada)	4	100	Una vez realizadas las pruebas, se desplegará el sistema y se iniciará un proceso de pruebas cerrado con usuarios reales.

61	Corrección de errores del sistema	4	100	Se corregirán los errores notificados por los usuarios del paso anterior.
----	-----------------------------------	---	-----	---

### 5.1.7 Despliegue del sistema (alpha 2)

Nº	Tarea	Nivel	Avance (%)	Descripción
63	Proceso de realización del plan de pruebas	4	100	Se realizará el paquete de pruebas (manuales y automáticas) establecido en el plan de pruebas.
64	Apertura de fase de pruebas a terceras personas (cerrada)	4	100	Una vez realizadas las pruebas, se desplegará el sistema y se iniciará un proceso de pruebas cerrado con usuarios reales.
65	Corrección de errores del sistema	4	100	Se corregirán los errores notificados por los usuarios del paso anterior.

### 5.1.8 Despliegue del sistema (beta 1)

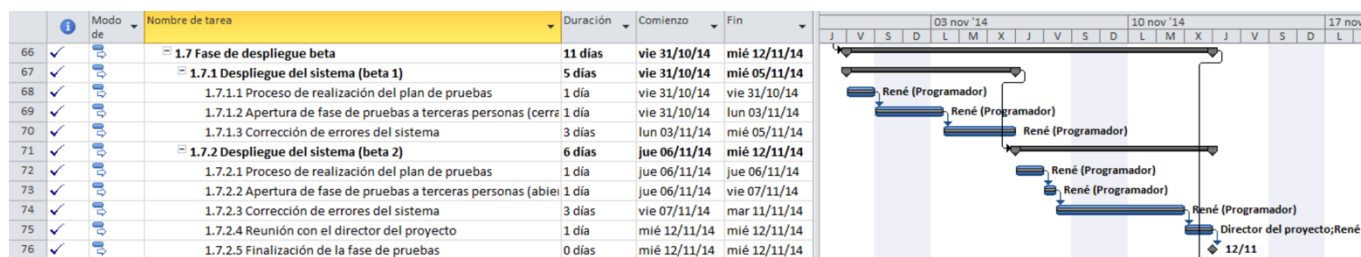


Ilustración 12. Planificación de la beta 1

Nº	Tarea	Nivel	Avance (%)	Descripción
68	Proceso de realización del plan de pruebas	4	100	Se realizará el paquete de pruebas (manuales y automáticas) establecido en el plan de pruebas.
69	Apertura de fase de pruebas a terceras personas (cerrada)	4	100	Una vez realizadas las pruebas, se desplegará el sistema y se iniciará un proceso de pruebas cerrado con usuarios reales.
70	Corrección de errores del sistema	4	100	Se corregirán los errores notificados por los usuarios del paso anterior.

### 5.1.9 Despliegue del sistema (beta 2)

Nº	Tarea	Nivel	Avance (%)	Descripción
72	Proceso de realización del plan de pruebas	4	100	Se realizará el paquete de pruebas (manuales y automáticas) establecido en el plan de pruebas.
73	Apertura de fase de pruebas a terceras personas (abierto)	4	100	Una vez realizadas las pruebas, se desplegará el sistema y se iniciará un proceso de pruebas cerrado con usuarios reales.
74	Corrección de errores del sistema	4	100	Se corregirán los errores notificados por los usuarios del paso anterior.
75	Reunión con el director del proyecto	4	100	Validación por parte del director de la fase de pruebas
76	Finalización de la fase de pruebas	Hito	100	Finalización de la fase de pruebas.



### 5.1.10 Configuración de un sistema de copias de seguridad

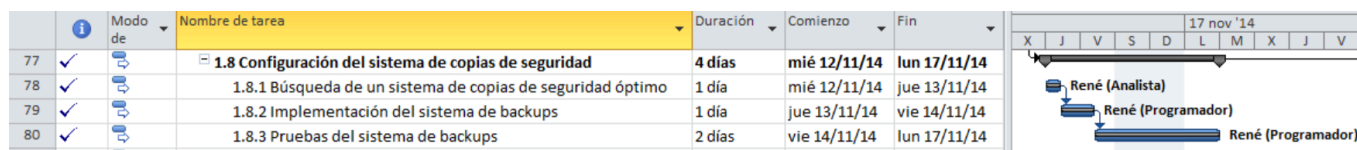


Ilustración 13. Planificación de la configuración del sistema de copias de seguridad

Otra tarea secundaria pero importante que ha sido necesario planificar es la configuración/creación de un sistema de copias de seguridad para almacenar la información del sitio web.

Nº	Tarea	Nivel	Avance (%)	Descripción
78	Búsqueda de un sistema de copias de seguridad óptimo	3	100	Debe buscarse un sistema o metodología para realizar copias de seguridad del proyecto.
79	Implementación del sistema de backups	3	100	Se implementará el sistema de copias de seguridad seleccionado.
80	Pruebas del sistema de backups	3	100	Se verificará el correcto funcionamiento del sistema de copias de seguridad.

### 5.1.11 Documentación

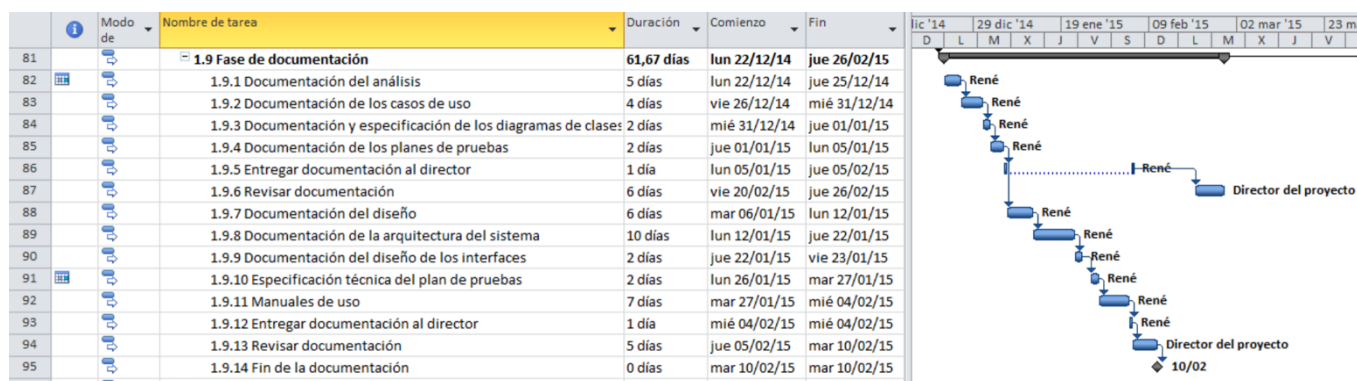


Ilustración 14. Planificación de la documentación

Tras los apartados anteriores, se realiza una planificación de la documentación del sistema.

Nº	Tarea	Nivel	Avance (%)	Descripción
82	Documentación del análisis	3	100	Realización de la documentación de la parte de análisis.
83	Documentación de los casos de uso	3	100	Realización de la documentación de los casos de uso
84	Documentación y especificación de los diagramas	3	100	Realización de la documentación de los



Nº	Tarea	Nivel	Avance (%)	Descripción
	de clases			diagramas de clases.
85	Documentación de los planes de pruebas	3	50	Realización de la documentación de los planes de pruebas.
86	Entregar documentación al director	3	0	
87	Revisar documentación	3	0	
88	Documentación del diseño	3	100	Realización de la documentación del diseño.
89	Documentación de la arquitectura del sistema	3	100	Realización de la documentación de la arquitectura.
90	Documentación del diseño de los interfaces	3	0	Realización de la documentación de las interfaces.
91	Especificación técnica del plan de pruebas	3	0	Realización de la documentación de la especificación del plan de pruebas.
92	Manuales de uso	3	0	Realización de los manuales de usuario de la aplicación.
93	Entregar documentación al director	3	0	Entrega de la documentación al director.
94	Revisar documentación	3	0	Revisión de la documentación por parte del director.
95	Fin de la documentación	Hito	0	Finalización de la documentación

### 5.1.12 Fase de presentación



Ilustración 15. Planificación de la presentación

Todas las tareas involucradas en la presentación del proyecto se encuentran establecidas en este apartado.

Nº	Tarea	Nivel	Avance (%)	Descripción
97	Creación de la presentación	3	0	Realización de la presentación del trabajo.
98	Entregar la presentación al director	3	0	Entrega de la presentación para ser revisada.
99	Revisión de la presentación	3	0	Revisión de la presentación por el director.
100	Entrega de la documentación	3	0	Entrega de la documentación previa a la presentación.
101	Entregables entregados	Hito	0	Se han entregado los entregables.
102	Realización del proyecto	3	0	Realización de la presentación y finalización del proyecto.
103	Fin del proyecto	Hito	0	Finalización del proyecto.

### 5.1.13 Mantenimiento del sitio web (cíclico)

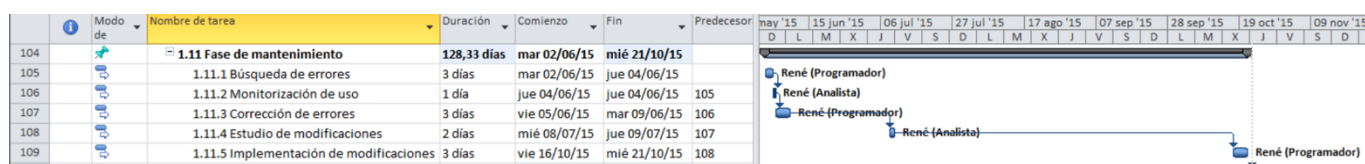


Ilustración 16. Planificación del mantenimiento

Por último, dos apartados que deben realizarse cada cierto tiempo, como es en el caso de este apartado las tareas de mantenimiento del sitio web.

Nº	Tarea	Nivel	Avance (%)	Descripción
100	Búsqueda de errores	3	0	Se recibirán notificaciones de errores tanto de usuarios como de pruebas unitarias realizadas en el sitio.
101	Monitorización de uso	3	0	Se analizará el uso que los usuarios darán al sitio web para buscar posibles mejoras.
102	Corrección de errores	3	0	Se corregirán los errores que se detecten.
103	Estudio de modificaciones	3	0	Se estudiarán las posibles modificaciones para mejorar el sitio web.
109	Implementación de modificaciones	3	0	Se implementarán las modificaciones indicadas.

### 5.1.14 Fase de comunicación (cíclico)

	Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predecesor
110		<b>1.12 Fase de comunicación</b>	2 días?	mié 21/10/15	jue 22/10/15	104
111		1.12.1 Actualización del blog	1 día	mié 21/10/15	mié 21/10/15	
112		1.12.2 Actualización de Twitter	1 día	jue 22/10/15	jue 22/10/15	111

Calendar view: oct '15, 26 oct '15, 02 nov '15. Assignees: René (Community Manager).

Ilustración 17. Planificación de la fase de comunicación

Y en este caso las tareas relativas al *community management* del sitio.

Nº	Tarea	Nivel	Avance (%)	Descripción
111	Actualización del blog	3	0	Se deberá actualizar el blog periódicamente.
112	Actualización de Twitter	3	0	Se deberá atender y actualizar la cuenta de Twitter.

## 5.2 Resumen del presupuesto

En este apartado se muestra un resumen del presupuesto establecido para desarrollar el presente proyecto. Se trata de los costes totales que no serían imputados en su totalidad al cliente final puesto que habría que establecer amortizaciones al hardware, software y también al aprendizaje de las tecnologías necesarias del mismo determinando su utilidad en futuros proyectos para otros clientes.

Se estima que el presupuesto del proyecto será el siguiente:

Ítem	Concepto	Cantidad	Precio Unitario	TOTAL
0	Hardware y conexión a Internet	1	€ 1980, 00	€ 1980, 00
1	Software	1	€ 103, 00	€ 103,00
2	Aprendizaje	1	€ 600, 00	€ 600, 00
3	Desarrollo	1	€ 32.835, 00	€ 32.835, 00
			Subtotal	€ 35.518,00
			IVA (21%)	€ 7391, 00
			<b>TOTAL</b>	<b>€ 42.560</b>

Se puede consultar el presupuesto detallado en el apartado 11 Presupuesto detallado.

## 6 Análisis

### 6.1 Definición del sistema

#### 6.1.1 Determinación del alcance del sistema

El proyecto será una aplicación web realizada en Python utilizando el framework web Django y será alojada en un servidor Apache que tendrá instalados todos los módulos necesarios para el correcto funcionamiento del sistema. Para su funcionamiento será necesario que los usuarios de la misma dispongan de una conexión a internet puesto que la interacción con la misma se realizará a través de un navegador web.

También se podrá acceder a los servicios ofrecidos por el sistema por medio de una API REST.

El principal objetivo de la aplicación, de nombre **CodeSnug** es el de proporcionar a los programadores y desarrolladores de software una plataforma que permita el almacenamiento y clasificación de su código reutilizable para un acceso posterior al mismo de una forma cómoda.

La organización del código se realizará a tres niveles:

- **Snippets:** Es el archivo de código propiamente dicho. Dispondrá de un nombre, un contenido y un lenguaje de programación así como un control de versiones para ver las versiones anteriores del mismo.
- **Goals:** Los Goals (Objetivos en inglés) englobarán uno o varios Archivos que en su conjunto tendrán una finalidad determinada. Por ejemplo un Goal de Backup estaría formado por un archivo con un shell script que realiza el backup de una base de datos y otro archivo con el código para programar la ejecución del mismo mediante CRON.
- **Etiquetas:** Los Goals se podrán etiquetar y clasificar.
- **Workspaces:** Servirán para englobar varios Goals relacionados entre sí y para establecer qué grupos de usuarios tendrán acceso a dichos Goals y sus permisos.

Por otro lado, se podrán crear **Grupos de Usuarios** y asignarlos a los distintos **Workspaces** para que tengan acceso a los Goals que hay en ellos. Los usuarios se añadirán a los grupos de usuarios mediante invitación y éstas podrán ser aceptadas o rechazadas. Un usuario podrá ser eliminado de un grupo de usuarios por el administrador del mismo.

Para que los programadores tengan el control sobre el código que comparten en CodeSnug, podrán exportarlo todo en formato JSON para su propia utilización. Esta es una característica que únicamente estará disponible a través del sitio web, no a través de la API.

Por medio de la **API REST** se podrán realizar todas las acciones que se pueden realizar desde el sitio web salvo el Registro y la exportación de los datos contenidos en la cuenta del usuario.

La aplicación estará disponible en inglés y en español.

## 6.2 Requisitos del sistema

### 6.2.1 Obtención de requisitos del sistema

#### 6.2.1.1 *Requisitos de usuario*

Código	Nombre del Requisito	Descripción del Requisito
<b>R 1.1</b>	Conocimientos básicos sobre el funcionamiento de un sitio web común	Sería que el usuario conociese el funcionamiento de los sitios web y su flujo de trabajo (registro de usuarios, utilización de elementos de formularios, etc).

#### 6.2.1.2 *Requisitos funcionales*

Código	Nombre del Requisito	Descripción del Requisito
<b>R 2.1</b>	Registro de Usuario	La aplicación debe permitir el registro de usuarios.
<b>R 2.2</b>	Login de Usuario	La aplicación debe permitir el login de usuarios.
<b>R 2.3</b>	Editar datos de Usuario	Un usuario podrá editar sus datos personales.
<b>R 2.4</b>	Resetear contraseña de Usuario	Un usuario podrá resetear y recuperar su contraseña en el caso de que sea necesario.
<b>R 2.5</b>	Crear Goal	Un usuario deberá poder Goals.
<b>R 2.6</b>	Editar Goal	Los Goals creados por el usuario podrán ser editados por él o por los usuarios a los que le conceda permiso.
<b>R 2.7</b>	Eliminar Goal	Un usuario podrá eliminar los Goal que ha creado, al igual que los usuarios a los que le ha otorgado permisos
<b>R 2.8</b>	Crear Snippet	El sistema debe permitir la creación de archivos de código.
<b>R 2.9</b>	Editar Snippet	Los archivos añadidos a un Goal se deben poder editar posteriormente.
<b>R 2.10</b>	Eliminar Snippet	Los archivos añadidos a un Goal deben poder eliminarse.
<b>R 2.11</b>	Añadir Archivo a Goal	Los archivos de código serán asignables a los Goal.
<b>R 2.12</b>	Crear Etiqueta	Se podrán crear etiquetas para clasificar los Goal.
<b>R 2.13</b>	Editar Etiqueta	Las etiquetas podrán ser editadas.

<b>R 2.14</b>	Eliminar Etiqueta	Será posible eliminar las etiquetas.
<b>R 2.15</b>	Asignar Etiqueta a Goal	Una etiqueta será asignable a un Goal para clasificarlo.
<b>R 2.16</b>	Crear Workspace	Se podrán crear Workspaces para organizar los Goal.
<b>R 2.17</b>	Editar Workspace	Un Workspace se podrá editar.
<b>R 2.18</b>	Eliminar Workspace	Los Workspaces podrán ser eliminados.
<b>R 2.19</b>	Asignar Workspace a Goal	Los Workspaces podrán ser asignados a los Goal.
<b>R 2.20</b>	Crear Grupo de Usuarios	Se podrán crear grupos de usuarios.
<b>R 2.21</b>	Editar Grupo de Usuarios	Los grupos de usuarios serán editables.
<b>R 2.22</b>	Modificar permisos a grupo de usuarios	Los permisos de los grupos de usuarios se podrán modificar para establecer qué tareas pueden realizar sobre los Goals de un Workspace asociado.
<b>R 3.23</b>	Eliminar Grupo de Usuarios	Se podrán eliminar los grupos de usuario.
<b>R 3.24</b>	Asignar Grupo de Usuarios a Workspace	Un grupo de usuarios podrá ser asignado a un Workspace.
<b>R 3.25</b>	Invitar Usuario a Grupo de Usuarios	Se podrán enviar invitaciones a usuarios para que se unan a un determinado grupo de usuarios.
<b>R 3.26</b>	Rechazar Invitación a Grupo de Usuarios	Un usuario puede rechazar una invitación a un grupo de usuarios.
<b>R 3.27</b>	Eliminar Usuario de Grupo de Usuarios	Los usuarios añadidos a un grupo de usuarios pueden ser eliminados, dejando de formar parte de ese grupo de usuarios.
<b>R 3.28</b>	Exportar Goal	Los Goal de un usuario podrán ser exportados fuera del sistema.

#### 6.2.1.2.1 Requisitos Tecnológicos

Código	Nombre del Requisito	Descripción del Requisito
<b>R 3.1</b>	Conexión a Internet	El usuario debe disponer de una conexión a Internet para utilizar la aplicación web.

### 6.2.1.2.2 Requisitos de seguridad

Código	Nombre del Requisito	Descripción del Requisito
R 4.1	Almacenamiento de contraseñas	Las contraseñas deberán almacenarse cifradas en la base de datos.
R 4.2	Cifrado de transmisiones	Todas las transmisiones con el sitio web estarán cifradas y utilizarán SSL.
R 4.3	Seguridad en formularios	Todos los formularios dispondrán de campos CSFR de verificación para evitar la suplantación de sesiones.

## 6.2.2 Identificación de los actores del sistema

A lo largo del tiempo de utilización de la aplicación intervienen los siguientes actores:

### 6.2.2.1 Aplicación web (sistema)

Es el programa desarrollado, sobre el que se realizarán todas las operaciones y sobre el que actuará directamente el usuario.

### 6.2.2.2 Usuario de la aplicación web

Será quien realizará todas las actividades para las que está diseñada la aplicación. Debe contar con conocimientos básicos de la utilización de sitios web y debería disponer de conocimientos de programación dada la naturaleza del sistema.

### 6.2.2.3 Usuario Administrador

Será el encargado de gestionar el sitio web y dispondrá de permisos especiales para garantizar su correcto funcionamiento, así como una gestión de usuarios avanzada.

### 6.2.2.4 Aplicación de terceros

Interactuará con el sistema utilizando los servicios web proporcionados por el mismo.



### 6.2.3 Identificación de los subsistemas

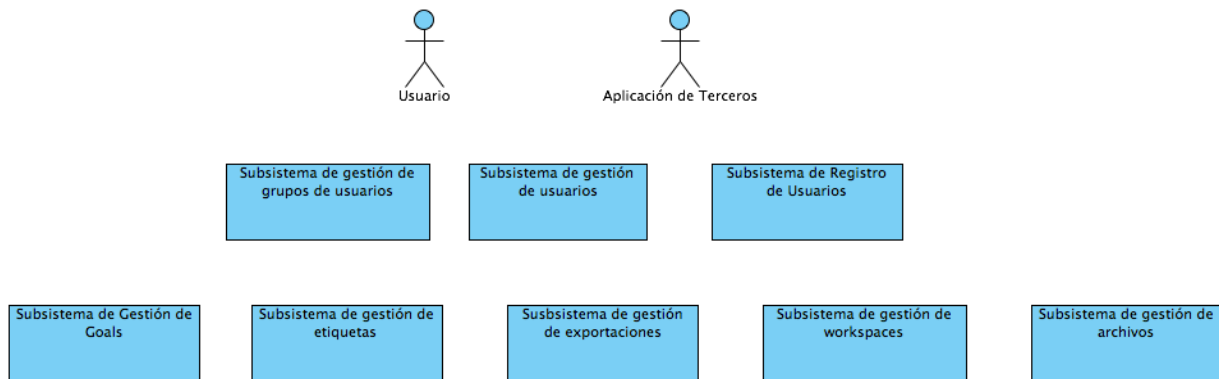


Ilustración 18. Subsistemas de la aplicación

A continuación se describen brevemente los subsistemas que forman parte de la aplicación.

#### 6.2.3.1 *Subsistema Gestión de Usuarios*

El subsistema de gestión de Usuarios es el encargado realizar todas las operaciones relativas a los usuarios del sistema y su inclusión y borrado de grupos de usuarios. Adicionalmente se encarga de su persistencia, y de toda la lógica de negocio implicada en su ciclo de vida.

#### 6.2.3.2 *Subsistema Registro de Usuarios*

Proporciona todas las funcionalidades relacionadas con esta parte del modelo, como pueden ser las operaciones de creación, modificación, listado y borrado de los mismos, así como el login y el control de acceso a las diferentes áreas de la aplicación

#### 6.2.3.3 *Subsistema de gestión de grupos de Usuarios*

Proporciona las funcionalidades relacionadas con la gestión de los grupos de usuarios: creación, modificación, eliminación y listado y la gestión de su persistencia en el sistema.

#### 6.2.3.4 *Subsistema de gestión de goals*

Proporciona las funcionalidades relacionadas con la gestión de los goals: creación, modificación, eliminación y listado y la gestión de su persistencia en el sistema.

#### 6.2.3.5 *Subsistema de gestión de etiquetas*

Proporciona las funcionalidades relacionadas con la gestión de las etiquetas: creación, modificación, eliminación y listado y la gestión de su persistencia en el sistema.

#### 6.2.3.6 *Subsistema de gestión de exportaciones*

Proporciona las funcionalidades relacionadas con exportación de los datos almacenados por los usuarios en la aplicación en distintos formatos.

#### 6.2.3.7 *Subsistema de gestión de workspaces*

Proporciona las funcionalidades relacionadas con la gestión de los workspaces: creación, modificación, eliminación y listado y la gestión de su persistencia en el sistema.

### 6.2.3.8 Subsistema de gestión de archivos (snippets)

Proporciona las funcionalidades relacionadas con la gestión de los snippets: creación, modificación, eliminación y listado y la gestión de su persistencia en el sistema.

## 6.2.4 Especificación de los casos de uso

### 6.2.4.1 Registro de Usuarios

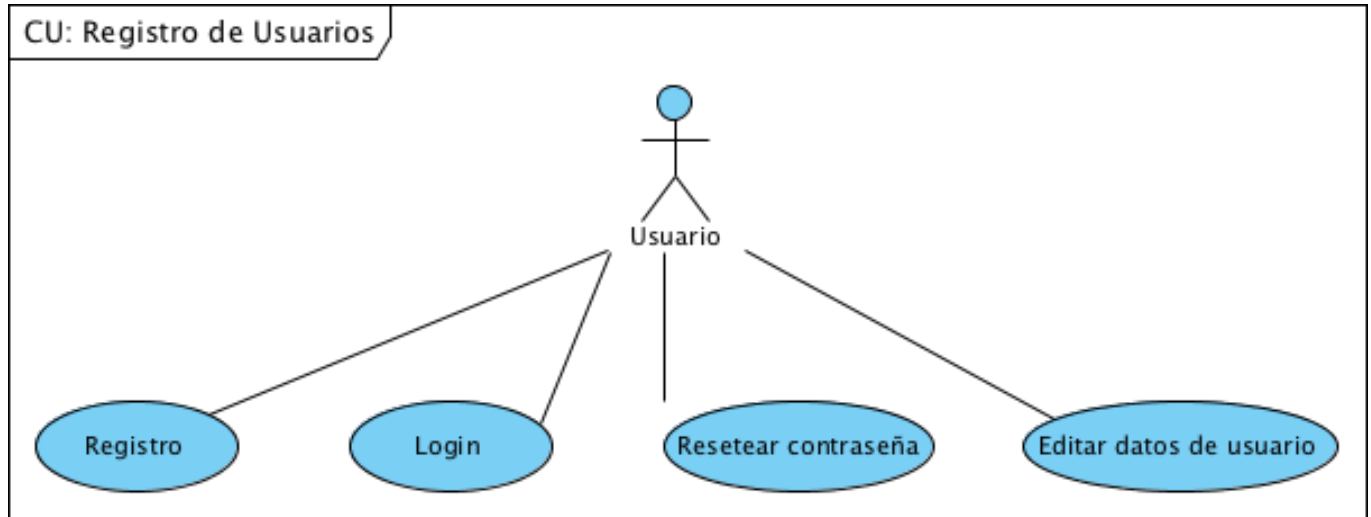


Ilustración 19. Diagrama de casos de uso de registro de usuarios

En un sistema que permite múltiples usuarios es básico disponer de un sistema de registro de usuarios si se desea que éstos interactúen con el mismo de manera autónoma. Por ello, los usuarios podrán registrarse y acceder a la aplicación introduciendo los datos solicitados por la misma por medio de diversos formularios.

Teniendo en cuenta la existencia de la posibilidad de que un usuario se olvide de su contraseña, se le proporcionará la opción de resetear la misma.

El usuario también podrá editar y actualizar sus datos desde su perfil.

#### 6.2.4.1.1 Caso de uso Registro

**Precondiciones:** N/A

**Descripción:** El usuario se dispone a darse de alta como nuevo usuario.

**Actores:** Usuario

**Escenario principal:**

1. El usuario accede al formulario de alta de usuarios.
2. Rellena debidamente todos los campos del formulario.
3. Acepta el formulario.
4. El sistema guarda el nuevo usuario en base de datos.

**Escenario alternativo 1:** El usuario no completa debidamente todos los campos del formulario.

1. El sistema aborta la creación del usuario en base de datos, y muestra el formulario de creación, señalando los errores cometidos en aquellos campos que tuviesen errores.
2. Se permite al usuario modificar los datos y retomar el ciclo descrito en el escenario principal

#### 6.2.4.1.2 Caso de uso Login

**Precondiciones:** El usuario está registrado

**Descripción:** El usuario se dispone a acceder al sitio web.

**Actores:** Usuario

**Escenario principal:**

1. El usuario accede al formulario de acceso.
2. Rellena debidamente todos los campos del formulario.
3. Acepta el formulario.
4. El sistema redirige al usuario a su cuenta.

**Escenario alternativo 1:** El usuario no completa debidamente todos los campos del formulario.

5. El sistema aborta el acceso del usuario en el sistema, y muestra el formulario de acceso nuevamente, señalando los errores cometidos en aquellos campos que tuviesen errores.
6. Se permite al usuario modificar los datos y retomar el ciclo descrito en el escenario principal

#### 6.2.4.1.3 Caso de uso Resetear Contraseña

**Precondiciones:** El usuario está registrado y no recuerda su contraseña.

**Descripción:** El usuario se dispone a acceder al sitio web pero se ha olvidado de su contraseña.

**Actores:** Usuario

**Escenario principal:**

1. El usuario accede al formulario de acceso.
2. El usuario se da cuenta de que no recuerda su contraseña, por lo que pulsa el botón de resetear contraseña.
3. El usuario introduce su email en los campos del formulario.
4. Acepta el formulario.
5. El usuario hace clic en el enlace que le llegará a su correo.
6. El usuario rellena el formulario con su nueva contraseña.

**Escenario alternativo 1:** Al usuario no le llega el email de reseteo de contraseña

7. El usuario envía un email al administrador para solicitarle estos datos.
8. El administrador verifica la identidad del usuario y le notifica su email.

9. El usuario vuelve a proceder con el reseteo.

#### 6.2.4.1.4 Caso de uso Editar Datos de Usuario

**Precondiciones:** El usuario está registrado y puede acceder a su cuenta

**Descripción:** El usuario desea modificar los datos de usuario de su perfil

**Actores:** Usuario

**Escenario principal:**

1. El usuario accede su página de perfil.
2. El usuario procede a editar sus datos.
3. El usuario salva sus datos y estos son actualizados en la base de datos.

**Escenario alternativo 1:** El usuario no completa debidamente todos los campos del formulario.

1. El sistema aborta el acceso del usuario en el sistema, y muestra el formulario de acceso nuevamente, señalando los errores cometidos en aquellos campos que tuviesen errores.
2. Se permite al usuario modificar los datos y retomar el ciclo descrito en el escenario principal

#### 6.2.4.2 Subsistema Gestión de Goals

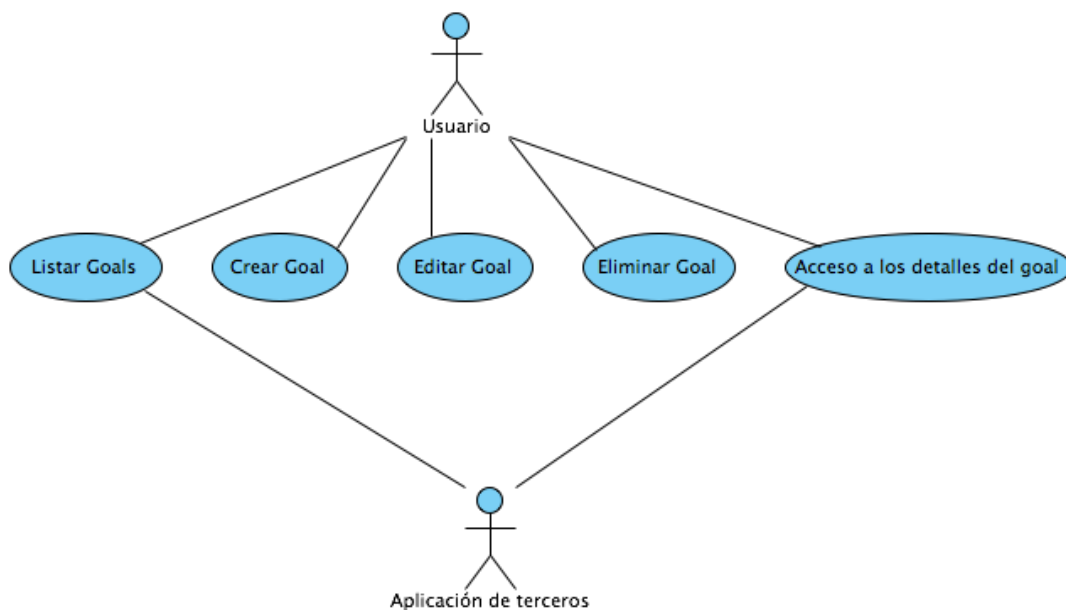


Ilustración 20. Diagrama de caso de uso de Gestión de Goals

El objetivo principal de la aplicación es el de almacenar código y mantenerlo organizado. El núcleo fundamental de esta organización son los Goal (Objetivo, en inglés). Por tanto, la aplicación permitirá a sus usuarios crearlos, editarlos (incluyendo la edición de los miembros que lo conforman) y eliminarlos.

Tanto los usuarios de la aplicación web como los de aplicaciones de terceros por medio de la API tendrán la posibilidad de hacerlo.

#### 6.2.4.2.1 Caso de uso Listar Goals

**Precondiciones:** El usuario dispone de acceso al sistema y ha creado goals.

**Descripción:** El usuario desea acceder al listado de los goals.

**Actores:** Usuario

**Escenario principal: Interfaz web**

1. El usuario accede a la página de listado de goals.
2. El sistema carga de la base de datos los goals correspondientes y los muestra en la página.

**Escenario alternativo 1: API**

1. El usuario accede a la URL de la API correspondiente al listado de los goals, indicando el identificador del Goal, su usuario y contraseña.
2. El sistema detecta que se trata de una llamada a la API y devuelve los datos en el formato solicitado (JSON).

#### 6.2.4.2.2 Caso de uso Crear Goal

**Precondiciones:** El usuario está registrado y ha accedido al sistema.

**Descripción:** El usuario desea crear un Goal en la aplicación.

**Actores:** Usuario

**Escenario principal:**

3. El usuario accede al formulario de creación de goals.
4. El usuario complementa todos los datos requeridos por el goal.
5. El usuario pulsa el botón de guardar.
6. El Goal es almacenado en el sistema.

**Escenario alternativo 1:** Ya existe un goal con ese nombre

1. El sistema recarga la página con los datos introducidos en el formulario previamente, señalando el error en el campo de título con un texto descriptivo.
2. Se permite al usuario modificar los datos y retomar el ciclo descrito en el escenario principal

#### 6.2.4.2.3 Caso de uso Editar Goal

**Precondiciones:** El Goal ya está creado en el sistema

**Descripción:** El usuario desea editar un Goal ya existente.

**Actores:** Usuario

**Escenario principal:**

1. El usuario accede a la página de detalles de alguno de sus goals o de algún goal al que tiene acceso.

2. El usuario actualiza los campos que quiera actualizar del goal.
3. El usuario pulsa el botón de guardar.
4. El goal es guardado en el sistema con los nuevos datos.

**Escenario alternativo 1:** El usuario no dispone de permisos de edición sobre el goal

1. El sistema no permite al usuario acceder a la página de edición del goal y lo redirige a otra página.

#### 6.2.4.2.4 Caso de uso Eliminar Goal

**Precondiciones:** El Goal ya está creado en el sistema

**Descripción:** El usuario desea eliminar un Goal ya existente.

**Actores:** Usuario

**Escenario principal:**

1. El usuario accede a la página de detalles de alguno de sus goals o de algún goal al que tiene acceso.
2. El usuario pulsa el botón de borrar.
3. El goal y todos sus snippets son borrados del sistema.

**Escenario alternativo 1:** El usuario no es el creador del goal

1. El sistema no permite al usuario acceder a la página de edición del goal y lo redirige a otra página.

#### 6.2.4.2.5 Caso de uso Acceso a los detalles del goal

**Precondiciones:** El Goal ya está creado en el sistema

**Descripción:** El usuario desea acceder a la información de un Goal existente.

**Actores:** Usuario

**Escenario principal: Interfaz web**

2. El usuario accede a la página de detalles de alguno de sus goals o de algún goal al que tiene acceso.

**Escenario alternativo 1: API**

3. El usuario accede a la URL de la API correspondiente a los detalles del Goal, indicando el identificador del Goal, su usuario y contraseña.
4. El sistema detecta que se trata de una llamada a la API y devuelve los datos en el formato solicitado (JSON).

### 6.2.4.3 Subsistema Gestión de Archivos (snippets)

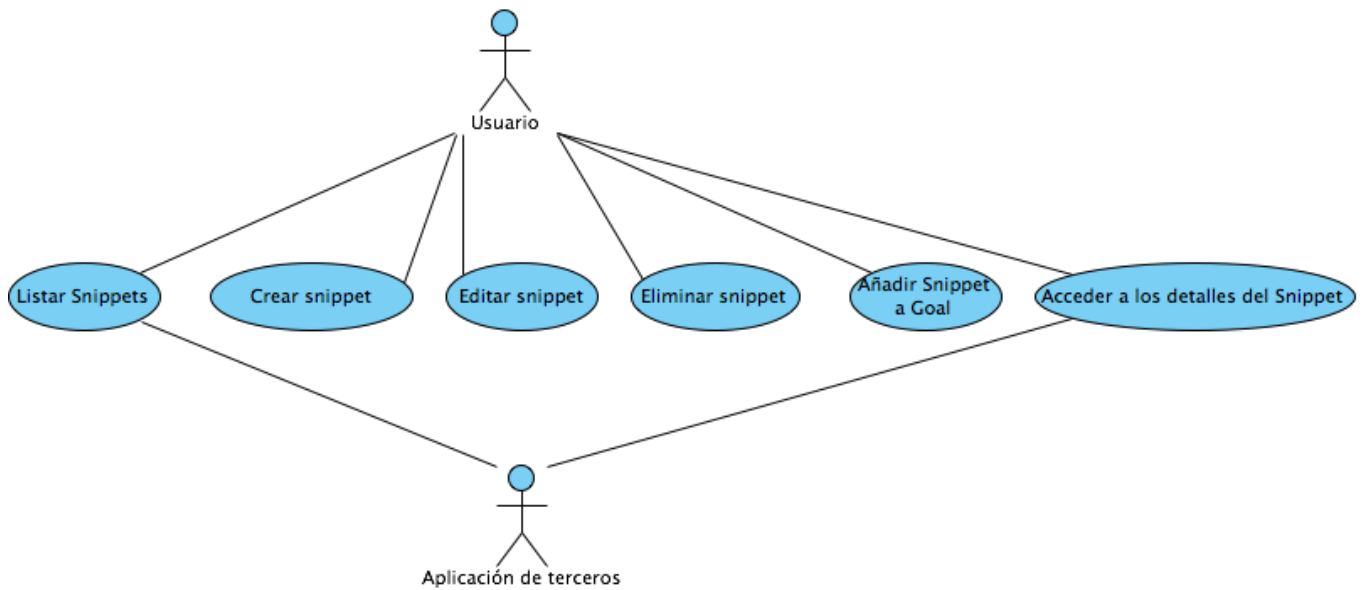


Ilustración 21. Diagrama de casos de uso de Gestión de Archivos

Como se mencionó anteriormente, un Goal estará formado por varios archivos de código (snippets). Estos archivos tendrán diversas propiedades y podrán ser creados, editados, eliminados y añadidos a un Goal.

Tanto los usuarios de la aplicación web como los de aplicaciones de terceros por medio de la API tendrán la posibilidad de hacerlo.

#### 6.2.4.3.1 Caso de uso Listar Snippets

**Precondiciones:** El usuario dispone de acceso al sistema y ha creado goals.

**Descripción:** El usuario desea acceder al listado de los snippets de un Goal.

**Actores:** Usuario

**Escenario principal: Interfaz web**

1. El procedimiento es el mismo que para acceder a los detalles de un goal.

**Escenario alternativo 1: API**

2. El usuario accede a la URL de la API correspondiente al listado de los snippets, indicando el identificador del Snippet, su usuario y contraseña.
3. El sistema detecta que se trata de una llamada a la API y devuelve los datos en el formato solicitado (JSON).

#### 6.2.4.3.2 Caso de uso Crear Archivo (snippet)

**Precondiciones:** El usuario está registrado y ha accedido al sistema.

**Descripción:** El usuario desea añadir un Snippet a uno de sus Goals.

**Actores:** Usuario

**Escenario principal:**

1. El usuario accede a la página de edición de uno de los goals que ya ha creado.
2. El usuario pulsa el botón correspondiente a añadir un nuevo snippet.
3. El sistema despliega los campos del formulario del nuevo Snippet.
4. El usuario cumplimenta todos los campos del formulario.
5. El usuario pulsa el botón de guardar.
6. El Goal es almacenado en el sistema con su nuevo Snippet.

**Escenario alternativo 1:** El usuario no rellena alguno de los campos requeridos.

1. El sistema recarga la página con los datos introducidos en el formulario previamente, señalando el error en el campo de título con un texto descriptivo.
2. Se permite al usuario modificar los datos y retomar el ciclo descrito en el escenario principal

*6.2.4.3.3 Caso de uso Eliminar archivo (snippet) de Goal*

**Precondiciones:** El usuario ha creado un goal en el sistema y ha añadido al menos dos snippets al mismo.

**Descripción:** El usuario desea eliminar un snippet de uno de sus goal.

**Actores:** Usuario

**Escenario principal:**

1. El usuario accede a la página de edición de alguno de sus goals o de algún goal al que tiene acceso.
2. El usuario pulsa el botón de eliminar snippet debajo de los campos del snippet deseado.
3. El usuario pulsa el botón guardar.
4. El sistema elimina el snippet del goal y actualiza la base de datos.

**Escenario alternativo 1:** El goal solamente dispone de un snippet

1. El sistema no permite muestra al usuario botón de eliminar snippet puesto que un goal debe contener al menos un snippet.

*6.2.4.3.4 Caso de uso Acceso a los detalles de un Snippet*

**Precondiciones:** El Goal que contiene el snippet ya está creado en el sistema

**Descripción:** El usuario desea acceder a la información de un Snippet existente.

**Actores:** Usuario

**Escenario principal: Interfaz web**

1. El usuario accede a la página de detalles de alguno de sus goals o de algún goal al que tiene acceso.

**Escenario alternativo 1: API**



2. El usuario accede a la URL de la API correspondiente a los detalles del Snippet, indicando el identificador del Snippet, su usuario y contraseña.
3. El sistema detecta que se trata de una llamada a la API y devuelve los datos en el formato solicitado (JSON).

#### 6.2.4.4 Subsistema de Gestión de Etiquetas

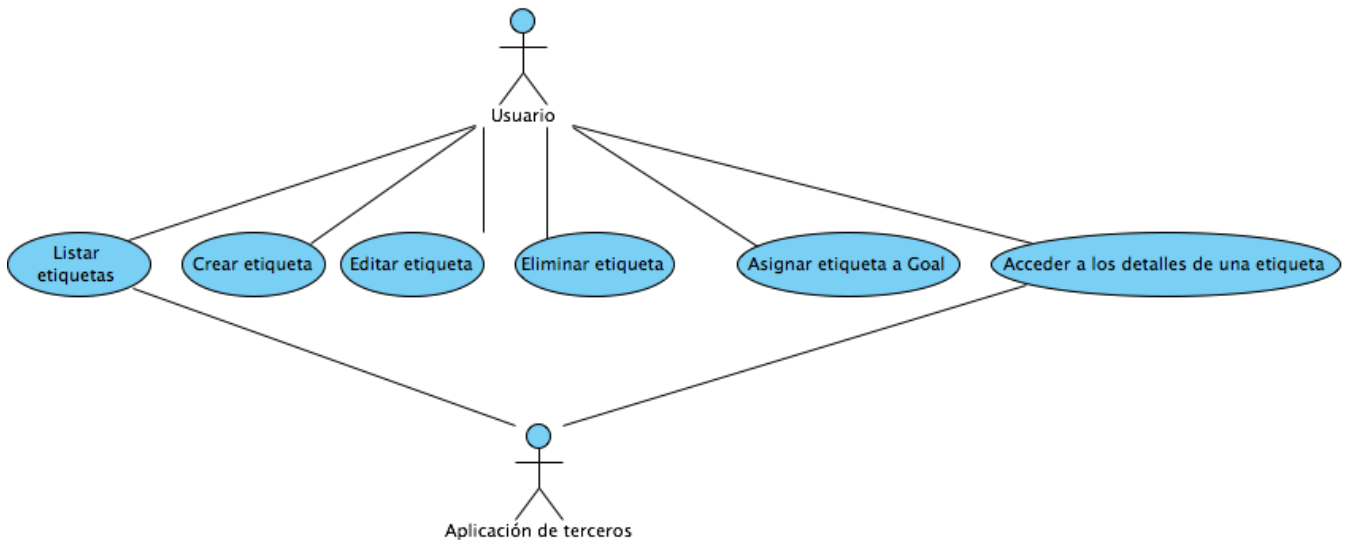


Ilustración 22. Diagrama de casos de uso de Gestión de Etiquetas

Los usuarios de la aplicación web y también los usuarios de aplicaciones de terceros podrán gestionar las etiquetas de los Goal. Estas etiquetas podrán ser creadas, editadas, eliminadas y asignadas a un Goal, de tal forma que éste quede clasificado por ellas.

Puesto que el flujo de creación, edición y eliminación de etiquetas se realiza de manera similar a la de un Goal (por medio de formularios con validaciones) no se detallarán sus escenarios. Si se desea, se pueden consultar los escenarios del subsistema de gestión de Goals en 6.2.4.2.

##### 6.2.4.4.1 Caso de uso Asignar Etiqueta a Goal

**Precondiciones:** El usuario está registrado y dispone de acceso al sistema.

**Descripción:** El usuario desea asignar una etiqueta ya creada a uno de sus goals.

**Actores:** Usuario

**Escenario principal:**

1. El usuario accede a la página de edición de alguno de sus goals.
2. El usuario selecciona en la barra lateral el checkbox de la etiqueta que desea asignar al Goal.
3. El usuario pulsa el botón guardar.
4. El sistema guarda el goal asignándole la nueva etiqueta.

**Escenario alternativo 1:** El usuario desea añadir una etiqueta que no existe.

1. El usuario pulsa el botón correspondiente a añadir nueva etiqueta en la página de edición de goal.

2. El sistema muestra un campo para el nombre de la etiqueta.
3. El usuario pulsa el botón Añadir Etiqueta.
4. La nueva etiqueta es almacenada en la base de datos del sistema y asignada automáticamente al Goal mediante AJAX.
5. El usuario pulsa el botón guardar.
6. El sistema actualiza los datos del goal añadiéndole la etiqueta recién creada.

#### 6.2.4.5 Subsistema de Gestión de Exportaciones

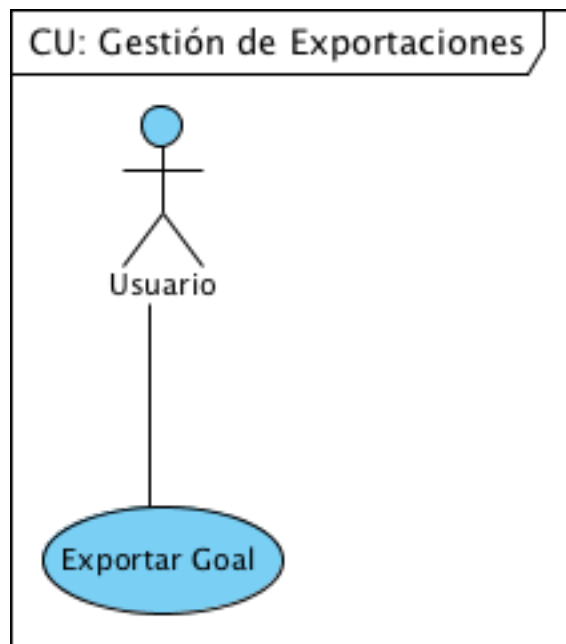


Ilustración 23. Diagrama de casos de uso de Gestión de Exportaciones

Los usuarios de la aplicación web podrán exportar sus Goals en determinados formatos a modo de backups o para ser utilizados en aplicaciones de terceros.

##### 6.2.4.5.1 Caso de uso Exportar Goal

**Precondiciones:** El usuario está registrado y dispone de acceso al sistema.

**Descripción:** El usuario desea exportar el código de uno de sus goals.

**Actores:** Usuario

##### Escenario principal:

1. El usuario accede a la página de detalles de alguno de sus goals o de un goal al que tiene acceso.
2. El pulsa el botón de descarga de Goal.
3. El sistema procesa la petición y serializa el contenido del goal.
4. El sistema genera una descarga del archivo con el código en formato txt.
5. El usuario se descarga el código.

**Escenario alternativo 1:** El usuario desea descargar todo el código almacenado en la aplicación.

1. El usuario accede a su página de perfil.
2. El usuario pulsa el botón de descargar datos.
3. El sistema procesa la petición y serializa el contenido de todos sus datos en formato JSON.
4. El sistema genera una descarga del archivo en formato JSON.
5. El usuario se descarga el archivo.

### 6.2.4.6 Subsistema de Gestión de Workspaces

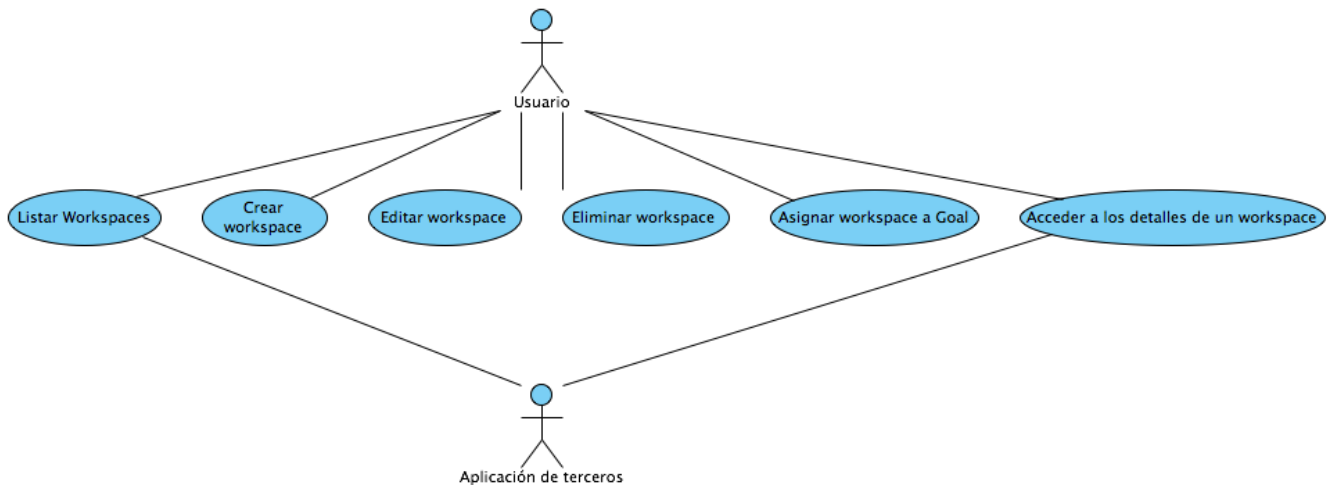


Ilustración 24. Diagrama de casos de uso de Gestión de Workspaces

Los Workspaces se utilizarán para organizar los Goals y los grupos de usuarios que tienen acceso a ellos. Tanto los usuarios de la aplicación web como los usuarios de aplicaciones de terceros podrán crearlos, editarlos, eliminarlos y asignar workspaces existentes a uno o varios Goals.

Puesto que el flujo de creación, edición y eliminación de workspaces se realiza de manera similar a la de un Goal (por medio de formularios con validaciones) no se detallarán sus escenarios. Si se desea, se pueden consultar los escenarios del subsistema de gestión de Goals en 6.2.4.2.

#### 6.2.4.6.1 Caso de uso Asignar Workspace a Goal

**Precondiciones:** El usuario está registrado y dispone de acceso al sistema.

**Descripción:** El usuario desea asignar un workspace ya creada a uno de sus goals.

**Actores:** Usuario

**Escenario principal:**

1. El usuario accede a la página de edición de alguno de sus goals.
2. El usuario selecciona en la barra lateral el checkbox del workspace que desea asignar al Goal.
3. El usuario pulsa el botón guardar.
4. El sistema guarda el goal asignándole el nuevo goal.

**Escenario alternativo 1:** El usuario desea añadir un workspace que no existe.

1. El usuario pulsa el botón correspondiente a añadir nuevo workspace en la página de edición de goal.
2. El sistema muestra un campo para el nombre del workspace.
3. El usuario pulsa el botón Añadir Workspace.
4. El nuevo workspace es almacenado en la base de datos del sistema y asignado automáticamente al Goal mediante AJAX.
5. El usuario pulsa el botón guardar.

6. El sistema actualiza los datos del goal añadiéndole el workspace recién creado.

#### 6.2.4.7 Subsistema de Gestión de grupos de usuarios

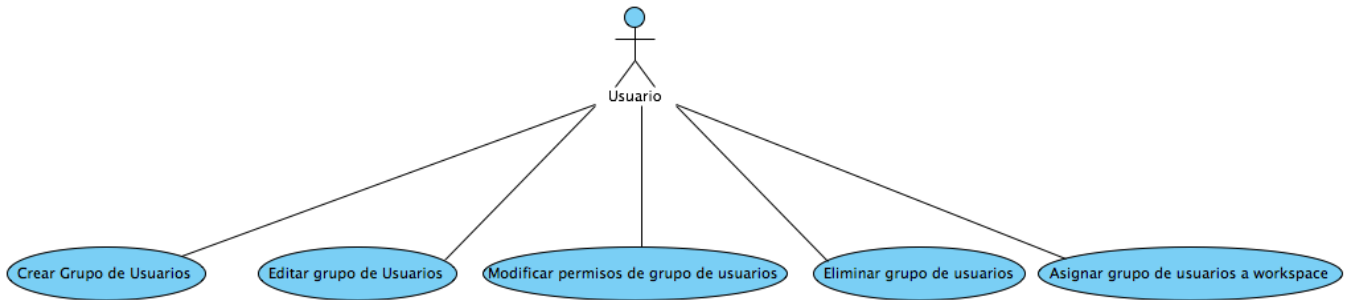


Ilustración 25. Diagrama de casos de uso de Gestión de grupos de usuarios

Para mantener una organización de los usuarios que tienen acceso a un determinado workspace se utilizarán los grupos de usuario. Estos grupos de usuario podrán gestionarse desde el sitio web y desde aplicaciones externas.

Las acciones que se podrán llevar a cabo sobre los grupos de usuario son la creación, edición, establecer permisos del grupo de usuarios, eliminarlo y asignarlo a un Workspace.

Puesto que el flujo de creación, edición y eliminación de grupos de usuarios se realiza de manera similar a la de un Goal (por medio de formularios con validaciones) no se detallarán sus escenarios. Si se desea, se pueden consultar los escenarios del subsistema de gestión de Goals en 6.2.4.2.

#### 6.2.4.7.1 Caso de uso *Modificar los permisos de un grupo de usuarios*

**Precondiciones:** El usuario está registrado, dispone de acceso al sistema y ha creado un grupo de usuarios al que le ha asignado un workspace.

**Descripción:** El usuario desea cambiar los permisos de un grupo sobre un determinado workspace.

**Actores:** Usuario

##### **Escenario principal:**

7. El usuario accede a la página de edición de un grupo de usuario.
8. El usuario selecciona en el desplegable correspondiente al workspace el permiso deseado.
9. El sistema modifica automáticamente los permisos de grupo de usuarios mediante una petición AJAX.

#### 6.2.4.7.2 Caso de uso *Asignar Grupo de Usuarios a Workspace*

**Precondiciones:** El usuario está registrado y dispone de acceso al sistema.

**Descripción:** El usuario desea asignar un workspace a un grupo de usuarios.

**Actores:** Usuario

##### **Escenario principal:**

1. El usuario accede a la página de edición de un grupo de usuarios.
2. El usuario pulsa el botón de añadir nuevo workspace.
3. El sistema redirige al usuario a la página de asignación de workspace.
4. El usuario selecciona en los desplegables el nombre del workspace y los permisos que dispondrán los usuarios del grupo de usuarios.
5. El usuario pulsa el botón guardar.
6. El sistema almacena la nueva información sobre el workspace del grupo de usuarios en la base de datos.
7. El sistema redirige al usuario a la página de detalles del grupo de usuarios.

**Escenario alternativo 1:** El workspace ya ha sido asignado previamente al grupo de usuarios

1. Se muestra un mensaje de error al usuario indicando que ese workspace ya está asignado.

#### 6.2.4.8 Subsistema de Gestión de usuarios

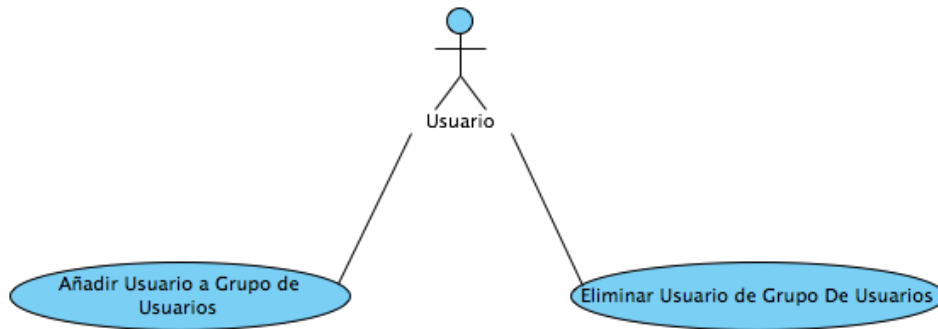


Ilustración 26. Diagrama de casos de uso de Gestión de usuarios

Tanto los usuarios de la aplicación web como de aplicaciones de terceros podrán gestionar qué usuarios pertenecen a un grupo de usuarios. Se podrán añadir usuarios a grupos de usuarios y el gestor de un grupo de usuarios podrá eliminar a un usuario de uno de ellos.

##### 6.2.4.8.1 Caso de uso Asignar Usuario a Grupo de Usuarios

**Precondiciones:** El usuario está registrado y dispone de acceso al sistema.

**Descripción:** El usuario desea asignar un usuario a un grupo de usuarios.

**Actores:** Usuario

**Escenario principal:**

1. El usuario accede a la página de edición de un grupo de usuarios.
2. El usuario pulsa el botón de añadir nuevo usuario.
3. El sistema redirige al usuario a la página de asignación de usuarios.
4. El usuario selecciona en los desplegables el nombre del usuario que desea añadir.
5. El usuario pulsa el botón guardar.
6. El sistema almacena la nueva información sobre el usuario del grupo de usuarios en la base de datos.
7. El sistema redirige al usuario a la página de detalles del grupo de usuarios.

**Escenario alternativo 1:** El usuario ya ha sido asignado previamente al grupo de usuarios

8. Se muestra un mensaje de error al usuario indicando que ese usuario ya está asignado.

##### 6.2.4.8.2 Caso de uso Eliminar usuario de grupo de usuarios

**Precondiciones:** El usuario está registrado y dispone de acceso al sistema.

**Descripción:** El usuario desea eliminar un usuario añadido a un grupo de usuarios.

**Actores:** Usuario

**Escenario principal:**

1. El usuario accede a la página de edición de un grupo de usuarios.

2. El usuario pulsa el botón de borrar junto al usuario que desea borrar.
3. El sistema muestra un popup para que el usuario confirme el borrado del usuario.
4. El usuario pulsa el botón de confirmación.
5. El sistema elimina al usuario del grupo de usuarios y recarga la página para reflejar los cambios realizados.



## 6.3 Análisis de interfaces de usuario

### 6.3.1 Diagrama de navegabilidad

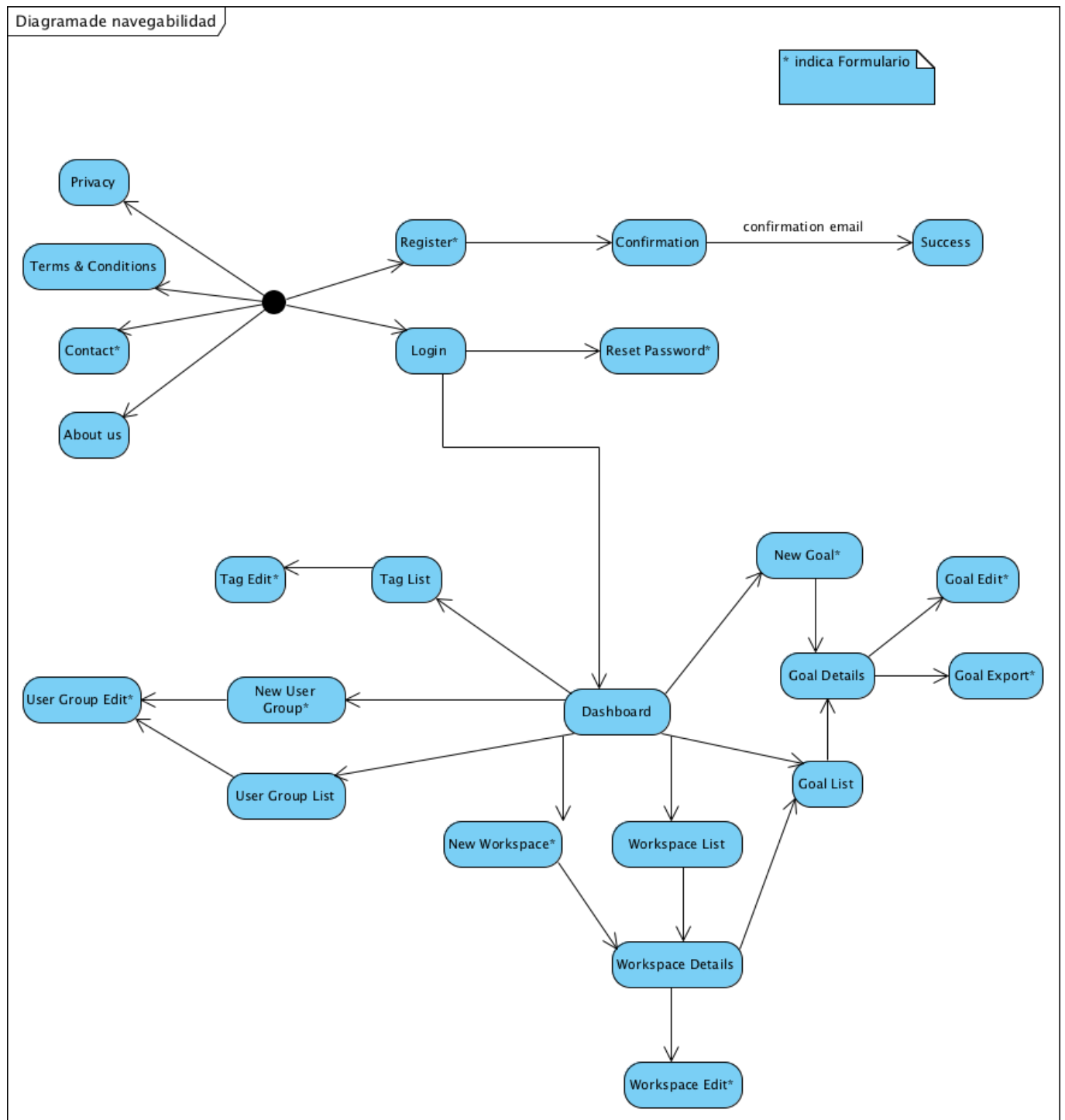


Ilustración 27. Diagrama de navegabilidad

La aplicación dispondrá de una ventana principal desde la que se accederá a todas las funcionalidades de la misma.

### 6.3.2 Pantallas del sitio web

Para determinar las funcionalidades de la aplicación, inicialmente se realizaron prototipos de las distintas vistas por las que estará compuesta la misma, estableciendo cómo serían posteriormente emplazadas cada una de sus características. Estas vistas no son definitivas, por lo que pueden variar en la versión final de la aplicación, pero se tomarán como base al desarrollar la interfaz durante la fase de desarrollo.

#### 6.3.2.1 New Goal

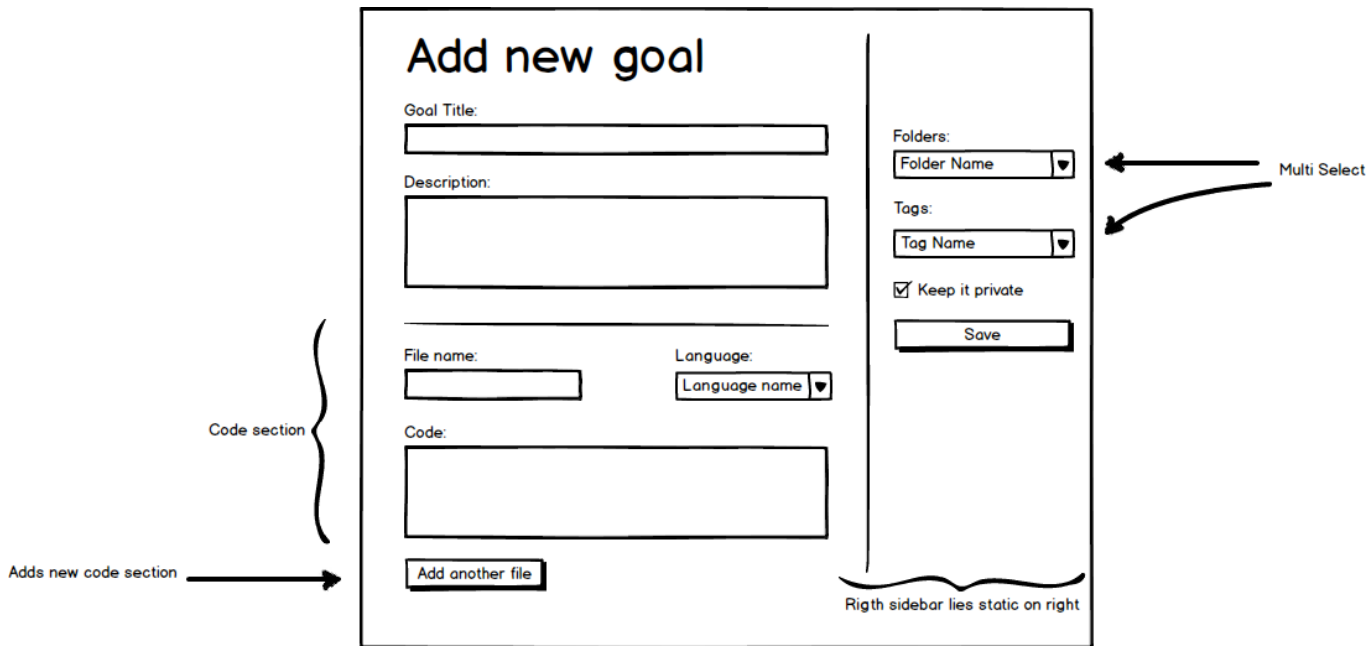


Ilustración 28. Muckup de la pantalla de creación de goal

La pantalla de creación de un nuevo goal debe disponer de un formulario con todos los campos requeridos para la creación del mismo. Debe permitir establecer sus etiquetas, espacios de trabajo así como añadir nuevos snippets a él. Se plantea que la barra lateral derecha permanezca estática para disponer de los elementos de esa zona (principalmente el botón de guardar) siempre a mano.

### 6.3.2.2 New Workspace

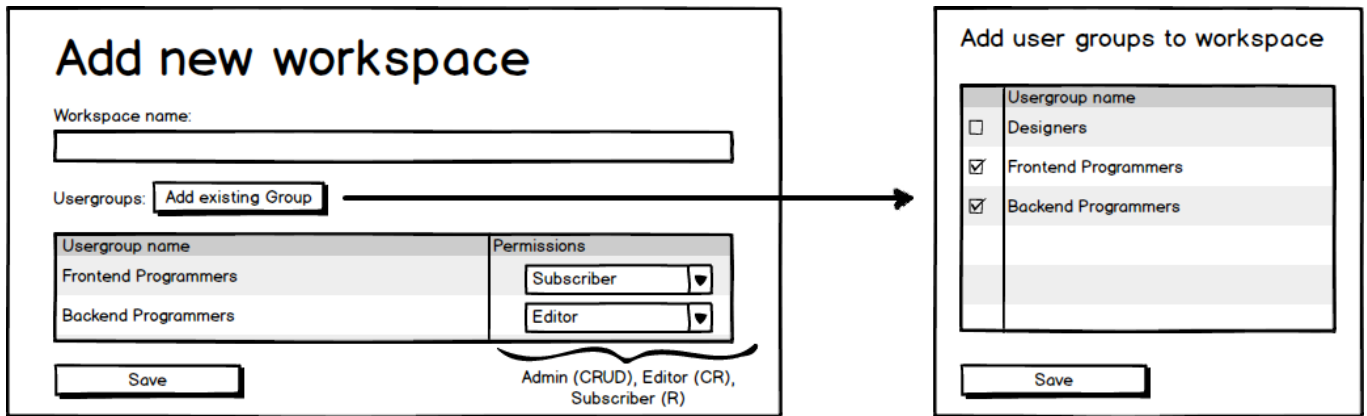


Ilustración 29. Mockup de la pantalla de creación de workspace

Se plantea que la vista de listado de espacios de trabajo (workspaces) incluya información sobre los distintos grupos de usuario que tienen acceso a ella. También se podrán asignar grupos de usuarios desde esta pantalla<sup>7</sup>.

### 6.3.2.3 New User Group

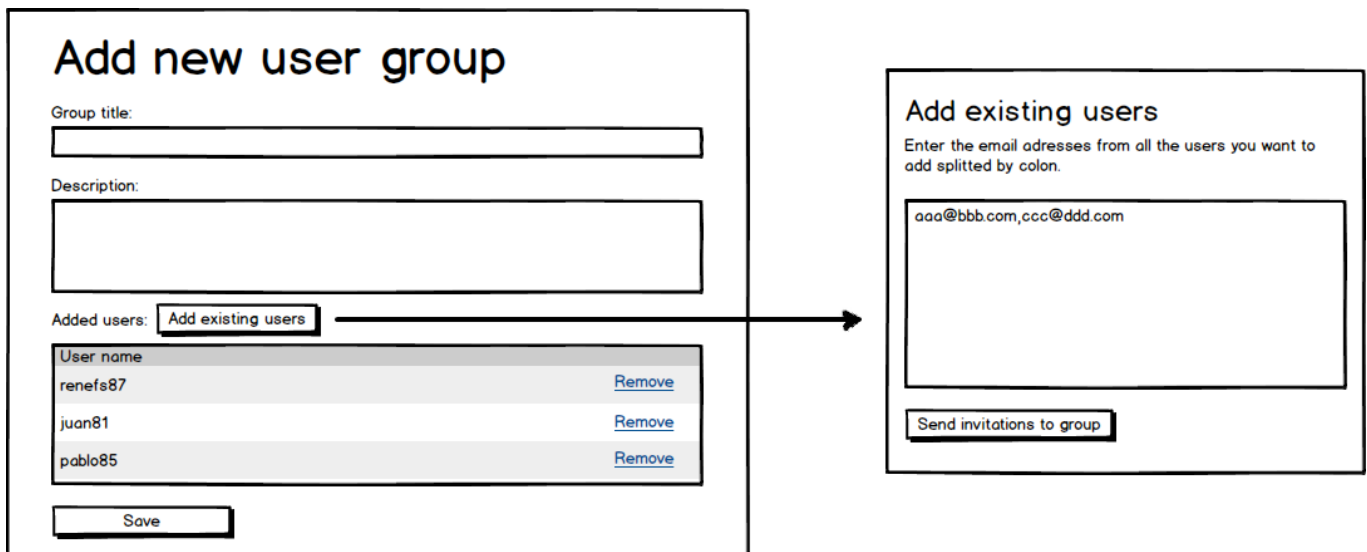


Ilustración 30. Mockup de la pantalla de creación de grupo de usuarios

La vista de detalles de un grupo de usuarios muestra los usuarios que se encuentran en esa lista y permite añadir más pulsando el botón correspondientes.

<sup>7</sup> En la versión final se ha desechado esta opción para mantener una clara diferenciación entre las pantallas de grupos y de workspaces y hacerla menos confusa.

### 6.3.2.4 Goal List

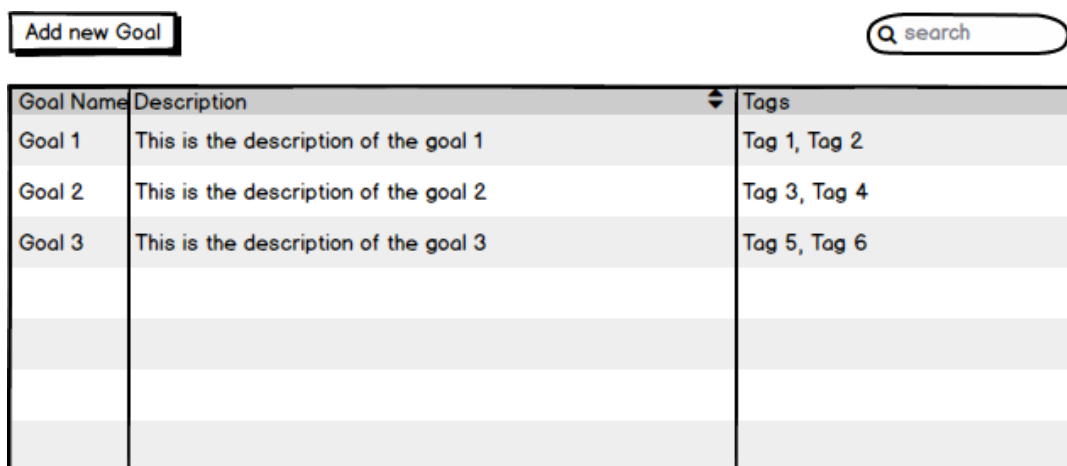


Ilustración 31. Mockup de la pantalla de listado de goals

La pantalla de listado de goals, además del listado de todos los goals del usuario mostrará un campo de búsqueda y un botón para añadir nuevos goals.

### 6.3.2.5 Goal Details

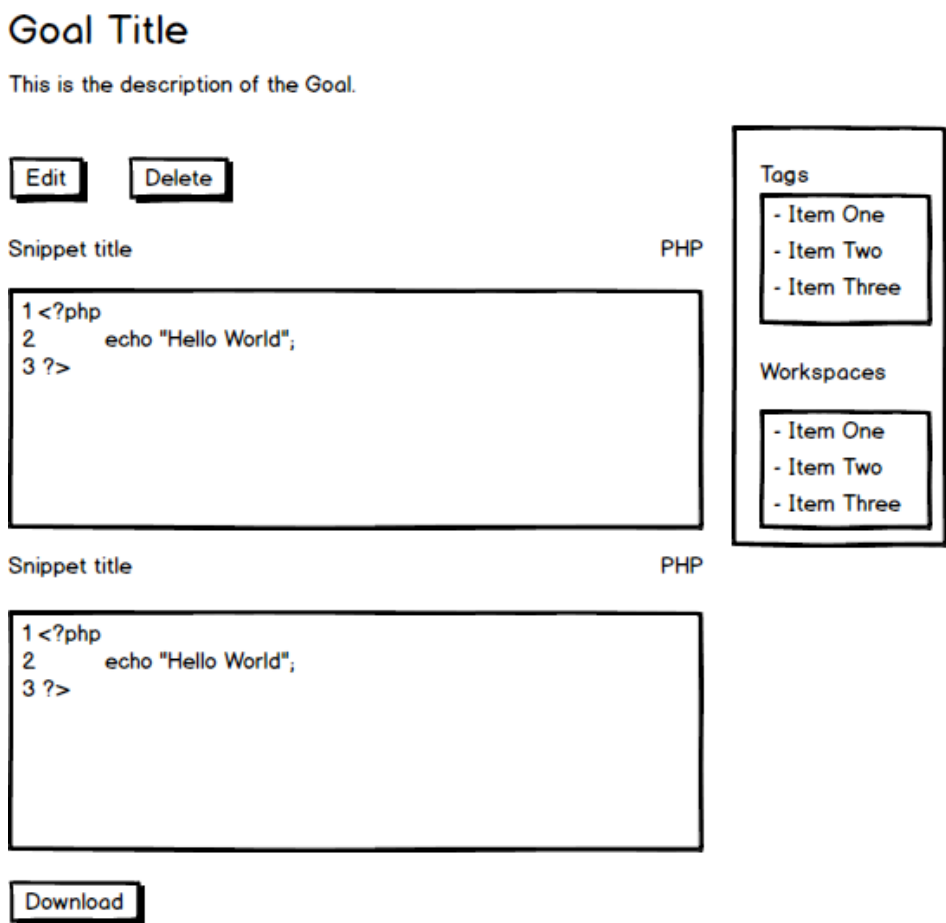


Ilustración 32. Mockup de la pantalla de detalles de goal

La página de detalles del goal mostrará todos los datos añadidos a un goal: sus snippets, lenguajes de programación de los mismos, etiquetas, espacios de trabajo, etc.

### 6.3.2.6 Usergroup Details

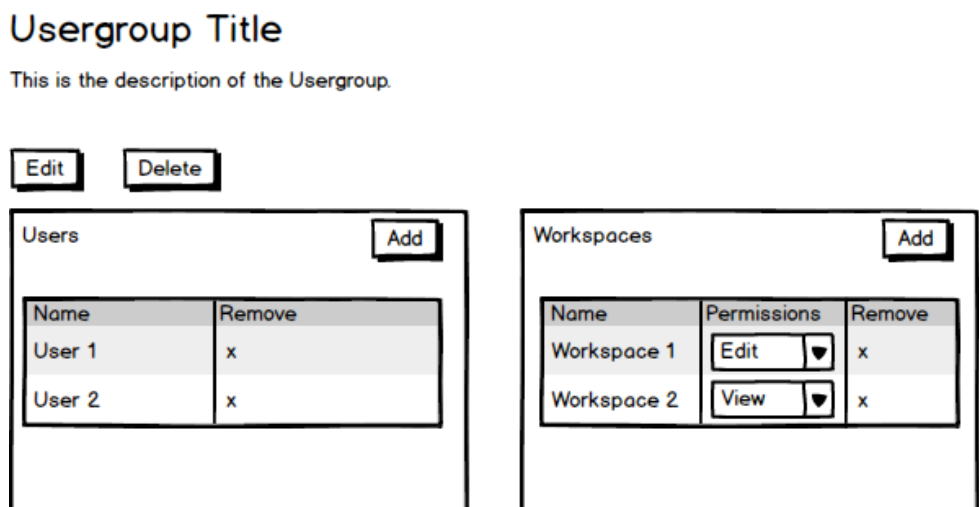


Ilustración 33. Mockup de la pantalla de detalles de grupo de usuarios

La pantalla de detalles de un grupo de usuario muestra los workspaces que forman parte de él y los usuarios que tienen acceso al mismo. También permite gestionarlos.

## 6.4 Diagrama de clases preliminar del análisis

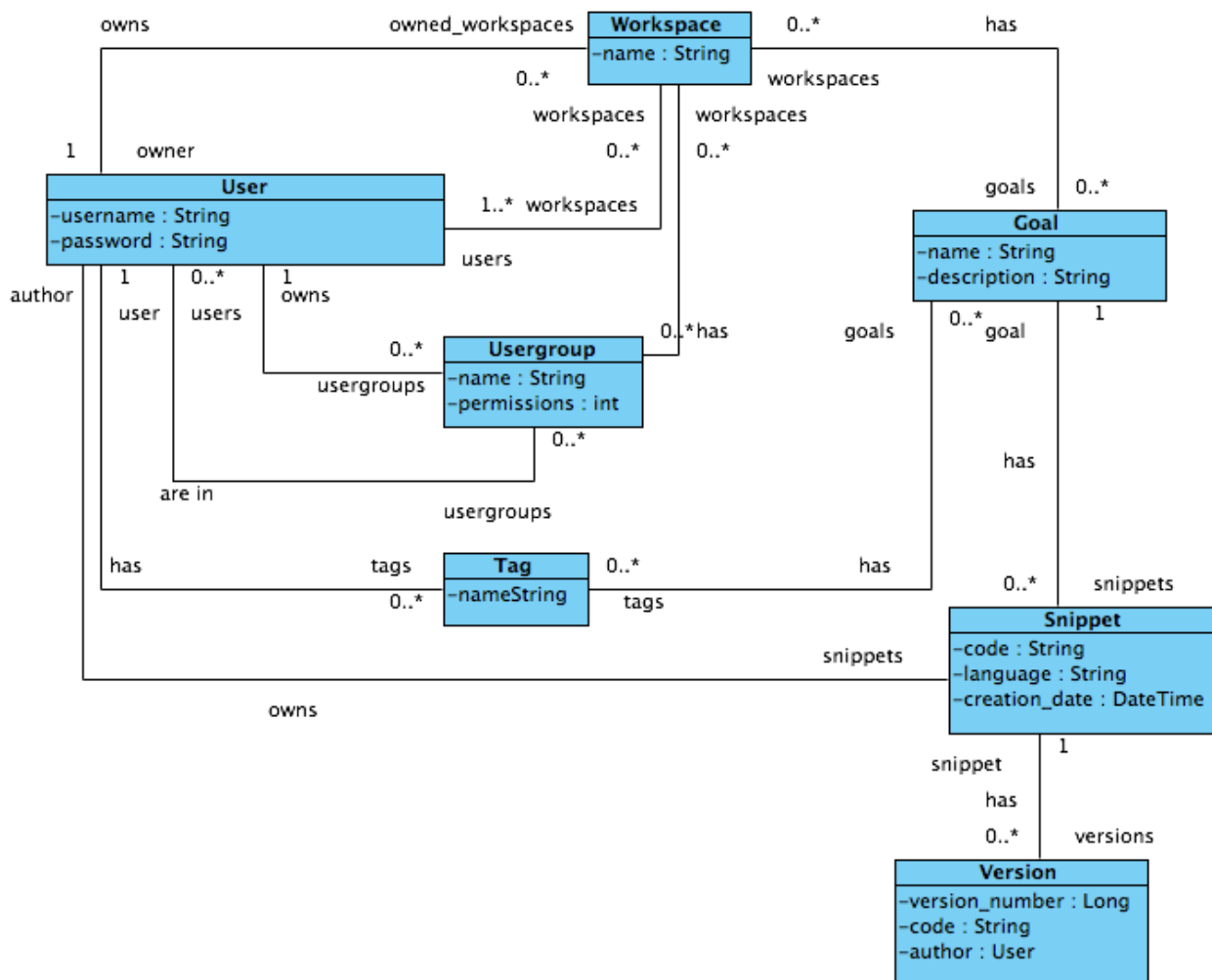


Ilustración 34. Diagrama de clases preliminar

### 6.4.1 Descripción de las clases

#### 6.4.1.1 Clase User

<b>Nombre de la clase</b>	User
<b>Descripción</b>	Representa a un usuario de la aplicación.
<b>Responsabilidades</b>	Gestiona los datos de los usuarios necesarios para acceder y utilizar la aplicación.
<b>Atributos propuestos</b>	<ul style="list-style-type: none"> <li>name: String</li> <li>password: String</li> </ul>

<b>Métodos propuestos</b>	<ul style="list-style-type: none"> <li>-</li> </ul>
---------------------------	---

#### 6.4.1.2 Clase Usergroup

<b>Nombre de la clase</b>	Usergroup
<b>Descripción</b>	Representa a un grupo de usuarios de la aplicación.
<b>Responsabilidades</b>	Es la clase encargada de gestionar a un grupo de usuarios para proporcionarles una serie de permisos y acceso a unos determinados workspaces.
<b>Atributos propuestos</b>	<ul style="list-style-type: none"> <li>name: String</li> </ul>
<b>Métodos propuestos</b>	<ul style="list-style-type: none"> <li>add_goal: Añade un Goal al Workspace.</li> <li>remove_goal: Elimina un Goal de un Workspace.</li> </ul>

#### 6.4.1.3 Clase Workspace

<b>Nombre de la clase</b>	Workspace
<b>Descripción</b>	Contenedor de Goals
<b>Responsabilidades</b>	Es la clase encargada de almacenar Goals de cara a ser compartidos con otros usuarios, de manera que estos se puedan añadir a un Workspace para posteriormente ser asignados a uno (o varios) Usergroups.
<b>Atributos propuestos</b>	<ul style="list-style-type: none"> <li>name: String</li> <li>password: String</li> </ul>
<b>Métodos propuestos</b>	<ul style="list-style-type: none"> <li>-</li> </ul>

#### 6.4.1.4 Clase Tag

<b>Nombre de la clase</b>	Usergroup
<b>Descripción</b>	Clasificador de Goals

<b>Responsabilidades</b>	Esta clase cumple una función similar a Workspace, pero su utilización está más enfocada a la organización de los Goals por parte del usuario para su propio uso.
<b>Atributos propuestos</b>	<ul style="list-style-type: none"> <li>• name: String</li> </ul>
<b>Métodos propuestos</b>	<ul style="list-style-type: none"> <li>• -</li> </ul>

#### 6.4.1.5 Clase Goal

<b>Nombre de la clase</b>	Goal
<b>Descripción</b>	Contenedor de Snippets de código que cumplen una función concreta conjunta.
<b>Responsabilidades</b>	El Goal es el principal núcleo de la aplicación y se utiliza para almacenar distintos Snippets de código.
<b>Atributos propuestos</b>	<ul style="list-style-type: none"> <li>• name: String</li> <li>• description: String</li> </ul>
<b>Métodos propuestos</b>	<ul style="list-style-type: none"> <li>• add_snippet: Añade un Snippet al Goal</li> <li>• remove_snippet: Elimina un Snippet del Goal.</li> <li>• add_tag: Añade una etiqueta al Goal</li> <li>• remove_tag: Elimina una etiqueta del Goal.</li> </ul>

#### 6.4.1.6 Clase Snippet

<b>Nombre de la clase</b>	Snippet
<b>Descripción</b>	Clase contenedora de un fragmento de código
<b>Responsabilidades</b>	Contiene la información de un fragmento de código almacenada en la aplicación, tal como lenguaje de programación, fecha de creación a parte del propio código.
<b>Atributos propuestos</b>	<ul style="list-style-type: none"> <li>• code: String</li> <li>• language: String</li> <li>• creation_date: DateTime</li> </ul>



<b>Métodos propuestos</b>	<ul style="list-style-type: none"> <li>-</li> </ul>
---------------------------	---

#### 6.4.1.7 Clase Version

<b>Nombre de la clase</b>	Version
<b>Descripción</b>	Representa el estado de un Snippet en un determinado momento
<b>Responsabilidades</b>	Cuando se realiza una modificación de un Snippet, se genera automáticamente una nueva versión, que almacenará el estado del Snippet en ese momento permitiendo su restauración o consulta posterior
<b>Atributos propuestos</b>	<ul style="list-style-type: none"> <li>version_number: Long</li> <li>code: String</li> <li>author: User</li> </ul>
<b>Métodos propuestos</b>	<ul style="list-style-type: none"> <li>add_goal: Añade un Goal al Workspace.</li> <li>remove_goal: Elimina un Goal de un Workspace.</li> </ul>

## 7 Diseño del sistema

### 7.1 Arquitectura del sistema

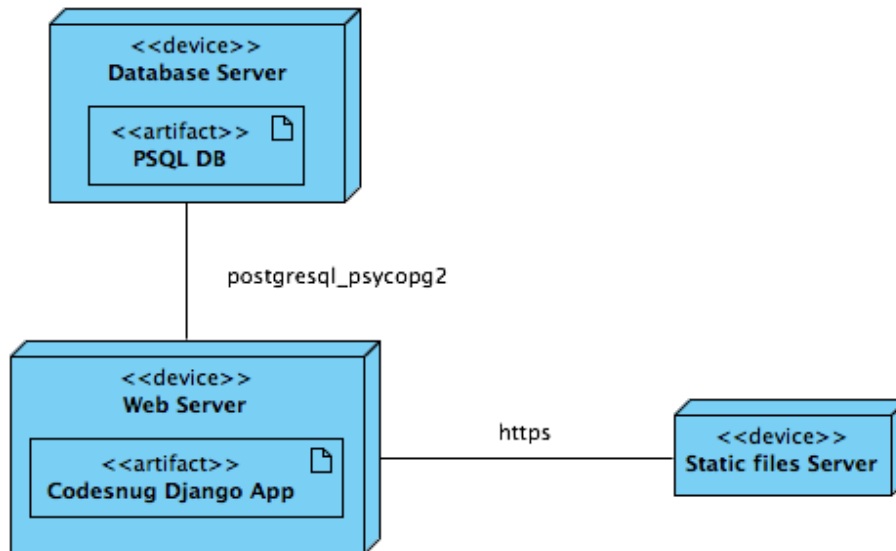


Ilustración 35. Diagrama de despliegue del sistema

El sistema diseñado contará con 3 dispositivos.

- **Servidor de datos estáticos:** Se encargará de servir los archivos estáticos de la aplicación: js, css, imágenes, tipografías, etc. Será accesible a través de la aplicación utilizando HTTPS.
- **Servidor de base de datos:** Alojara la base de datos PostgreSQL de la aplicación. Se accederá a ella desde la aplicación utilizando el driver postgresql\_psycopg2 de Python.
- **Servidor Web de la aplicación:** Será el que aloje la la aplicación web realizada en Django. Se tratará de un servidor Apache2 que ejecutará la aplicación utilizando las especificación WSGI.

### 7.1.1 Estructura de la Aplicación web

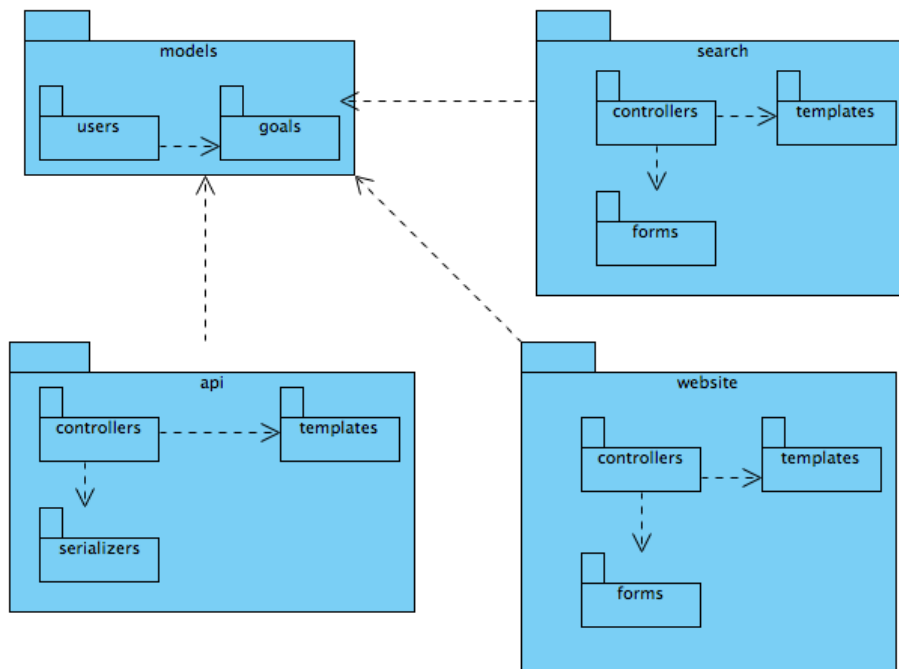


Ilustración 36. Diagrama de módulos de la aplicación

La aplicación web se desarrollará utilizando el framework Django. Este framework basa su estructura en el patrón de diseño MVT (Model-View-Template), similar al MVC (Model-View-Controller). En este patrón, los controladores de las plantillas se encuentran en los archivos de vistas (Por ejemplo views.py), mientras que las plantillas (templates) son archivos html cargados por dichos controladores, que son plantillas en el lenguaje DTL (Django Template Language).

La aplicación estará formada por cuatro módulos diferentes, cada uno de los cuales incluirá (excepto en el caso de los modelos) sus propios controladores y plantillas:

- **models:** Incluye los modelos de datos para el sistema de usuarios y también para los datos almacenados en la aplicación. Ese módulo será utilizado por todos los demás.
- **api:** Se corresponderá con la API REST de la aplicación.
- **search:** Sistema de búsqueda de la aplicación que se utilizará para realizar búsquedas internas de elementos en el sitio web.
- **website:** El sitio web como tal, que incluirá todos los elementos necesarios para que el mismo funcione y que incluirá los manejadores de los formularios.

### 7.1.1.1 models

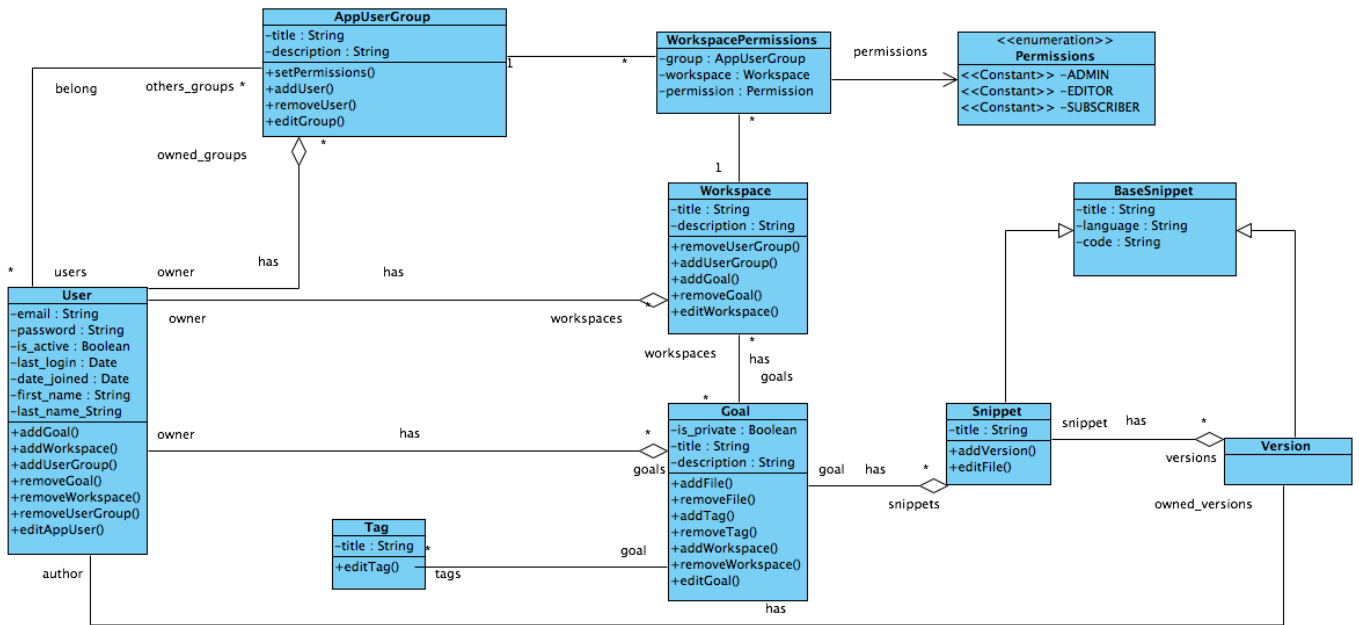


Ilustración 37. Modelos de datos la aplicación

La aplicación estará compuesta por dos tipos de datos principales diferentes: los goals y los usuarios. Puesto que el sistema permitirá el registro de usuarios, su organización en grupos y la modificación de sus datos, estos deberán poder ser gestionados de una manera óptima. Por otro lado, los propios datos que manejará la aplicación estarán incluidos en el submódulo de goals, que incluirá todos los modelos de datos que almacenará la aplicación correspondiente a sus funcionalidades.

### 7.1.1.1.1 goals

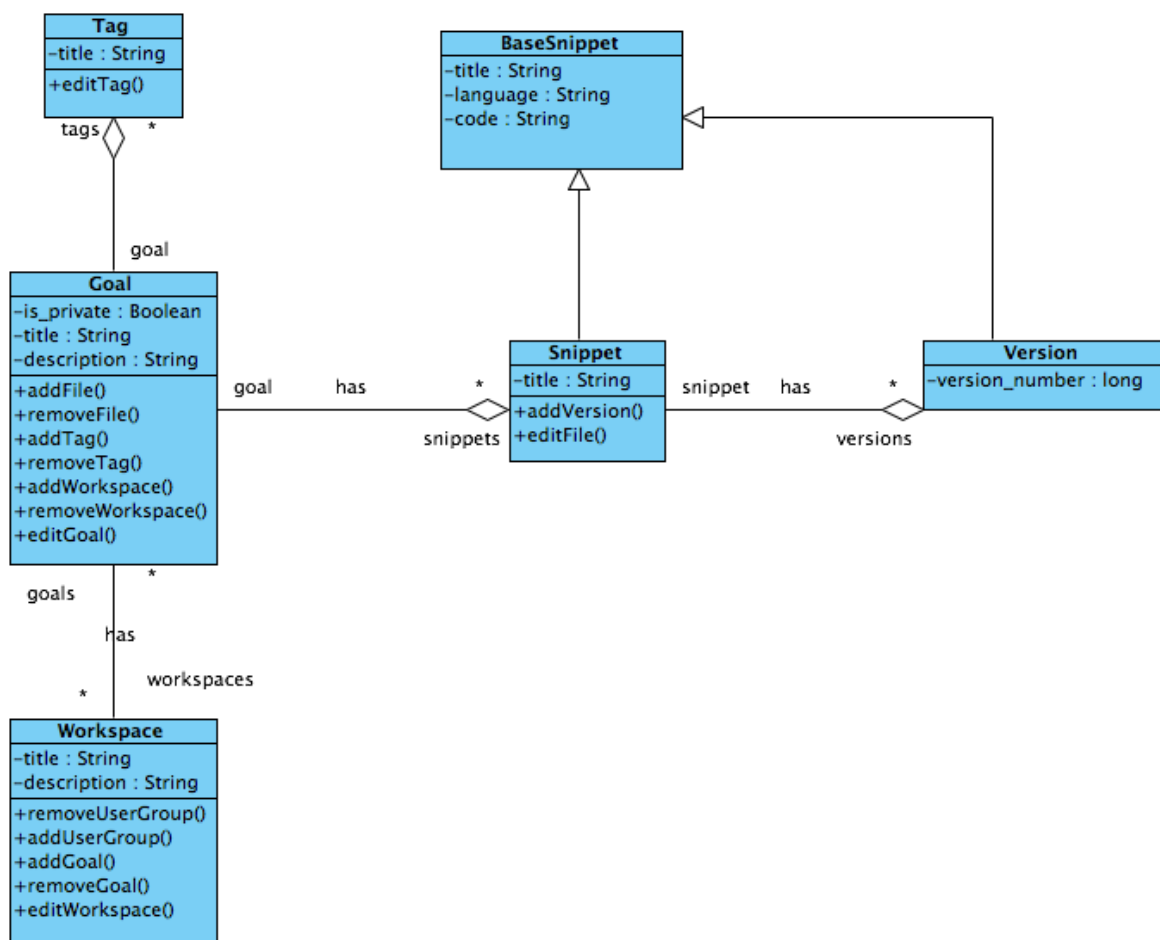


Ilustración 38. Modelos del módulo de goals

Los modelos referentes a los datos funcionales de la aplicación estarán organizados en este submódulo, que estará organizado en las siguientes clases. La mayoría de los atributos hacen referencias a campos de la base de datos de modelos Django (Charfield, TextField, Booleanfield, etc) y a las distintas referencias que ofrecen (ForeignKey, ManyToManyField) para poder disponer de un mapeo automático a la BD utilizando el ORM incluido en Django.

#### 7.1.1.1.1.1 Clase Goal

Nombre	Descripción	Hereda de	
Goal	Dato principal de almacenamiento de la aplicación, que permite agrupar distintos snippets de código en un determinado contexto.	django.db.models.Model	
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
Clase	URL	get_absolute_url	-

Atributos		
Acceso	Tipo o Clase	Nombre
público	ForeignKey(MyUser)	owner
público	CharField	title
público	TextField	description
público	DateTimeField	created_at
público	ManyToManyField(Tag)	tags
público	ManyToManyField(Workspace)	workspaces
público	BooleanField	is_private
público	UUIDField <sup>8</sup>	uuid

### Ejemplo de modelo

```

class Goal(models.Model):
    owner = models.ForeignKey(MyUser, related_name='owned_goals')
    title = models.CharField(max_length=200)
    description = models.TextField(blank=True)
    created_at = models.DateTimeField(auto_now_add=True)
    tags = models.ManyToManyField(Tag, blank=True, related_name='goals')
    workspaces = models.ManyToManyField(Workspace, blank=True, related_name='goals')
    is_private = models.BooleanField(default=True)
    uuid = UUIDField(auto=True)

    class Meta:
        unique_together = (("title", "owner"),)
        ordering = ('created_at',)

    def __unicode__(self):
        return self.uuid.__unicode__()

    def get_absolute_url(self):
        from django.core.urlresolvers import reverse
        url = reverse("goal_detail", args=[str(self.uuid), str(self.owner.username)])
        return url

```

<sup>8</sup> Para la utilización de UUIDField es necesario instalar el módulo correspondiente por medio de PIP. Más información disponible en la web del autor de módulo: <https://github.com/dcramer/django-uuidfield> .

**7.1.1.1.1.2 Clase Tag**

Nombre	Descripción	Hereda de
Tag	Etiqueta que se utilizará para clasificar Goals.	django.db.models.Model
Atributos		
Acceso	Tipo o Clase	Nombre
público	ForeignKey(MyUser)	owner
público	CharField	title
público	UUIDField	uuid

**7.1.1.1.1.3 Clase Workspace**

Nombre	Descripción	Hereda de
<b>Workspace</b>	Utilizada para almacenar y organizar varios Goals y permitir que estos sean compartidos con otros usuarios.	django.db.models.Model
Atributos		
Acceso	Tipo o Clase	Nombre
público	ForeignKey(MyUser)	owner
público	CharField	title
público	TextField	description
público	DateTimeField	created_at
público	UUIDField	uuid

**7.1.1.1.1.4 Clase BaseSnippet**

Nombre	Descripción	Hereda de
<b>BaseSnippet</b>	Clase padre que contiene los atributos básicos de un Snippet.	django.db.models.Model
Atributos		
Acceso	Tipo o Clase	Nombre
público	CharField	language
público	IntegerField	version_number
público	TextField	text
público	DateTimeField	created_at
público	UUIDField	uuid



**7.1.1.1.1.5 Clase Snippet**

Nombre	Descripción	Hereda de
Snippet	Clase que representa un fragmento de código escrito en un determinado lenguaje de programación.	BaseSnippet
Atributos		
Acceso	Tipo o Clase	Nombre
público	ForeignKey(MyUser)	author
público	ForeignKey(Goal)	goal

**7.1.1.1.1.6 Clase Version**

Nombre	Descripción	Hereda de
Version	Tipo de BaseSnippet que representa una versión de un Snippet en un momento determinado.	BaseSnippet
Atributos		
Acceso	Tipo o Clase	Nombre
público	ForeignKey(MyUser)	author
público	ForeignKey(Snippet)	snippet

## 7.1.1.1.2 users

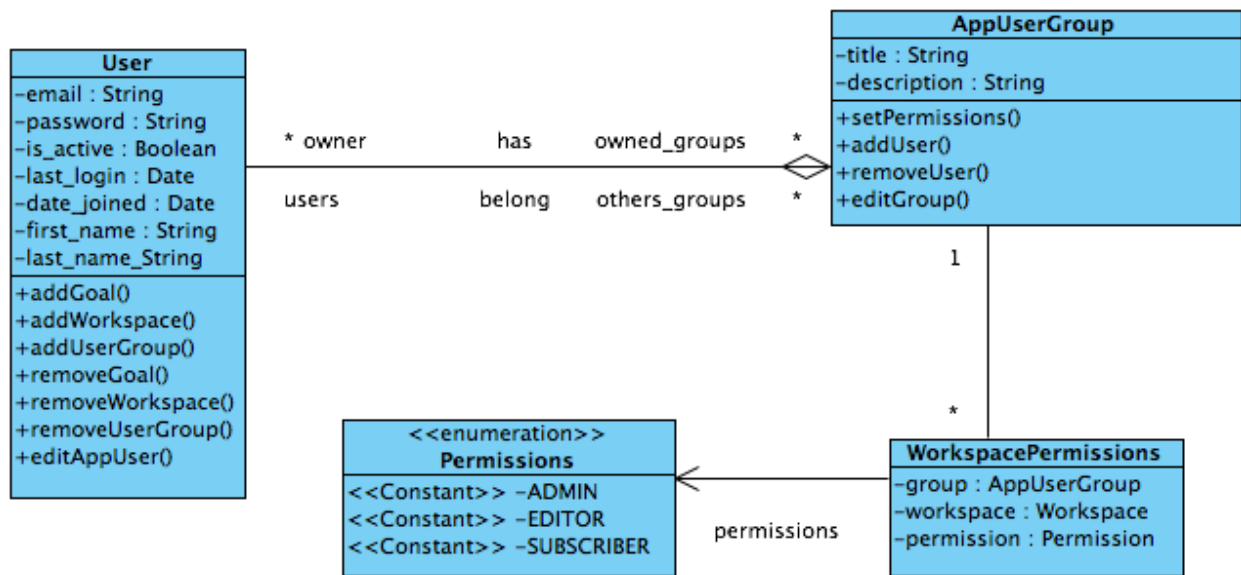


Ilustración 39. Modelos del módulo users

Los modelos referentes a la gestión de usuarios de la aplicación, incluyendo el propio modelo para los usuarios y también los grupos de los mismos, utilizados para compartir Goals entre ellos. Es necesaria también la clase **WorkspacePermissions** para relacionar los grupos de usuarios (**AppUserGroup**) con los Workspaces y con los permisos asignados a cada uno de ellos, representado por la enumeración **Permissions**. Existen 3 permisos diferentes:

- **Admin**: Administrador y creador de un Goal determinado.
- **Editor**: Usuario con capacidad de edición de un Goal
- **Subscriber**: Usuario con capacidad de visualizar un Goal y su contenido, pero no editarlo.

## 7.1.1.1.2.1 Clase User (MyUser)

Nombre	Descripción		Hereda de
<b>MyUser</b>	Clase que representa a un usuario de la aplicación.		django.contrib.auth.models.AbstractUser  django.contrib.auth.models.PermissionsMixin
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
<b>Instancia</b>	URL	get_absolute_url	-
<b>Instancia</b>	String	get_full_name	
<b>Instancia</b>	String	get_short_name	
<b>Instancia</b>	void	email_user	<ul style="list-style-type: none"> <li>• subject: Asunto del email</li> <li>• message: Texto del email</li> <li>• from_email: Emisor del email</li> </ul>
Atributos			
Acceso	Tipo o Clase	Nombre	
<b>público</b>	ForeignKey(MyUser)	owner	
<b>público</b>	CharField	title	
<b>público</b>	TextField	description	
<b>público</b>	DateTimeField	created_at	
<b>público</b>	ManyToManyField(Tag)	tags	
<b>público</b>	ManyToManyField(Workspace)	workspaces	
<b>público</b>	BooleanField	is_private	
<b>público</b>	UUIDField	uuid	

## 7.1.1.1.2.2 Clase AppUserGroup (Usergroup)

Nombre	Descripción		Hereda de
<b>Usergroup</b>	Clase que representa un grupo de usuarios de la aplicación. Es creado por los propios usuarios de la misma.		django.contrib.auth.models.Model
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
<b>Instancia</b>	void	save	<ul style="list-style-type: none"> <li>- *args: Dict</li> <li>- **kwargs: Dic</li> </ul>
Atributos			
Acceso	Tipo o Clase	Nombre	
<b>público</b>	ForeignKey(MyUser)	owner	
<b>público</b>	CharField	title	
<b>público</b>	TextField	description	
<b>público</b>	DateTimeField	created_at	
<b>público</b>	ManyToManyField(MyUser)	users	
<b>público</b>	UUIDField	uuid	

### 7.1.1.1.2.3 Clase WorkspacePermissions

Esta clase será la encargada de relacionar los módulos goals y users y será necesaria puesto que identificará qué permisos tiene un determinado grupo de usuarios (Usergroup) sobre un Workspace.

Los permisos que estarán disponibles para los grupos serán los siguientes:

- **Admin** (0): Administrador o creador de un determinado elemento.
- **Editor** (1): Usuario con habilidad de modificar un elemento.
- **Subscriber** (2): Usuario con habilidad de visualizar un elemento, pero no editarlo.

Nombre	Descripción	Hereda de
<b>WorkspacePermissions</b>	Representa la relación entre Workspaces y grupos de usuarios	django.contrib.auth.models.Model
Atributos		
Acceso	Tipo o Clase	Nombre
público	ForeignKey(Usergroup)	group
público	ForeignKey(Workspace)	workspace
público	IntegerField(WORKSPACE_PERMISSIONS)	permission
público	UUIDField	uuid

### 7.1.1.2 search

#### 7.1.1.2.1 controllers

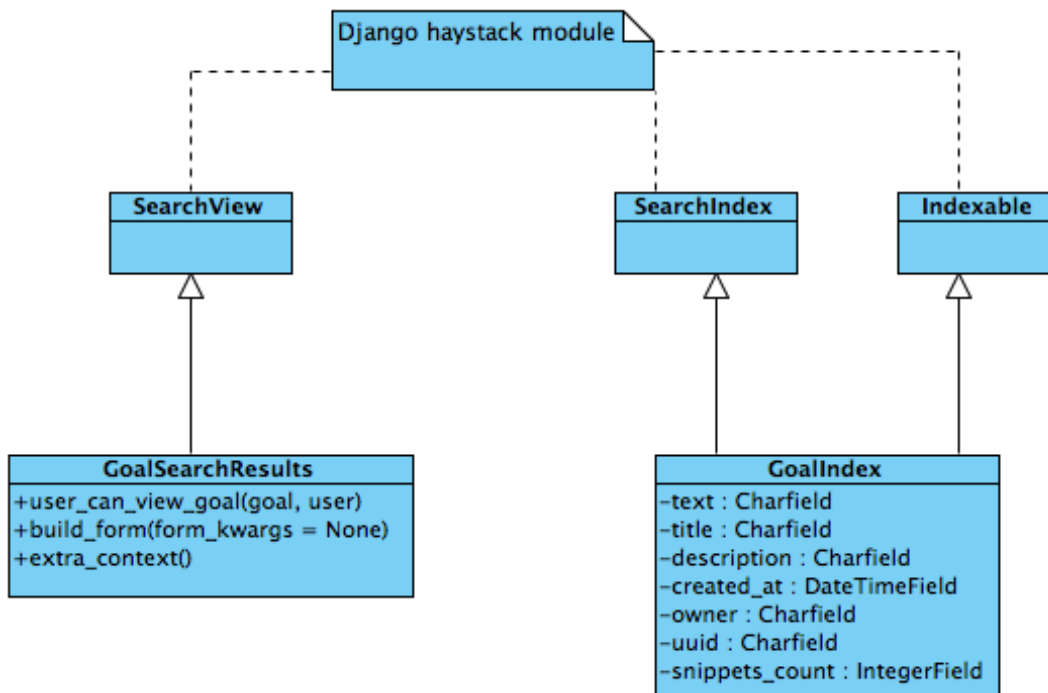


Ilustración 40. Controladores del módulo search

El sistema de búsqueda interno de la aplicación utilizará Django Haystack<sup>9</sup> que permite integrar distintos motores de búsquedas en el sitio web (como Solr, Woosh o Elasticsearch) actuando como un una interfaz entre la aplicación y estos motores, permitiendo que en caso de decidir cambiar a otro motor el proceso sea trivial.

En este caso, se utilizará Woosh<sup>10</sup> para una puesta en marcha inicial puesto que los recursos disponibles son limitados y está implementado en Python, por lo que la integración con el resto del sistema será considerablemente más sencilla y no habrá que disponer de otro servidor para manejar las búsquedas.

El sistema permitirá indexar los Goals cada vez que son creados o actualizados, para lo que se necesitará un índice de búsqueda que actúe como modelo (GoalIndex) y también un controlador para la vista de resultados de búsqueda que gestionará lo que mostrará esa página y también a qué datos tiene acceso el usuario actual (GoalSearchResults).

#### 7.1.1.2.1.1 Clase GoalIndex

Nombre	Descripción	Hereda de
GoalIndex	Representa el modelo de Goal para un resultado de búsqueda.	haystack.indexes.SearchIndex haystack.indexes.Indexable
Atributos		

<sup>9</sup> Página oficial de Django Haystack: <http://haystacksearch.org/>

<sup>10</sup> Página oficial de Woosh: <https://bitbucket.org/mchaput/whoosh/wiki/Home>

Acceso	Tipo o Clase	Nombre
público	ForeignKey(MyUser)	owner
público	CharField	title
público	TextField	description
público	DateTimeField	created_at
público	CharField()	text
público	UUIDField	uuid
público	IntegerField	snippets_count

#### 7.1.1.2.1.2 Clase GoalSearchResults

Nombre	Descripción	Hereda de	
GoalSearchResults	Clase controladora de la página de resultados de búsqueda.	haystack.views.SearchView	
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
Instancia	Boolean	user_can_view_goal	- goal: Goal - user: MyUser
Instancia	SearchForm	build_form	- form_kwargs: Dict
Instancia	Dict	extra_content	

## 7.1.1.2.2 forms

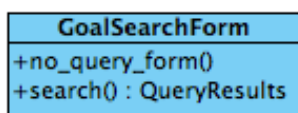


Ilustración 41. Formulario del módulo search

## 7.1.1.2.2.1 Clase GoalSearchForm

Nombre	Descripción		Hereda de
GoalSearchForm	Formulario de búsqueda de Goals en el sitio.		haystack.forms.SearchForm
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
Instancia	QueryResults	no_query_found	
Instancia	QueryResults	search	



### 7.1.1.3 api

La aplicación dispondrá de una API REST para acceder a la información almacenada por sus usuarios de manera que estos puedan consultarla desde sus propias aplicaciones. Para la implementación de esta API se utilizará el módulo Django REST Framework puesto que permite la creación de una API de una manera relativamente sencilla y con muchas características útiles, como diversos sistemas de seguridad o acceso a la API desde un navegador web.

#### 7.1.1.3.1 serializers

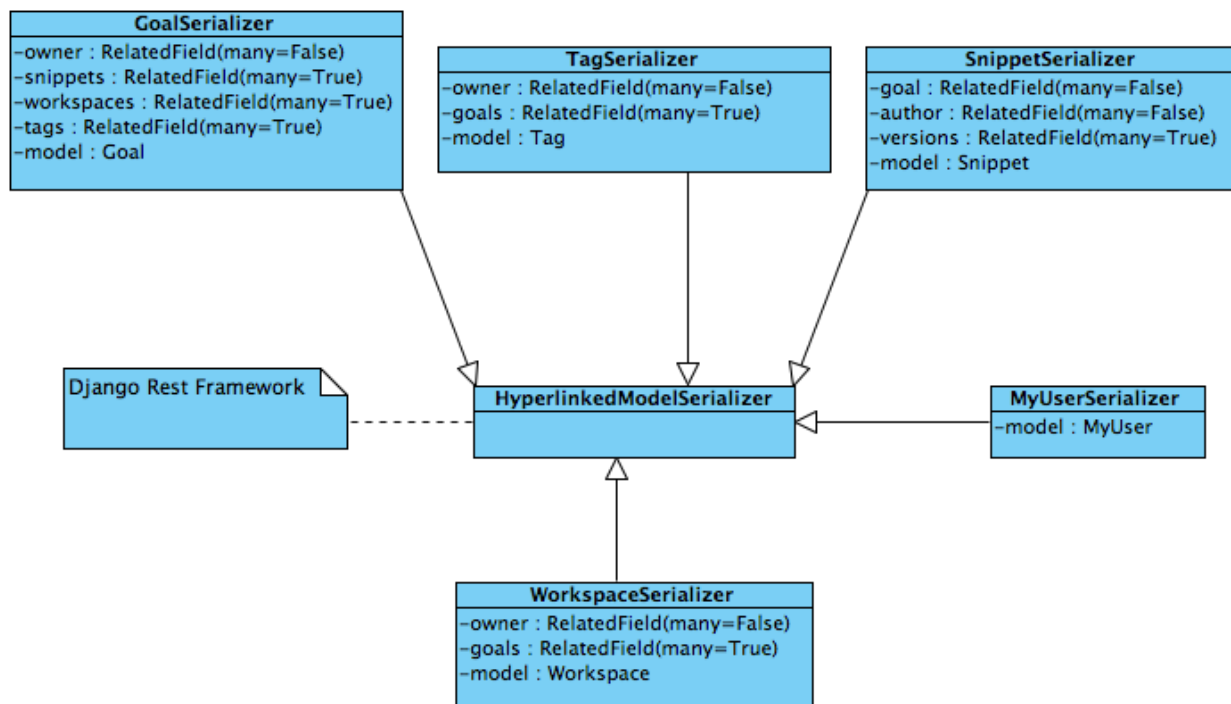


Ilustración 42. Serializadores de la api

Para determinar cuales de los objetos son serializables por la API se deberá crear un modelo de objetos serializables en el que se establezcan las relaciones entre los objetos. Si una relación no aparece, no se mostrará el objeto a partir del acceso a otro. Los campos mostrados para cada objeto vendrán determinados por el atributo “fields”, que será un diccionario que hará referencia a los campos del objeto.

#### 7.1.1.3.1.1 Clase GoalSerializer

Nombre	Descripción	Hereda de
GoalSerializer	Clase serializadora de Goals para la API.	rest_framework.serializers.HyperlinkedModelSerializer
Atributos		
Acceso	Tipo o Clase	Nombre
público	RelatedField	owner

público	RelatedField	snippets
público	RelatedField	workspaces
público	RelatedField	tags
público	Dict	fields

#### 7.1.1.3.1.2 Clase SnippetSerializer

Nombre	Descripción	Hereda de
SnippetSerializer	Clase serializadora de Snippets para la API.	rest_framework.serializers.HyperlinkedModelSerializer
Atributos		
Acceso	Tipo o Clase	Nombre
público	RelatedField	goal
público	RelatedField	author
público	RelatedField	versions
público	Dict	fields

#### 7.1.1.3.1.3 Clase WorkspaceSerializer

Nombre	Descripción	Hereda de
WorkspaceSerializer	Clase serializadora de Workspaces para la API.	rest_framework.serializers.HyperlinkedModelSerializer
Atributos		
Acceso	Tipo o Clase	Nombre
público	RelatedField	owner
público	RelatedField	goals
público	Dict	fields

**7.1.1.3.1.4 Clase TagSerializer**

Nombre	Descripción	Hereda de
TagSerializer	Clase serializadora de Tags para la API.	rest_framework.serializers.HyperlinkedModelSerializer
Atributos		
Acceso	Tipo o Clase	Nombre
público	RelatedField	owner
público	RelatedField	goals
público	Dict	fields

**7.1.1.3.1.5 Clase MyUserSerializer**

Nombre	Descripción	Hereda de
MyUserSerializer	Clase serializadora de usuarios para la API.	rest_framework.serializers.HyperlinkedModelSerializer
Atributos		
Acceso	Tipo o Clase	Nombre
público	RelatedField	owner
público	RelatedField	snippets
público	RelatedField	workspaces
público	Dict	fields

### 7.1.1.4 controllers

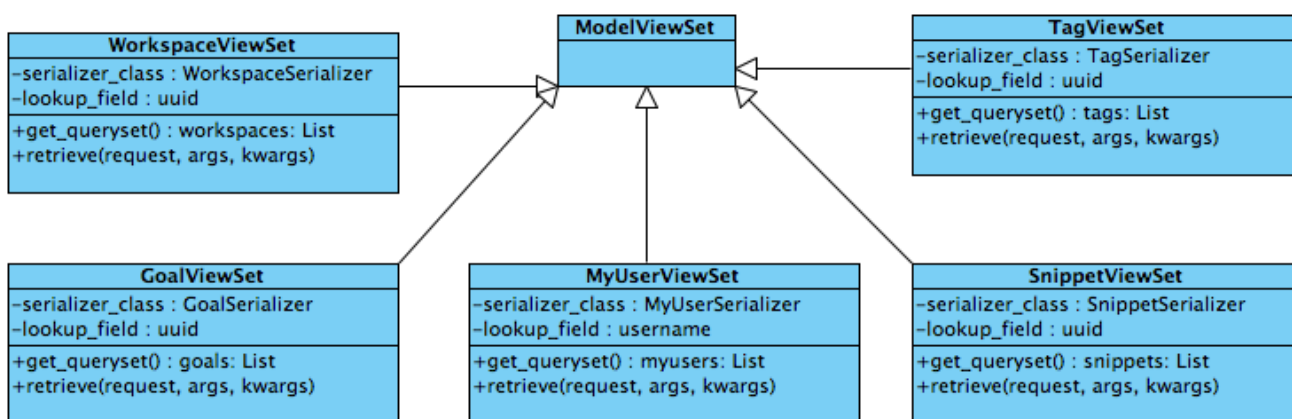


Ilustración 43. Controladores de la api

Estas clases controlarán la información que devolverá cada una de las URL de la API, habiendo una para cada tipo de modelo. Puesto que el funcionamiento de todas ellas es similar, únicamente se detallará una.

#### 7.1.1.4.1.1 Clase WorkspaceViewSet

Nombre	Descripción	Hereda de	
WorkspaceViewSet	Clase que devuelve la información relativa a los Workspaces ya serializada para ser consumida por la API.	rest_framework.viewsets.ModelViewSet	
Atributos			
Acceso	Tipo o Clase	Nombre	
público	Class	serializer_class	
público	String	lookup_field	
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
Instancia	List	get_queryset	- goal: Goal - user: MyUser
Instancia	Response	retrieve	- request: Dict - args: Dict - kwargs: Dict

### 7.1.1.5 *website*

Este paquete incluye la aplicación web en sí misma, incluyendo los controladores de las vistas y de sus distintos formularios.

#### 7.1.1.5.1 *controllers*

Debido a que la aplicación dispone de muchas páginas diferentes, los controladores estarán agrupados principalmente en goals y users, al igual que en el resto del sistema. Para que los archivos de controladores no ocupen demasiado y para mantenerlo todo más organizado, aquellos archivos que tengan un número considerable de líneas serán también divididos, tal y como están los siguientes archivos:

- **goals:**
  - **goals:** Controladores de las vistas que gestionan Goals.
  - **workspaces:** Controladores de las vistas que gestionan Workspaces.
  - **tags:** Controladores de las vistas que gestionan Tags.
  - **snippets:** Controladores de las vistas que gestionan Snippets.
  - **download:** Controladores de las vistas que gestionan las descargas de datos.
- **users:** Al disponer de pocos controladores, se establecerán en un mismo archivo.

#### 7.1.1.5.1.1 *goals goals*

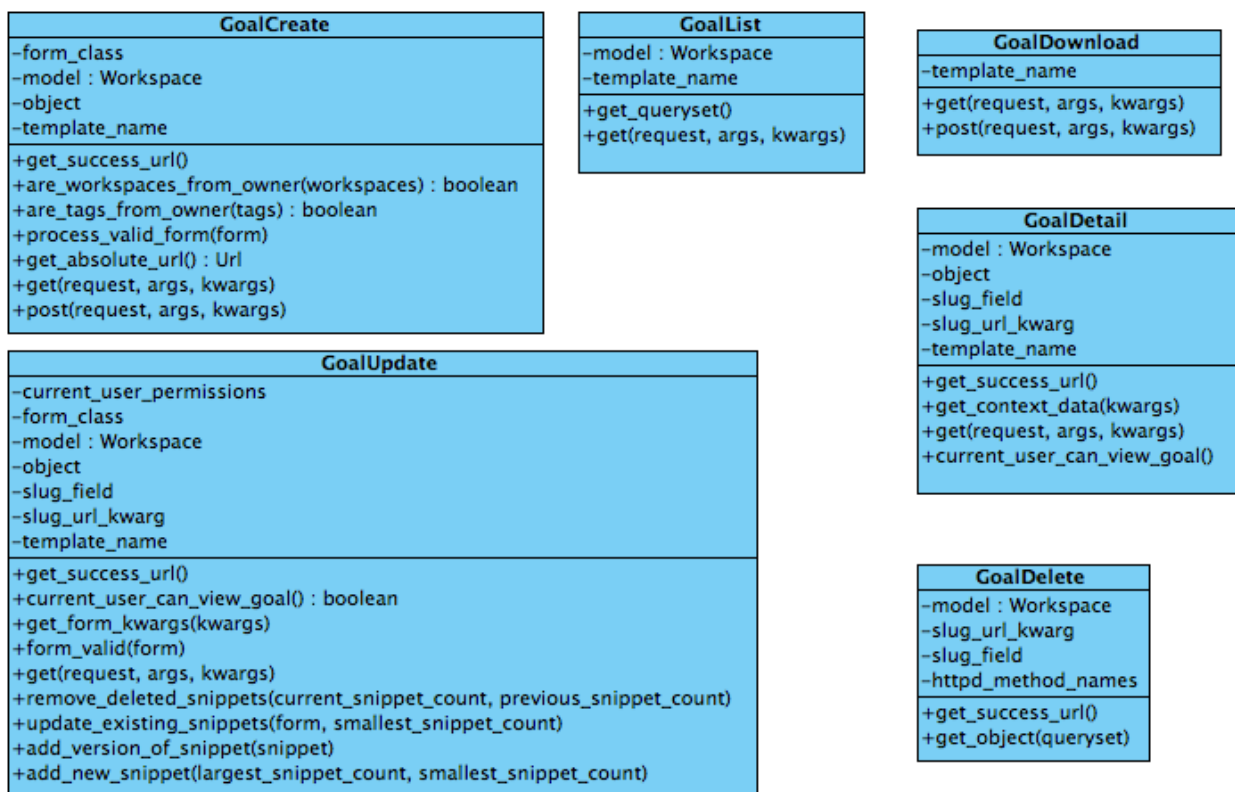


Ilustración 44. Controladores de las vistas de goals del sitio web

En este módulo se incluyen las vistas encargadas de gestionar y de mostrar la información relativa a los Goals.

## 7.1.1.5.1.1.1 Clase GoalCreate

Nombre	Descripción	Hereda de	
<b>GoalCreate</b>	Controlador de la vista de creación de un Goal.	braces.views.LoginRequiredMixin django.views.generic.CreateView	
Atributos			
Acceso	Tipo o Clase	Nombre	
público	Class	model	
público	String	form_class	
	String	template_name	
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
Instancia	URL	get_success_url	
Instancia	Boolean	are_workspaces_from_owner	- workspaces: List
Instancia	Boolean	are_tags_from_owner	- tags: List
Instancia	Goal	process_valid_form	- form: Form
Instancia	URL	get_absolute_url	
Instancia	URL	get	- args: dict - kwargs: dict
Instancia	URL	post	- args: dict - kwargs: dict

## 7.1.1.5.1.1.2 Clase GoalUpdate

Nombre	Descripción	Hereda de	
GoalUpdate	Controlador de la vista de actualización de un Goal.	braces.views.LoginRequiredMixin django.views.generic.UpdateView	
Atributos			
Acceso	Tipo o Clase	Nombre	
público	Class	model	
público	String	form_class	
	String	template_name	
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
Instancia	URL	get_success_url	
Instancia	Boolean	current_user_can_view_goal	
Instancia	dict	get_form_kwargs	- kwargs: dict
Instancia	HttpResponseRedirect	form_valid	- form: Form
Instancia	URL	get_absolute_url	
Instancia	URL	get	- args: dict - kwargs: dict
Instancia	URL	post	- args: dict - kwargs: dict
Instancia	int, int	remove_deleted_snippets	- current_snippet_count: int - previous_snippet_count: int
Instancia	void	update_existing_snippets	- form: Form

			- smallest_snippet_count: int
<b>Instancia</b>	void	add_version_of_snippet	- snippet: Snippet
<b>Instancia</b>	void	add_new_snippets	- form: Form - largest_snippet_count: int - smallest_snippet_count: int

## 7.1.1.5.1.1.3 Clase GoalList

Nombre	Descripción		Hereda de
<b>GoalList</b>	Controlador de vista de listado de Goals.		braces.views.LoginRequiredMixin django.views.generic.ListView
Atributos			
Acceso	Tipo o Clase	Nombre	
<b>público</b>	Class	model	
<b>público</b>	String	template_name	
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
<b>Instancia</b>	URL	get_queryset	
<b>Instancia</b>	URL	get	- args: dict - kwargs: dict

## 7.1.1.5.1.1.4 Clase GoalDelete

Nombre	Descripción		Hereda de
<b>GoalDelete</b>	Controlador de vista de borrado de Goals.		braces.views.LoginRequiredMixin django.views.generic.DeleteView
Atributos			



Acceso	Tipo o Clase	Nombre	
público	Class	model	
público	String	slug_field	
	String	slug_url_kwarg	
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
Instancia	URL	get_success_url	
Instancia	Goal	get_object	<ul style="list-style-type: none"> <li>queryset: dict</li> </ul>

## 7.1.1.5.1.1.5 Clase GoalDetail

Nombre	Descripción	Hereda de	
GoalDetail	Clase controlador de la vista detallada de Goal.	braces.views.LoginRequiredMixin django.views.generic.DetailView	
Atributos			
Acceso	Tipo o Clase	Nombre	
público	Class	model	
público	int	current_user_permissions	
público	String	slug_field	
público	String	slug_url_kwarg	
público	String	template_name	
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
Instancia	URL	get_success_url	

<b>Instancia</b>	Context	get_context_data	- kwargs: dict
<b>Instancia</b>	URL	get	- args: dict - kwargs: dict
<b>Instancia</b>	Boolean	current_user_can_view_goal	

## 7.1.1.5.1.1.6 Clase GoalDownload

Nombre	Descripción	Hereda de	
<b>GoalDownload</b>	Controlador de la vista de descarga de Goal.	braces.views.LoginRequiredMixin django.views.generic.TemplateView	
Atributos			
Acceso	Tipo o Clase	Nombre	
	String	template_name	
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
<b>Instancia</b>	URL	get	- args: dict - kwargs: dict
<b>Instancia</b>	URL	post	- args: dict - kwargs: dict

### 7.1.1.5.1.2 goals workspaces

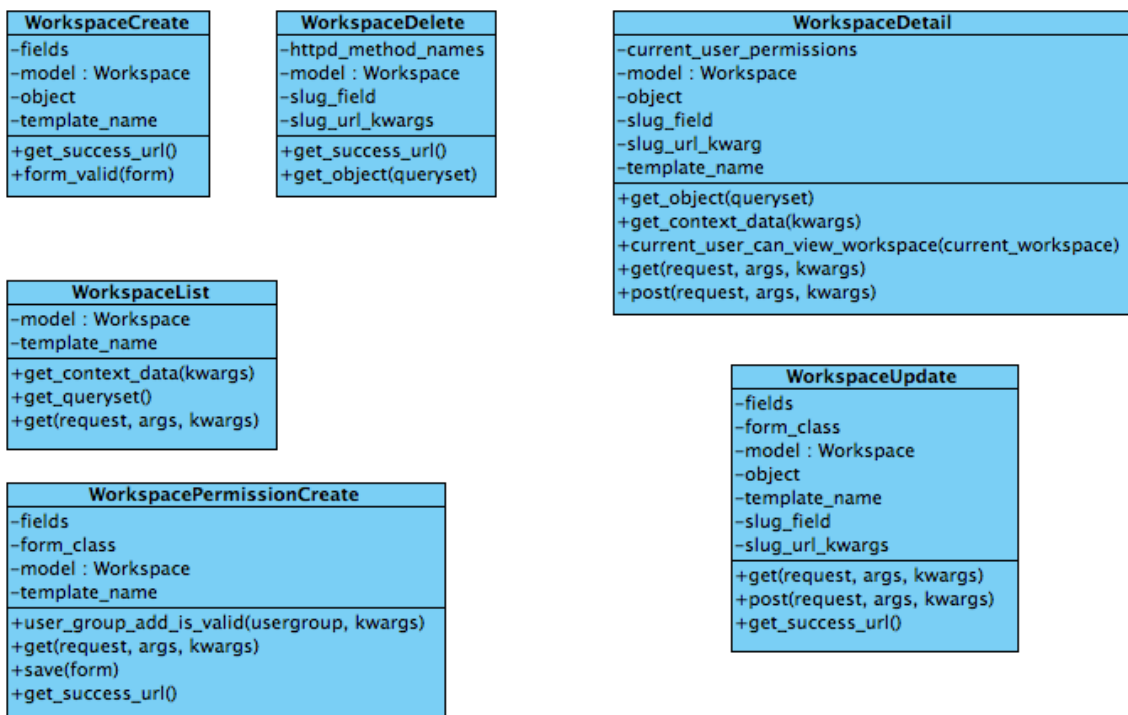


Ilustración 45. Controladores de las vistas de workspaces del sitio web

En este módulo se incluyen las vistas encargadas de gestionar y de mostrar la información relativa a los Workspaces.

#### 7.1.1.5.1.2.1 Clase WorkspaceDetail

Nombre	Descripción	Hereda de
<b>WorkspaceDetail</b>	Clase controladora que gestiona la vista de detalle de un Workspace.	braces.views.LoginRequiredMixin django.views.generic.DetailView
Atributos		
Acceso	Tipo o Clase	Nombre
público	Workspace	model
público	String	slug_field
público	Workspace	object
público	String	slug_url_kwarg
público	String	template_name

público	Integer	Current_user_permissions	
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
Instancia	URL	get	<ul style="list-style-type: none"> <li>request: Request</li> <li>args: dict</li> <li>kwargs: dict</li> </ul>
Instancia	URL	post	<ul style="list-style-type: none"> <li>request: Request</li> <li>args: dict</li> <li>kwargs: dict</li> </ul>
Instancia	Workspace	get_object	<ul style="list-style-type: none"> <li>queryset: dict</li> </ul>
Instancia	dict	get_context_data	<ul style="list-style-type: none"> <li>kwargs: dict</li> </ul>
Instancia	Boolean	current_user_can_view_workspace	<ul style="list-style-type: none"> <li>current_workspace: Workspace</li> </ul>

## 7.1.1.5.1.2.2 Clase WorkspaceCreate

Nombre	Descripción	Hereda de
WorkspaceCreate	Clase controladora que gestiona la vista de creación de un Workspace.	braces.views.LoginRequiredMixin django.views.generic.CreateView
Atributos		
Acceso	Tipo o Clase	Nombre
público	List	fields
público	Workspace	model
público	Workspace	object
público	String	template_name

Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
Instancia	URL	get_success_url	
Instancia	Form	form_valid	<ul style="list-style-type: none"> <li>Form: Form</li> </ul>

7.1.1.5.1.2.3 Clase *WorkspaceDelete*

Nombre	Descripción		Hereda de
<b>WorkspaceDelete</b>	Clase controladora que gestiona la vista de borrado de un Workspace		braces.views.LoginRequiredMixin django.views.generic.DeleteView
Atributos			
Acceso	Tipo o Clase	Nombre	
público	Workspace	model	
público	String	slug_field	
público	String	slug_url_kwarg	
público	List	http_method_names	
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
Instancia	Workspace	get_object	<ul style="list-style-type: none"> <li>queryset: dict</li> </ul>
Instancia	URL	get_success_url	

7.1.1.5.1.2.4 Clase *WorkspaceList*

Nombre	Descripción		Hereda de
<b>WorkspaceList</b>	Clase controladora que gestiona la vista de listado de Workspaces.		braces.views.LoginRequiredMixin django.views.generic.ListView
Atributos			
Acceso	Tipo o Clase	Nombre	
público	Workspace	model	
público	String	template_name	
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
Instancia	Context	get_context_data	<ul style="list-style-type: none"> <li>kwargs: dict</li> </ul>
Instancia	List	get_queryset	
Instancia	URL	get	<ul style="list-style-type: none"> <li>request: Request</li> <li>args: dict</li> <li>kwargs: dict</li> </ul>

7.1.1.5.1.2.5 Clase *WorkspaceUpdate*

Nombre	Descripción		Hereda de
<b>WorkspaceUpdate</b>	Clase controladora que gestiona la vista de actualización de un Workspace.		braces.views.LoginRequiredMixin django.views.generic.UpdateView
Atributos			
Acceso	Tipo o Clase	Nombre	
público	Workspace	model	
público	String	template_name	

público	Form	form_class	
público	List	fields	
público	String	slug_field	
público	String	slug_url_kwargs	
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
Instancia	URL	get_success_url	
Instancia	URL	get	<ul style="list-style-type: none"> <li>request: Request</li> <li>args: dict</li> <li>kwargs: dict</li> </ul>
Instancia	URL	post	<ul style="list-style-type: none"> <li>request: Request</li> <li>args: dict</li> <li>kwargs: dict</li> </ul>

### 7.1.1.5.1.3 goals tags

TagsList
-model : Tag -template_name
+get_queryset() +get(request, args, kwargs)

TagCreate
-fields -model : Tag -object
+get_success_url() +form_valid(form)

TagDetail
-current_user_permissions -model : Tag -object -slug_field -slug_url_kwarg -template_name
+get(request, args, kwargs) +post(request, args, kwargs)

TagUpdate
-fields -model : Tag -object -slug_field -slug_url_kwarg -template_name
+get(request, args, kwargs) +post(request, args, kwargs) +get_success_url()

TagDelete
-http_method_names -model : Workspace -slug_field -slug_url_kwarg
+get_success_url() +get_object(queryset)

Ilustración 46. Controladores de las vistas de tags del sitio web

En este módulo se incluyen las vistas encargadas de gestionar y de mostrar la información relativa a las Tags.

7.1.1.5.1.3.1 Clase *TagsList*

Nombre	Descripción	Hereda de	
<b>TagsList</b>	Clase controladora que gestiona la vista de listado de Tags.	braces.views.LoginRequiredMixin django.views.generic.ListView	
Atributos			
Acceso	Tipo o Clase	Nombre	
público	Tag	model	
público	String	template_name	
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
Instancia	URL	get	- args: dict - kwargs: dict
Instancia	List	get_queryset	

## 7.1.1.5.1.3.1.1 Ejemplo del controlador

```

class TagList(LoginRequiredMixin, ListView):
    model = Tag
    template_name = 'goals/tag_list.html'

    def get_queryset(self):
        return Tag.objects.filter(owner=self.request.user)

    def get(self, request, *args, **kwargs):
        current_username = kwargs.pop('username', None)
        if not self.request.user.is_authenticated() or self.request.user.username !=
current_username:
            return HttpResponseRedirect(reverse('home'))
        else:
            return super(TagList, self).get(request, *args, **kwargs)

```



## 7.1.1.5.1.3.2 Clase TagCreate

Nombre	Descripción	Hereda de	
TagCreate	Clase controladora que gestiona la vista de creación de Tags.	braces.views.LoginRequiredMixin django.views.generic.CreateView	
Atributos			
Acceso	Tipo o Clase	Nombre	
público	Tag	model	
público	List	fields	
público	Tag	object	
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
Instancia	URL	get_success_url	
Instancia	void	form_valid	- form: Form

## 7.1.1.5.1.3.2.1 Ejemplo del controlador

```

class TagCreate(LoginRequiredMixin, CreateView):
    model = Tag
    fields = ['title']
    template_name = 'goals/tag_form.html'

    def get_success_url(self):
        url = reverse("tags_list",
                    kwargs={'username': self.request.user.username})
        return HttpResponseRedirect(url)

    def form_valid(self, form):
        self.object = form.save(commit=False)
        self.object.owner = self.request.user
        try:
            self.object.save()
        except IntegrityError:
            messages.add_message(self.request, messages.ERROR,
                                "Already exists a tag with that name")
            return render(self.request, self.template_name, {'form': form})
        return self.get_success_url()

```

## 7.1.1.5.1.3.3 Clase TagDelete

Nombre	Descripción	Hereda de	
<b>TagDelete</b>	Clase controladora que gestiona la vista de borrado de Tags.	braces.views.LoginRequiredMixin django.views.generic.DeleteView	
Atributos			
Acceso	Tipo o Clase	Nombre	
público	Tag	model	
público	List	http_method_names	
público	String	slug_field	
público	String	slug_url_kwarg	
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
Instancia	URL	get_success_url	
Instancia	Tag	get_object	- queryset: dict

## 7.1.1.5.1.3.3.1 Ejemplo del controlador

```

class TagDelete(LoginRequiredMixin, DeleteView):
    model = Tag
    slug_url_kwarg = 'uuid'
    slug_field = 'uuid'
    http_method_names = ['post']

    def get_success_url(self):
        return reverse("tags_list", kwargs={'username': self.request.user.username})

    def get_object(self, queryset=None):

        uuid = self.kwargs.get(self.slug_url_kwarg, None)

        if not valid_uuid(uuid):
            raise Http404

        obj = super(TagDelete, self).get_object()
        if not obj.owner == self.request.user:
            raise Http404

```

```
messages.add_message(self.request, messages.SUCCESS, _("Tag deleted successfully"))  
return obj
```

## 7.1.1.5.1.3.4 Clase TagDetail

Nombre	Descripción	Hereda de	
<b>TagDetail</b>	Clase controladora que gestiona la vista de detalle de Tags.	braces.views.LoginRequiredMixin django.views.generic.DetailView	
Atributos			
Acceso	Tipo o Clase	Nombre	
público	Tag	model	
público	String	slug_field	
público	Tag	object	
público	String	slug_url_kwarg	
público	String	template_name	
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
<b>Instancia</b>	URL	get	<ul style="list-style-type: none"> <li>request: Request</li> <li>args: dict</li> <li>kwargs: dict</li> </ul>
<b>Instancia</b>	URL	post	<ul style="list-style-type: none"> <li>request: Request</li> <li>args: dict</li> <li>kwargs: dict</li> </ul>

## 7.1.1.5.1.3.4.1 Ejemplo de controlador

```

class TagDetail(LoginRequiredMixin, RemoveGoalFromListMixin, DetailView):
    model = Tag
    template_name = "goals/tag_details.html"
    slug_url_kwarg = 'uuid'
    slug_field = 'uuid'
    current_user_permissions = 3

    def get(self, request, *args, **kwargs):

```

```
self.object = self.get_object()
try:
    if request.user.pk is not self.get_object().owner.pk:
        raise Http404
except Http404:
    return HttpResponseRedirect(reverse('home'))

context = self.get_context_data(object=self.get_object())

if self.get_object() is not None:
    return self.render_to_response(context)
else:
    raise Http404

def post(self, request, *args, **kwargs):

    if self.remove_goal_from_list(self.get_object().goals, request):
        messages.add_message(request, messages.SUCCESS, _("Goal was removed successfully
from tag"))
    else:
        messages.add_message(request, messages.ERROR, _("Goal can not be removed from
tag"))

    url = reverse("tag_detail",
                  kwargs={'uuid': self.get_object().uuid, 'username':
self.get_object().owner.username})

    return HttpResponseRedirect(url)
```

7.1.1.5.1.3.5 Clase *TagUpdate*

Nombre	Descripción	Hereda de	
<b>TagUpdate</b>	Clase controladora que gestiona la vista de actualización de Tags	braces.views.LoginRequiredMixin django.views.generic.UpdateView	
Atributos			
Acceso	Tipo o Clase	Nombre	
público	Tag	model	
público	List	fields	
público	String	slug_field	
público	Tag	object	
público	String	slug_url_kwarg	
público	String	template_name	
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
Instancia	URL	get	<ul style="list-style-type: none"> <li>request: Request</li> <li>args: dict</li> <li>kwargs: dict</li> </ul>
Instancia	URL	post	<ul style="list-style-type: none"> <li>request: Request</li> <li>args: dict</li> <li>kwargs: dict</li> </ul>
Instancia	URL	get_success_url	

## 7.1.1.5.1.3.5.1 Ejemplo de controlador

```
class TagUpdate(LoginRequiredMixin, UpdateView, CheckOwnerMixin):
    model = Tag
    template_name = "goals/tag_update_form.html"
```

```

fields = ['title']
slug_url_kwarg = 'uuid'
slug_field = 'uuid'

def get(self, request, *args, **kwargs):
    if not self.is_owner_current_user():
        raise Http404
    else:
        return super(TagUpdate, self).get(request, *args, **kwargs)

def get_success_url(self):
    url = reverse("tag_detail", kwargs={'uuid': self.get_object().uuid, 'username':
self.request.user.username})
    return url

def post(self, request, *args, **kwargs):

    if not self.is_owner_current_user():
        raise Http404

    self.object = self.get_object()
    form_class = self.get_form_class()
    form = self.get_form(form_class)
    if form.is_valid():
        messages.add_message(self.request, messages.SUCCESS, _("Tag updated
successfully"))
        return self.form_valid(form)
    else:
        messages.add_message(self.request, messages.ERROR, _("Tag can not be updated"))
        return self.form_invalid(form)

```

#### 7.1.1.5.1.4 goals snippets

SnippetDetail
-model : Workspace
-object
-slug_field
-slug_url_kwarg
-template_name
+get_success_url()
+get_context_data(kwargs)
+current_user_can_view_goal()
+get(request, args, kwargs)
+restore_snippet_version(request)
+post(request, args, kwargs)

Ilustración 47. Controlador de la vista de detalles de Snippet

En este módulo se incluyen las vistas encargadas de gestionar y de mostrar la información relativa a los Snippets.

##### 7.1.1.5.1.4.1 Clase SnippetDetail

Nombre	Descripción	Hereda de
SnippetDetail	Clase controladora de la vista de detalle de un Snippet.	braces.views.LoginRequiredMixin

		django.views.generic.DetailView	
Atributos			
Acceso	Tipo o Clase	Nombre	
público	String	template_name	
público	dict	current_user_permissions	
público	Class	model	
público	Snippet	object	
público	String	slug_field	
público	String	slug_url_kwarg	
público	String	template_name	
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
Instancia	URL	get	- args: dict - kwargs: dict
Instancia	URL	post	- args: dict - kwargs: dict
Instancia	URL	get_success_url	
Instancia	Boolean	current_user_can_view_goal	
Instancia	void	restore_snippet_version	- args: dict

#### 7.1.1.5.1.5 goals download

DataDownload	SnippetDownload
-template_name	-template_name
+get(request, args, kwargs)	+get(request, args, kwargs)
+post(request, args, kwargs)	+post(request, args, kwargs)

Ilustración 48. Controladores de las vistas de descarga de datos



En este módulo se incluyen las vistas encargadas de gestionar y de mostrar la información relativa a la descarga de datos desde la aplicación.

#### 7.1.1.5.1.5.1 Clase `DataDownload`

Nombre	Descripción	Hereda de	
<b>DataDownload</b>	Clase controladora de las descargas de todos los datos de la aplicación.	braces.views.LoginRequiredMixin django.views.generic.TemplateView	
Atributos			
Acceso	Tipo o Clase	Nombre	
	String	template_name	
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
Instancia	URL	get	- args: dict - kwargs: dict
Instancia	URL	post	- args: dict - kwargs: dict

##### 7.1.1.5.1.5.1.1 Ejemplo del formato del código descargado

Cuando se llama a este controlador de vista, el sistema genera un fichero con todos los datos del usuario serializados en formato JSON. Estos datos incluyen toda la información que el usuario pudiera necesitar para gestionar sus goals y snippets por su cuenta. A continuación se muestra un ejemplo del código generado:

```
[
  {
    "description": "Export your database using a Cronjob",
    "title": "Database Backup Cronjob",
    "creation_date": "2015-04-19 09:35:58.528305+00:00",
    "is_private": false,
    "id": "071197e2b3b34e268a0e377bff57f9d8",
    "snippets": [
      {
        "language": "sh",
        "title": "cronjob.sh",
        "text": "DBNAME=\"<DB_NAME>\" \r\nDBPASS=\"<PASSWORD>\" \r\nDBUSER=\"<USER>\" \r\nEMAI
L=\"<EMAIL>\" \r\n \r\nmysqldump --opt -u $DBUSER -
p$DBPASS $DBNAME > backup.sql\r\nngzip backup.sql\r\nDATE=`date +%Y%m%d` ; mv backup.sql.gz $DB
NAME-backupFULL-$DATE.sql.gz\r\nuuencode $DBNAME-backupFULL-$DATE.sql.gz $DBNAME-backupFULL-
$DATE.sql.gz | mail -s \"<SUBJECT>\" $EMAIL\r\nrm $DBNAME-backupFULL-$DATE.sql.gz",
        "creation_date": "2015-04-19 09:36:01.517975+00:00",

```

```

        "version":1,
        "goal_id":"071197e2b3b34e268a0e377bff57f9d8",
        "id":"89d6f7c964784fc180fd94592bf939b3"
    }
  ],
  ...
]

```

#### 7.1.1.5.1.5.2 Clase SnippetDownload

Nombre	Descripción	Hereda de	
<b>SnippetDownload</b>	Clase controladora de las descargas individuales de Snippets.	braces.views.LoginRequiredMixin django.views.generic.TemplateView	
Atributos			
Acceso	Tipo o Clase	Nombre	
	String	template_name	
Métodos			
Acceso   Modo	Tipo de Retorno	Nombre	Parámetros y tipos
<b>Instancia</b>	URL	get	- args: dict - kwargs: dict
<b>Instancia</b>	URL	post	- args: dict - kwargs: dict

##### 7.1.1.5.1.5.2.1 Formato del código descargado

Cuando se descarga individualmente el código de un snippet o de un goal, la descarga se realiza en formato txt independientemente del formato del archivo descargado. En este caso no se descarga como JSON porque se determina que el usuario desea utilizar el código directamente, no procesarlo ni hacer una copia de seguridad del mismo.

### 7.1.1.5.2 forms



Ilustración 49. Módulos de formularios del sitio web

Los formularios principales de la aplicación web se dividen en dos grupos: users, para gestionar los formularios relacionados con la gestión de los usuarios y goals, para gestionar los datos de la aplicación.

La utilización de estos formularios es para validar en el *backend* los datos introducidos por los usuarios en los formularios del *frontend* desde un punto de vista orientado a objetos.

#### 7.1.1.5.2.1 users

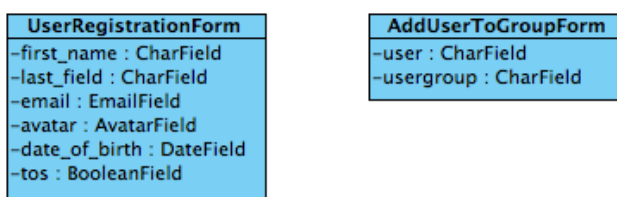


Ilustración 50. Modelos de los formularios de usuarios

Formularios relacionados con los usuarios de la aplicación, como los formularios de creación de un nuevo usuario de de adición a un grupo de usuarios.

##### 7.1.1.5.2.1.1 Clase UserRegistrationForm

Nombre	Descripción	Hereda de
<b>UserRegistrationForm</b>	Clase que valida el formulario de registro de nuevos usuarios en la aplicación.	registration.forms.RegistrationFormUniqueEmail
Atributos		
Acceso	Tipo o Clase	Nombre
público	CharField	first_name
público	CharField	last_name
público	EmailField	email
público	AvatarField	avatar
público	DateField	date_of_birth

público	BooleanField	tos
---------	--------------	-----

7.1.1.5.2.1.1.1 Ejemplo de formulario

```

class UserRegistrationForm(RegistrationFormUniqueEmail):
    #user
    first_name = forms.CharField(required=False, max_length=30)
    last_name = forms.CharField(required=False, max_length=100)
    email = forms.EmailField(required=True, label='Your e-mail address')
    avatar = avatar_forms.AvatarField(required=False, disable_preview=True, width=200,
height=200)

    #profile
    date_of_birth = forms.DateField(required=True, input_formats=settings.DATE_INPUT_FORMATS)

    #form
    tos = forms.BooleanField(widget=forms.CheckboxInput,
label=(u'I have read and agree to the Terms of Service'),
error_messages={'required': _("You must agree to the terms to
register")})
    
```

7.1.1.5.2.1.1.2 Proceso del registro de usuarios

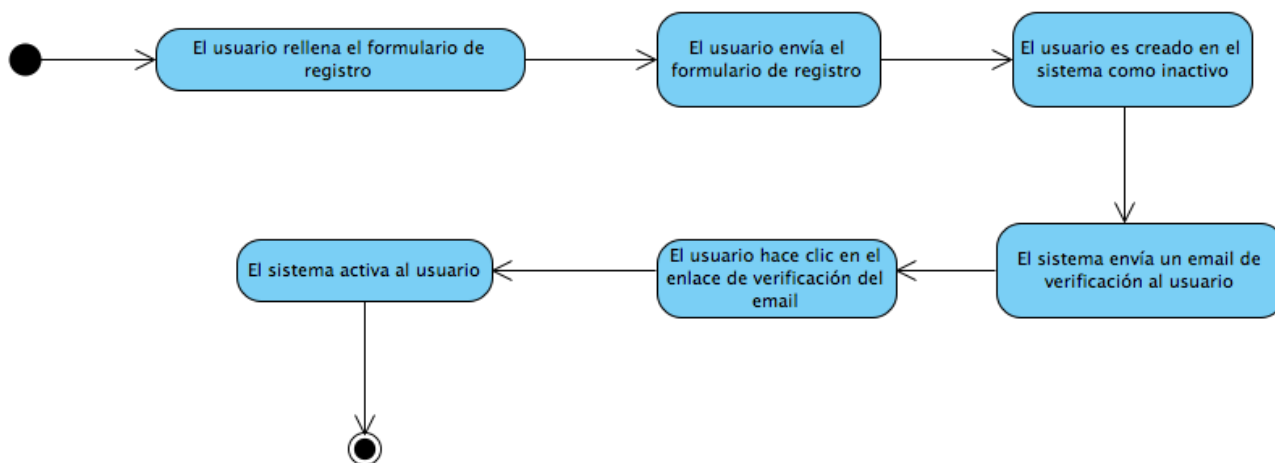


Ilustración 51. Diagrama de actividad del registro de usuarios

7.1.1.5.2.1.2 Clase AddUserToGroupForm

Nombre	Descripción	Hereda de
AddUserToGroupForm	Clase que valida el formulario que añade un usuario a un grupo de usuarios.	django.forms.Form
Atributos		
Acceso	Tipo o Clase	Nombre
público	CharField	user

público	CharField	usergroup
---------	-----------	-----------

7.1.1.5.2.1.3 goals

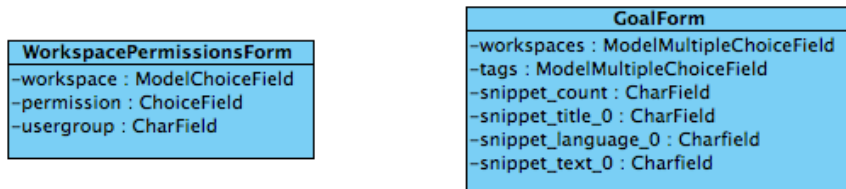


Ilustración 52. Modelos de los formularios de goals

Formularios relacionados con la gestión de Workspaces y Goals dentro de la aplicación.

7.1.1.5.2.1.3.1 Clase WorkspacePermissionForm

Nombre	Descripción	Hereda de
<b>WorkspacePermissionForm</b>	Clase que valida el formulario de adición de un Workspace a un grupo de usuarios con unos determinados permisos	django.forms.ModelForm
Atributos		
Acceso	Tipo o Clase	Nombre
público	ModelChoiceField	workspace
público	ChoiceField	permission
público	CharField	usergroup

7.1.1.5.2.1.3.2 Clase GoalForm

Nombre	Descripción	Hereda de
<b>GoalForm</b>	Clase que valida los formularios de creación y actualización de Goals.	django.forms.ModelForm
Atributos		
Acceso	Tipo o Clase	Nombre
público	ModelMultipleChoiceField	workspaces

<b>público</b>	ModelMultipleChoiceField	tags
<b>público</b>	CharField	snippet_count
<b>público</b>	CharField	snippet_title_0
<b>público</b>	CharField	snippet_language_0
<b>público</b>	CharField	snippet_text_0

## 7.2 Estructura de la base de datos

La gestión de la base de datos de la aplicación no se realiza directamente, si no que se utiliza el ORM de Django para ello.

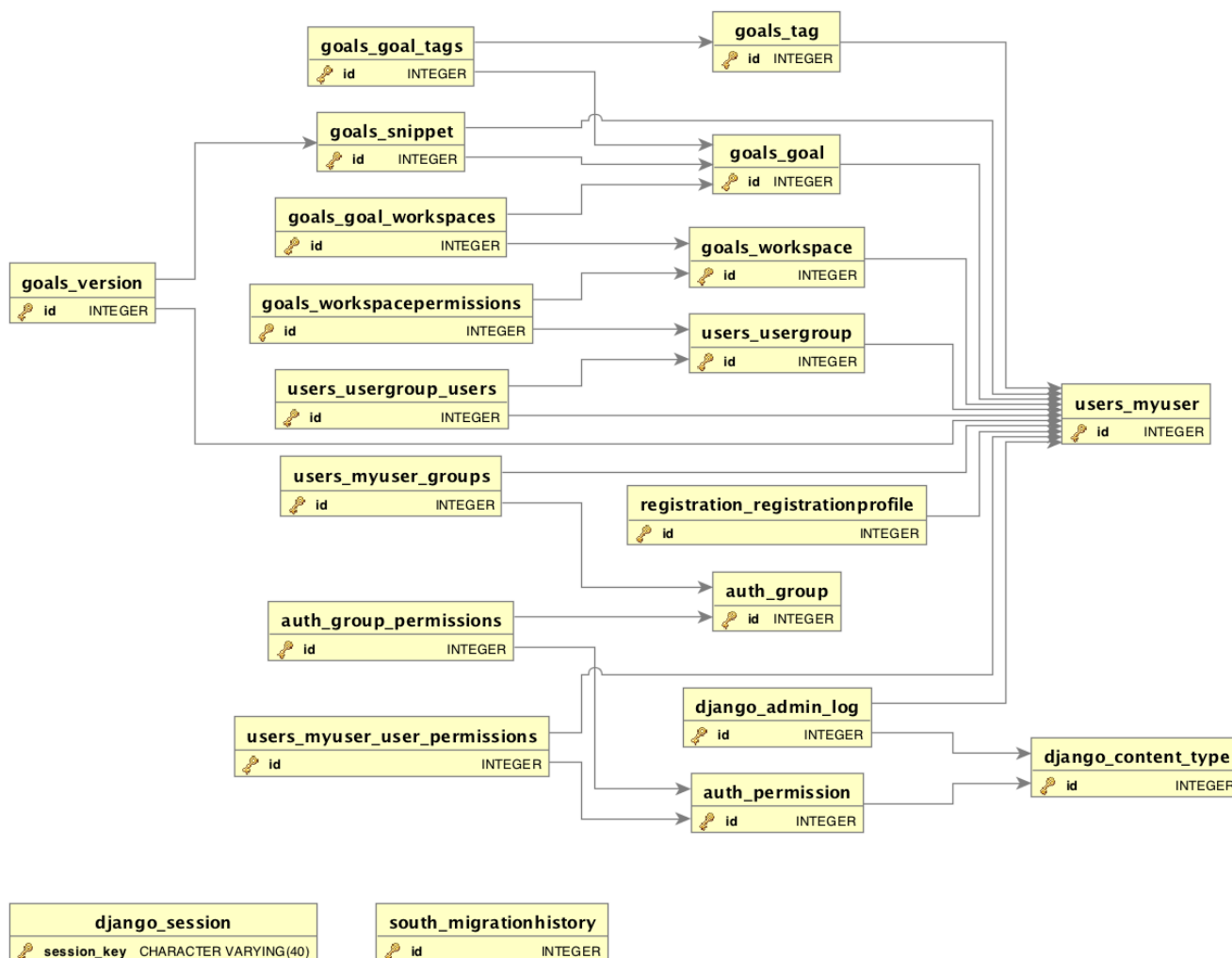


Ilustración 53. Estructura de las claves primarias de la base de datos

El diagrama superior muestra la estructura de las claves primarias de la base de datos, incluyendo tablas referentes al funcionamiento de Django. Como se puede apreciar, todos los elementos están relacionados con la tabla users\_myuser que hace referencia a los usuarios de la aplicación y existe una clara relación con los diagramas de clases del apartado 7.1.1.1.2.

Todos los elementos de la base de datos están relacionados entre sí por medio de un id que utiliza Django internamente. Por defecto este id se utiliza también en las URL y para realizar acciones sobre los objetos, pero por motivos de seguridad se ha decidido utilizar un UUID para hacer referencia a los objetos de manera pública. Se puede encontrar más información referente a esto en el apartado 8.3.4.3

### 7.2.1 Backups de la base de datos

Puesto que el sistema se encontrará en producción y albergará información de cierto número de usuarios, se realizarán copias de seguridad periódicas de la base de datos mediante un Shell Script. Estas copias se

almacenarán en el servidor. El script utilizado se trata de un script opensource obtenido de Sourceforge basado en AutoMySQLbackup.<sup>11</sup> llamado AutoPgSQLbackup<sup>12</sup> que se ejecuta en el servidor por medio de un CRON Job.

A continuación se incluye un ejemplo del código utilizado para realizar las copias de seguridad.

```
#!/bin/bash
# Username to access the PostgreSQL server e.g. dbuser
USERNAME=postgres
# Password
DBHOST=localhost
# List of DBNAMES for Daily/Weekly Backup e.g. "DB1 DB2 DB3"
DBNAMES="all"
# Backup directory location e.g /backups
BACKUPDIR="/home/backups/pgsql_backups"

# Mail setup
MAILCONTENT="log"
# Set the maximum allowed email size in k. (4000 = approx 5MB email [see docs])
MAXATTSIZE="4000"
# Email Address to send mail to? (user@domain.com)
MAILADDR="root@localhost"
# List of DBNAMES for Monthly Backups.
MDBNAMES="template1 $DBNAMES"
# List of DBNAMES to EXCLUDE if DBNAMES are set to all (must be in " quotes)
DBEXCLUDE=""
# Include CREATE DATABASE in backup?
CREATE_DATABASE=yes
# Separate backup directory and file for each DB? (yes or no)
SEPDIR=yes
# Which day do you want weekly backups? (1 to 7 where 1 is Monday)
DOWEEKLY=6
# Choose Compression type. (gzip or bzip2)
COMP=gzip
# Command to run before backups (uncomment to use)
#PREBACKUP="/etc/pgsql-backup-pre"
# Command run after backups (uncomment to use)
POSTBACKUP="bash /home/backups/scripts/ftp_pgsql"
```

---

<sup>11</sup> Página oficial de AutoMySQLBackup: <http://sourceforge.net/projects/automysqlbackup/>

<sup>12</sup> Página oficial de AutoPgSQLBackup: <http://autopgsqlbackup.frozenpc.net/>



```

PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/postgres/bin:/usr/local/pgsql/bin
DATE=`date +%Y-%m-%d`           # Datestamp e.g 2002-09-21
DOW=`date +%A`                 # Day of the week e.g. Monday
DNOW=`date +%u`                 # Day number of the week 1 to 7 where 1
represents Monday
DOM=`date +%d`                 # Date of the Month e.g. 27
M=`date +%B`                   # Month e.g January
W=`date +%V`                   # Week Number e.g 37
VER=1.0                         # Version Number
LOGFILE=$BACKUPDIR/$DBHOST-`date +%N`.log # Logfile Name
OPT=""                          # OPT string for use with mysqldump ( see man mysqldump
)
BACKUPFILES=""                 # thh: added for later mailing

# Create required directories
if [ ! -e "$BACKUPDIR" ]       # Check Backup Directory exists.
    then
        mkdir -p "$BACKUPDIR"
    fi

if [ ! -e "$BACKUPDIR/daily" ] # Check Daily Directory exists.
    then
        mkdir -p "$BACKUPDIR/daily"
    fi

if [ ! -e "$BACKUPDIR/weekly" ] # Check Weekly Directory exists.
    then
        mkdir -p "$BACKUPDIR/weekly"
    fi

if [ ! -e "$BACKUPDIR/monthly" ] # Check Monthly Directory exists.
    then
        mkdir -p "$BACKUPDIR/monthly"
    fi

# IO redirection for logging.
touch $LOGFILE
exec 6>&1          # Link file descriptor #6 with stdout.

```

```

                # Saves stdout.
exec > $LOGFILE      # stdout replaced with file $LOGFILE.

# Functions

# Database dump function
dbdump () {
pg_dump --username=$USERNAME $HOST $OPT $1 > $2
return 0
}

# Compression function
SUFFIX=""
compression () {
if [ "$COMP" = "gzip" ]; then
    gzip -f "$1"
    echo
    echo Backup Information for "$1"
    gzip -l "$1.gz"
    SUFFIX=".gz"
elif [ "$COMP" = "bzip2" ]; then
    echo Compression information for "$1.bz2"
    bzip2 -f -v $1 2>&1
    SUFFIX=".bz2"
else
    echo "No compression option set, check advanced settings"
fi
return 0
}

# Run command before we begin
if [ "$PREBACKUP" ]
then
    echo =====
    echo "Prebackup command output."
    echo

```

```

eval $PREBACKUP
echo
echo =====
echo
fi

if [ "$SEPDIR" = "yes" ]; then # Check if CREATE DATABASE should be included in Dump
    if [ "$CREATE_DATABASE" = "no" ]; then
        OPT="$OPT"
    else
        OPT="$OPT --create"
    fi
else
    OPT="$OPT"
fi

# Hostname for LOG information
if [ "$DBHOST" = "localhost" ]; then
    DBHOST="`hostname -f`"
    HOST=""
else
    HOST="-h $DBHOST"
fi

# If backing up all DBs on the server
if [ "$DBNAMES" = "all" ]; then
    DBNAMES="`psql -U $USERNAME $HOST -l -A -F: | sed -ne '/:/ { /Name:Owner/d;
/template0/d; s/:.*/;/; p }`"

    # If DBs are excluded
    for exclude in $DBEXCLUDE
    do
        DBNAMES=`echo $DBNAMES | sed "s/\b$exclude\b//g"`
    done

    MDBNAMES=$DBNAMES
fi

```

```

echo =====
echo AutoPostgreSQLBackup VER $VER
echo http://autopgsqlbackup.frozenpc.net/
echo
echo Backup of Database Server - $DBHOST
echo =====

# Test is seperate DB backups are required
if [ "$SEPDIR" = "yes" ]; then
echo Backup Start Time `date`
echo =====

    # Monthly Full Backup of all Databases
    if [ $DOM = "01" ]; then
        for MDB in $MDBNAMES
        do

                # Prepare $DB for using
                MDB=`echo $MDB | sed 's%/ /g'`

                if [ ! -e "$BACKUPDIR/monthly/$MDB" ]           # Check Monthly DB Directory
exists.
                then
                    mkdir -p "$BACKUPDIR/monthly/$MDB"
                fi
                echo Monthly Backup of $MDB...
                    dbdump "$MDB" "$BACKUPDIR/monthly/$MDB/${MDB}_$DATE.$M.$MDB.sql"
                    compression "$BACKUPDIR/monthly/$MDB/${MDB}_$DATE.$M.$MDB.sql"
                    BACKUPFILES="$BACKUPFILES
$BACKUPDIR/monthly/$MDB/${MDB}_$DATE.$M.$MDB.sql$SUFFIX"
                echo -----
-
                    done
        fi

        for DB in $DBNAMES
        do
            # Prepare $DB for using
            DB=`echo $DB | sed 's%/ /g'`

```

```

# Create Seperate directory for each DB
if [ ! -e "$BACKUPDIR/daily/$DB" ]           # Check Daily DB Directory exists.
then
    mkdir -p "$BACKUPDIR/daily/$DB"
fi

if [ ! -e "$BACKUPDIR/weekly/$DB" ]         # Check Weekly DB Directory exists.
then
    mkdir -p "$BACKUPDIR/weekly/$DB"
fi

# Weekly Backup
if [ $DNOW = $DOWEEKLY ]; then
    echo Weekly Backup of Database \( $DB \)
    echo Rotating 5 weeks Backups...
    if [ "$W" -le 05 ];then
        REMW=`expr 48 + $W`
    elif [ "$W" -lt 15 ];then
        REMW=0`expr $W - 5`
    else
        REMW=`expr $W - 5`
    fi
    eval rm -fv "$BACKUPDIR/weekly/$DB/week.$REMW.*"
    echo
    dbdump "$DB" "$BACKUPDIR/weekly/$DB/${DB}_week.$W.$DATE.sql"
    compression "$BACKUPDIR/weekly/$DB/${DB}_week.$W.$DATE.sql"
    BACKUPFILES="$BACKUPFILES
$BACKUPDIR/weekly/$DB/${DB}_week.$W.$DATE.sql$SUFFIX"
    echo -----

# Daily Backup
else
    echo Daily Backup of Database \( $DB \)
    echo Rotating last weeks Backup...
    eval rm -fv "$BACKUPDIR/daily/$DB/*.DOW.sql.*"
    echo
    dbdump "$DB" "$BACKUPDIR/daily/$DB/${DB}_$DATE.$DOW.sql"
    compression "$BACKUPDIR/daily/$DB/${DB}_$DATE.$DOW.sql"

```

```

                BACKUPFILES="$BACKUPFILES
$BACKUPDIR/daily/$DB/${DB}_${DATE}.$DOW.sql$SUFFIX"
            echo -----
            fi
            done
echo Backup End `date`
echo =====

else # One backup file for all DBs
echo Backup Start `date`
echo =====
    # Monthly Full Backup of all Databases
    if [ $DOM = "01" ]; then
        echo Monthly full Backup of \( $MDBNAMES \)...
        dbdump "$MDBNAMES" "$BACKUPDIR/monthly/$DATE.$M.all-databases.sql"
        compression "$BACKUPDIR/monthly/$DATE.$M.all-databases.sql"
        BACKUPFILES="$BACKUPFILES $BACKUPDIR/monthly/$DATE.$M.all-
databases.sql$SUFFIX"
        echo -----
    fi

    # Weekly Backup
    if [ $DNOW = $DOWEEKLY ]; then
        echo Weekly Backup of Databases \( $DBNAMES \)
        echo
        echo Rotating 5 weeks Backups...
        if [ "$W" -le 05 ];then
            REMW=`expr 48 + $W`
        elif [ "$W" -lt 15 ];then
            REMW=0`expr $W - 5`
        else
            REMW=`expr $W - 5`
        fi
        eval rm -fv "$BACKUPDIR/weekly/week.$REMW.*"
        echo
        dbdump "$DBNAMES" "$BACKUPDIR/weekly/week.$W.$DATE.sql"
        compression "$BACKUPDIR/weekly/week.$W.$DATE.sql"
        BACKUPFILES="$BACKUPFILES $BACKUPDIR/weekly/week.$W.$DATE.sql$SUFFIX"
    fi

```

```

        echo -----

# Daily Backup
else
    echo Daily Backup of Databases \( $DBNAMES \)
    echo
    echo Rotating last weeks Backup...
    eval rm -fv "$BACKUPDIR/daily/*.DOW.sql.*"
    echo
        dbdump "$DBNAMES" "$BACKUPDIR/daily/$DATE.$DOW.sql"
        compression "$BACKUPDIR/daily/$DATE.$DOW.sql"
        BACKUPFILES="$BACKUPFILES $BACKUPDIR/daily/$DATE.$DOW.sql$SUFFIX"
    echo -----

fi
echo Backup End Time `date`
echo =====
fi
echo Total disk space used for backup storage..
echo Size - Location
echo `du -hs "$BACKUPDIR"`
echo

# Run command when we're done
if [ "$POSTBACKUP" ]
    then
        echo =====
        echo "Postbackup command output."
        echo
        eval $POSTBACKUP
        echo
        echo =====
fi

#Clean up IO redirection
exec 1>&6 6>&-      # Restore stdout and close file descriptor #6.

if [ "$MAILCONTENT" = "files" ]

```

```
then
    #Get backup size
    ATTSIZE=`du -c $BACKUPFILES | grep "[[:digit:][:space:]]total$" | sed s/\s*total//`
    if [ $MAXATTSIZE -ge $ATTSIZE ]
    then
        BACKUPFILES=`echo "$BACKUPFILES" | sed -e "s# # -a #g"` #enable multiple
    attachments
        mutt -s "PostgreSQL Backup Log and SQL Files for $DBHOST - $DATE" $BACKUPFILES
    $MAILADDR < $LOGFILE #send via mutt
    else
        cat "$LOGFILE" | mail -s "WARNING! - PostgreSQL Backup exceeds set maximum
    attachment size on $HOST - $DATE" $MAILADDR
    fi
elif [ "$MAILCONTENT" = "log" ]
then
    cat "$LOGFILE" | mail -s "PostgreSQL Backup Log for $DBHOST - $DATE" $MAILADDR
else
    cat "$LOGFILE"
fi

# Clean up Logfile
eval rm -f "$LOGFILE"

exit 0
```



## 7.3 Diseño de la interfaz de usuario

La interfaz de la aplicación ha sido creada utilizando MHTML5 y CSS mediante el framework de creación de aplicaciones web Bootstrap y el kit de interfaz de usuario Bootflat, que proporciona un mayor número de elementos de diseño y un estilo “Flat Design” aprovechando todas las cualidades de Bootstrap.

También existen elementos de la *UI*<sup>13</sup> escritos en Javascript utilizando principalmente el framework JQuery para incluir elementos visuales como acordeones en ciertas partes de la página, burbujas de ayuda al pasar el ratón sobre ciertos elementos o pop-ups para mostrar mensajes de confirmación al realizar ciertas acciones.

A continuación se muestran las capturas de las páginas más importantes de la aplicación. Al haber muchas y ser todas ellas bastante parecidas se obviarán aquellas que tengan una similitud notable con otras.

La aplicación se encuentra traducida a dos idiomas: español e inglés y se ha optado por mostrar la versión en español en las capturas.

---

<sup>13</sup> Siglas de *User Interface* o Interfaz de Usuario en inglés.

### 7.3.1 Página de inicio



Ilustración 54. Zona 1 de la página de inicio

La primera zona de la página de inicio tiene como objetivo llamar la atención del usuario de una manera visual, permitiéndole saber cuál es el principal objetivo de la aplicación con palabras concisas como “Almacena”, “Organiza”, etc.

El menú superior estará presente en todas las páginas del sitio web y supondrá una de las alternativas que tendrá el usuario para realizar ciertas acciones comunes (Crear un goal, acceder a sus goals, acceder a su perfil, etc).



Ilustración 55. Menú superior del sitio web

El menú superior varía en función de si un usuario está registrado o no. En el caso de que no esté, mostrará los botones de registro, acceso y también uno para notificar errores.

#	Nombre del Goal	Descripción	Snippets	Espacios de Trabajo
<b>Privado</b>	Python Views	Views of the Users module. <i>Etiquetas: Django, python</i>	2	PHP Maps, Python Project
<b>Público</b>	Project templates	HTML template from snippet download. <i>Etiquetas: Django, HTML</i>	1	PHP Maps
<b>Público</b>	Form with Database	Form with database code in PHP <i>Etiquetas: MySQL, PHP</i>	1	

Con Codesnug puedes mantener tu código organizado y clasificado. Se almacenará en la nube, por lo que siempre lo tendrás disponible.

Accede a tu código en cualquier momento, cópialo, descárgalo, expórtalo... ¡Es tuyo!

**Detector de Robots**

Inicio / Goals / Goal: Detector de Robots

Editar | **Borrar** | Volver

Código para detectar si un personaje es un robot. De una fiabilidad pasmosa.

```

detector.php
1 <?php
2 $characters = array("willon", "starbucks", "s1x",
3 "roslin", "eight", "baltan", "boomer");
4 $slyons_found = 0;
5 echo "Cylon detector started... OK";
6 foreach($characters as $character){
7     $syes = rand(0,1);
8     if($syes) $slyonCount++;
9     echo $character . " is" . (($syes)? "" : " not") . " a cylon";
10 }
11 echo "Cylon detector finished... OK";
12 echo $slyons_found . " clyons found.";
13 ?>
                
```

Información del Goal

Privacidad: **Privado**

Número de snippets: 1

Autor: rene1

Fecha de creación: 20 de Noviembre de 2014 a las 15:06

Espacios de Trabajo

- Trabajo

Comparte tu código con otras personas y colabora con ellos permitiéndoles editar tu código.

**Espacios de Trabajo Asignados**

+ Asignar Espacio de Trabajo

Nombre del Espacio de Trabajo	Permisos
Python Project	Subscriber
PHP Maps	Editor

## Otras características

- Más de 30 lenguajes de programación
- Exportación de código (texto plano y json)
- Sistema de búsqueda de código
- Posibilidad de código como privado o público
- API Rest para extender la aplicación
- Compatible con tablets y móviles

Ilustración 56. Zona 2 de la página de inicio

La segunda zona de la página de inicio tiene como objetivo ahondar un poco más en las características del sitio web, mostrando más capturas de pantalla sobre ellas y unos textos más extensos que los de la primera zona.

**¡Pruébalo!**

Nombre de usuario\*  
loladmin2014

Contraseña\*  
[Ocultar]

Email\*  
[Ocultar]

Contraseña (otra vez)\*  
[Ocultar]

Fecha de Nacimiento\*  
[Ocultar]  
Formato: dd/mm/aaaa.

Acepto la [Términos y Condiciones](#) de este sitio web. \*

[Crea una nueva cuenta](#)

Ilustración 57. Zona 3 de la página de inicio

La tercera zona está pensada para ofrecer una forma rápida de registro a aquellos visitantes que han hecho scroll hasta abajo de la página tras haber conocido las características de la aplicación. Este formulario muestra solamente los campos estrictamente necesarios para el registro para agilizar el proceso lo máximo posible.



Ilustración 58. Pie de página del sitio web

El pie de la página, al igual que el menú superior también es común en todas las páginas del sitio web. En él se mostrarán enlaces a las secciones de información del sitio web, como blog, política de privacidad, términos y condiciones, etc.

### 7.3.2 Página de registro

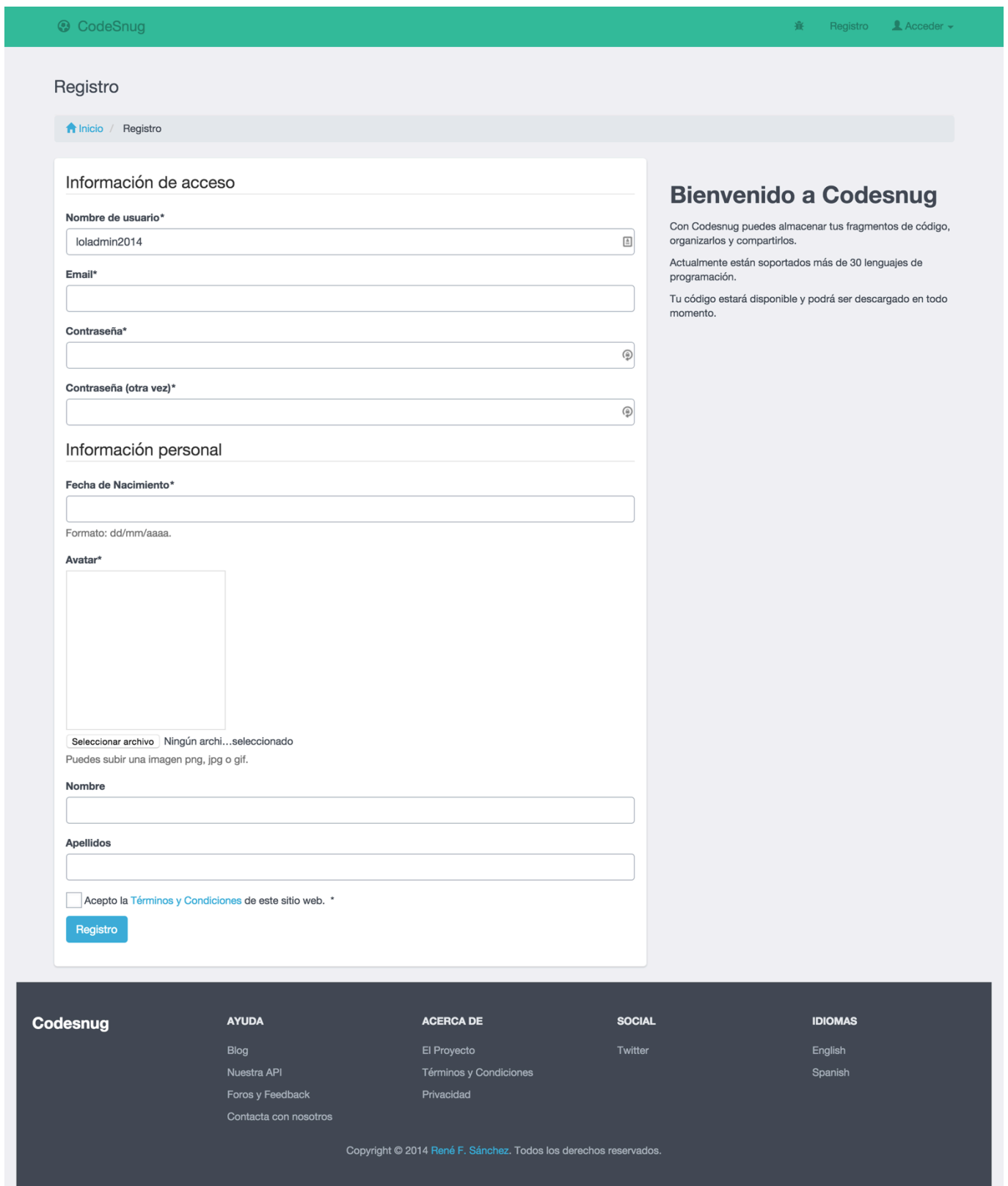


Ilustración 59. Página de registro del sitio web

La página de registro incluye un formulario para que el usuario introduzca sus datos para utilizar la aplicación. El formulario dispone de validación en el lado del cliente y del servidor y los errores se mostraran junto a los distintos campos del mismo.

### 7.3.3 Página de acceso

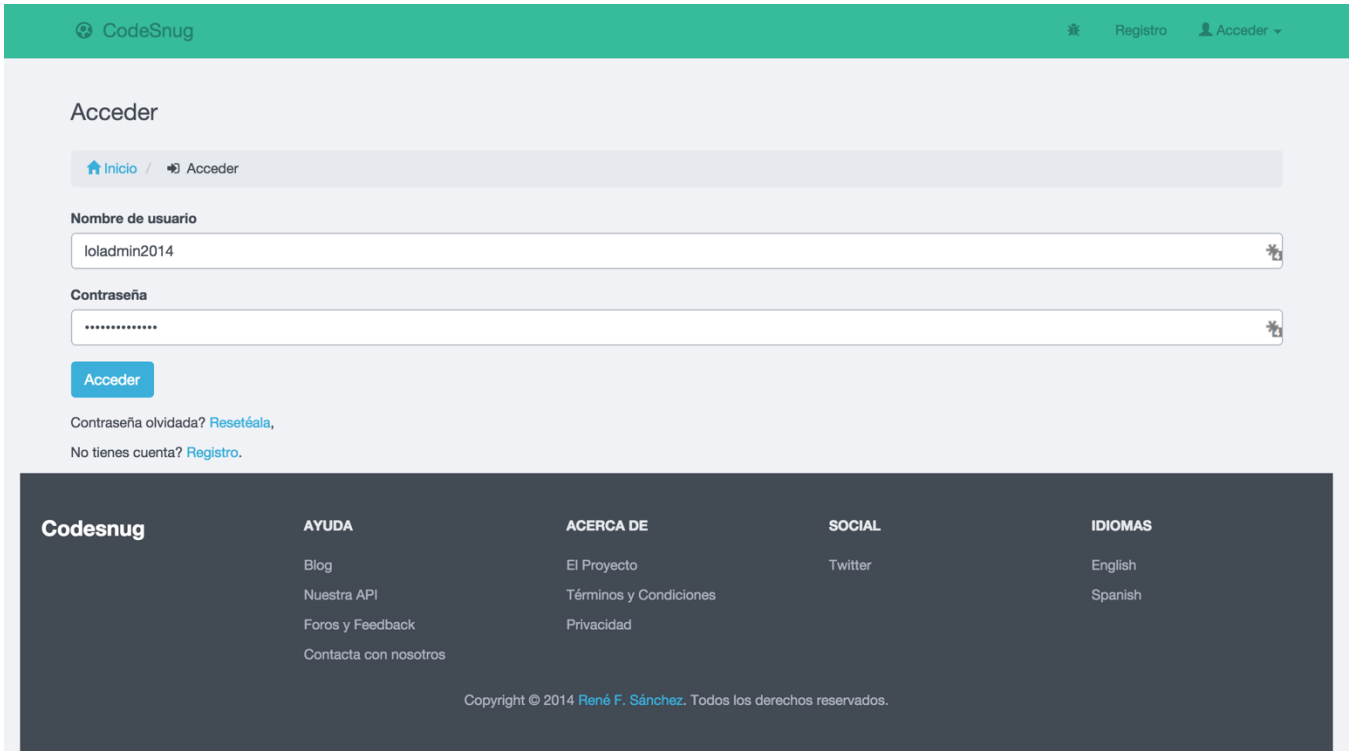


Ilustración 60. Página de acceso del sitio web

La página de acceso incluye el formulario de acceso al sitio web y los enlaces para restaurar la contraseña y registrarse. Al igual que en el caso del formulario de registro, si sucediese algún error al completar los campos, este sería notificaría junto a ellos.

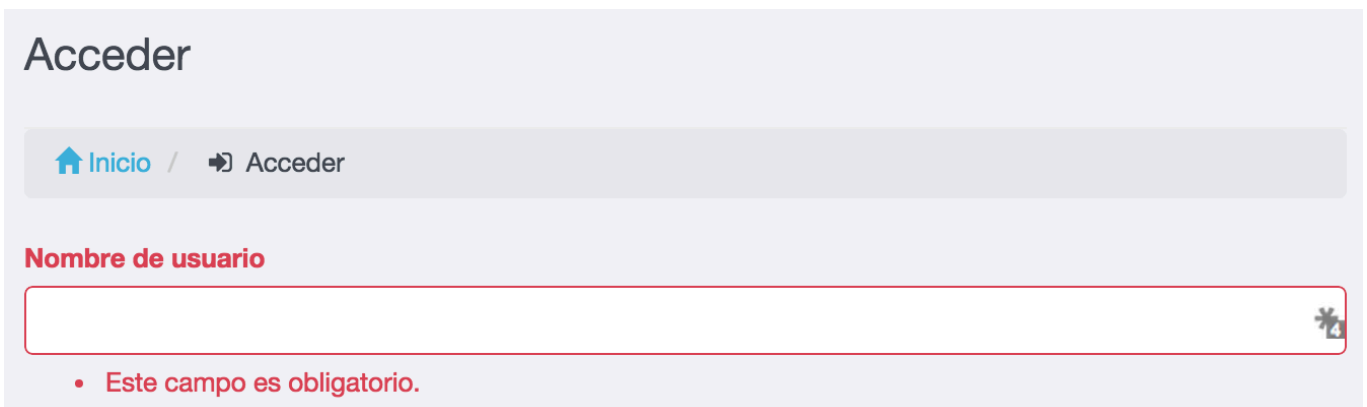


Ilustración 61. Ejemplo de error en un campo de formulario

### 7.3.4 Página de listado de goals

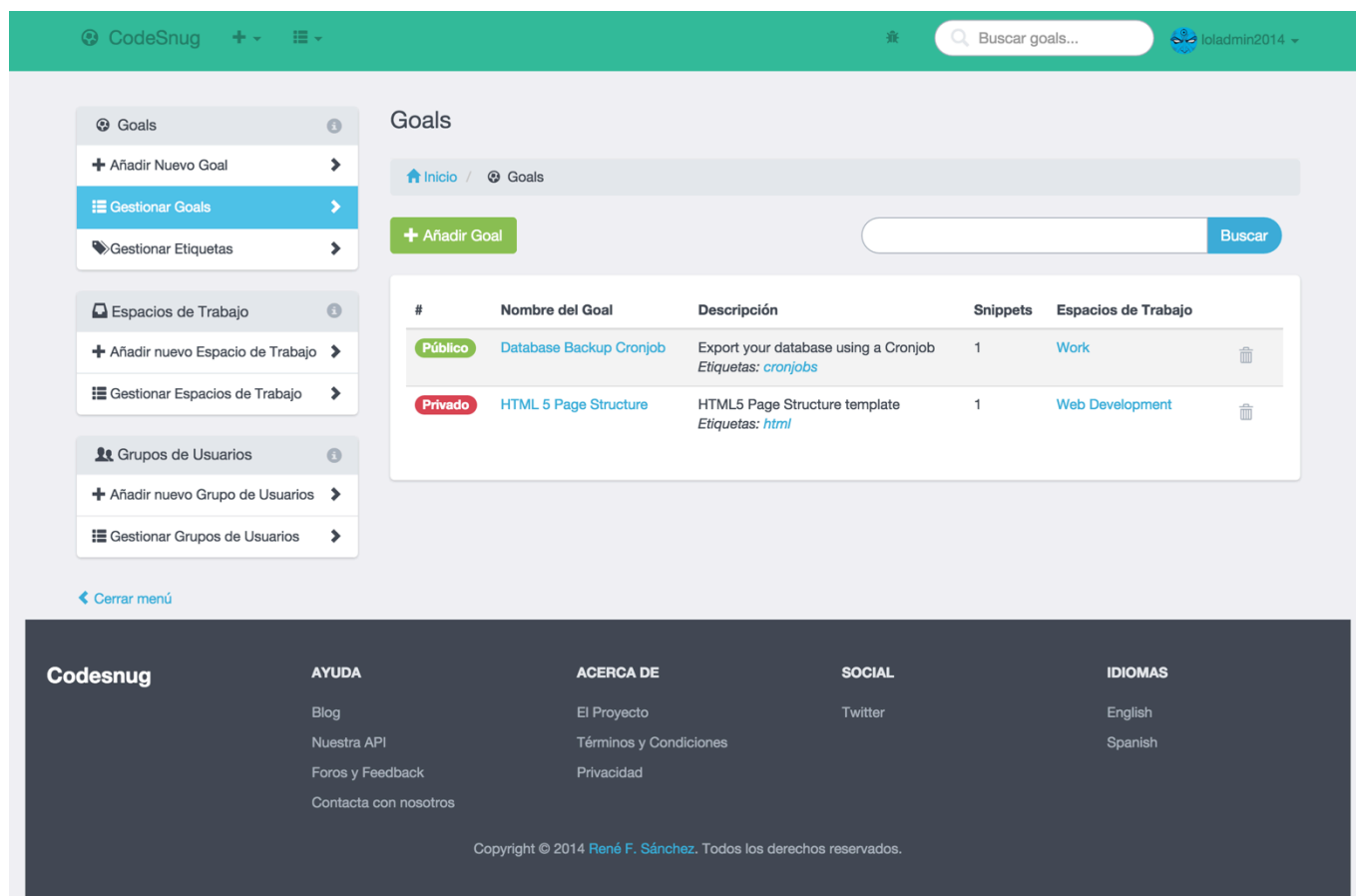


Ilustración 62. Página de listado de goals

Muestra una lista de todos los goals del usuario, una barra lateral y un menú superiores para acceder a las distintas partes del sitio web desde varios sitios diferentes como marcan las pautas de accesibilidad de sitios web.

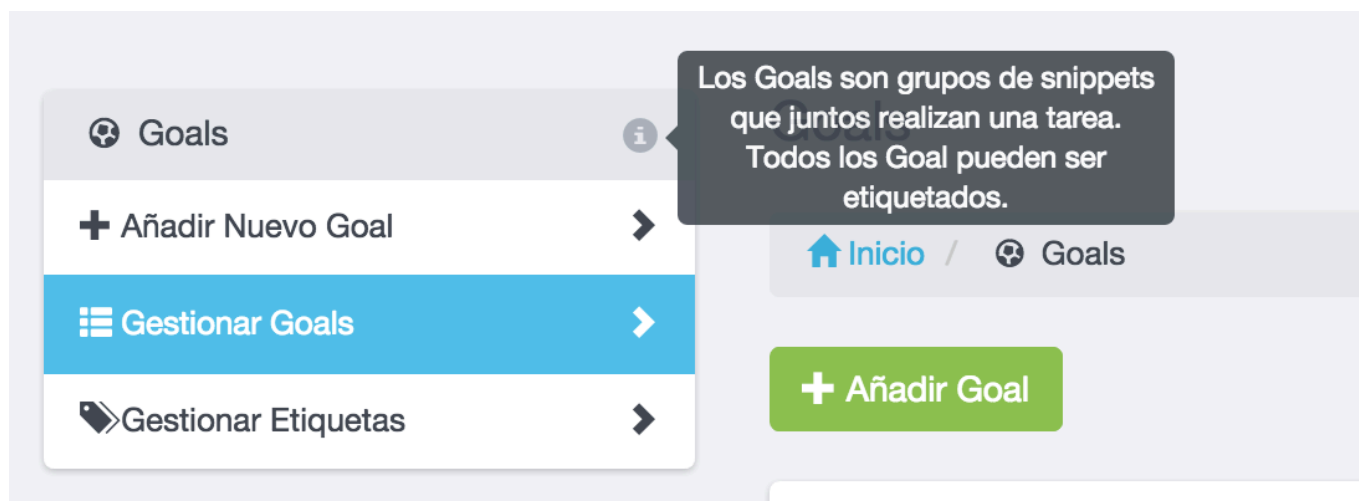


Ilustración 63. Información de la barra lateral del sitio web

Se han incluido botones de en la barra lateral, sobre los cuales se puede pasar el cursor para recibir información sobre la sección correspondiente.

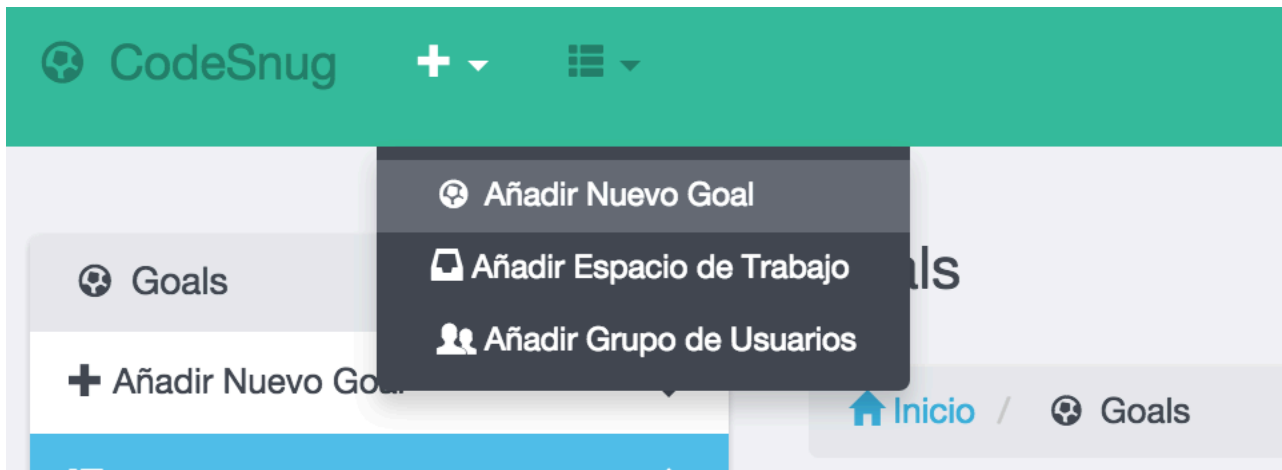


Ilustración 64. Zona izquierda del menú superior para usuarios registrados

Como se comentaba anteriormente, al estar registrado el usuario el menú superior cambia permitiéndole el acceso a las secciones internas de la página web por medio de un menú desplegable.

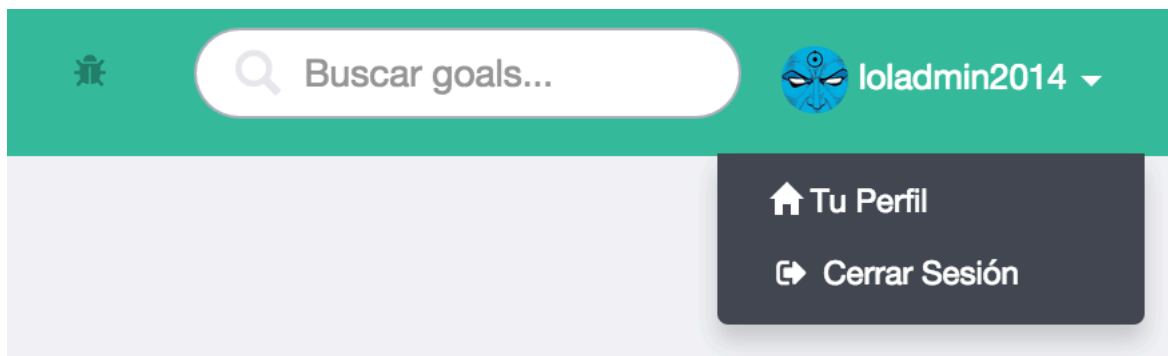


Ilustración 65. Zona derecha del menú superior para usuarios registrados

En el otro extremo del menú superior se muestra un formulario de búsqueda y los enlaces al perfil del usuario y al cierre de sesión.



### 7.3.5 Página de detalles de goal

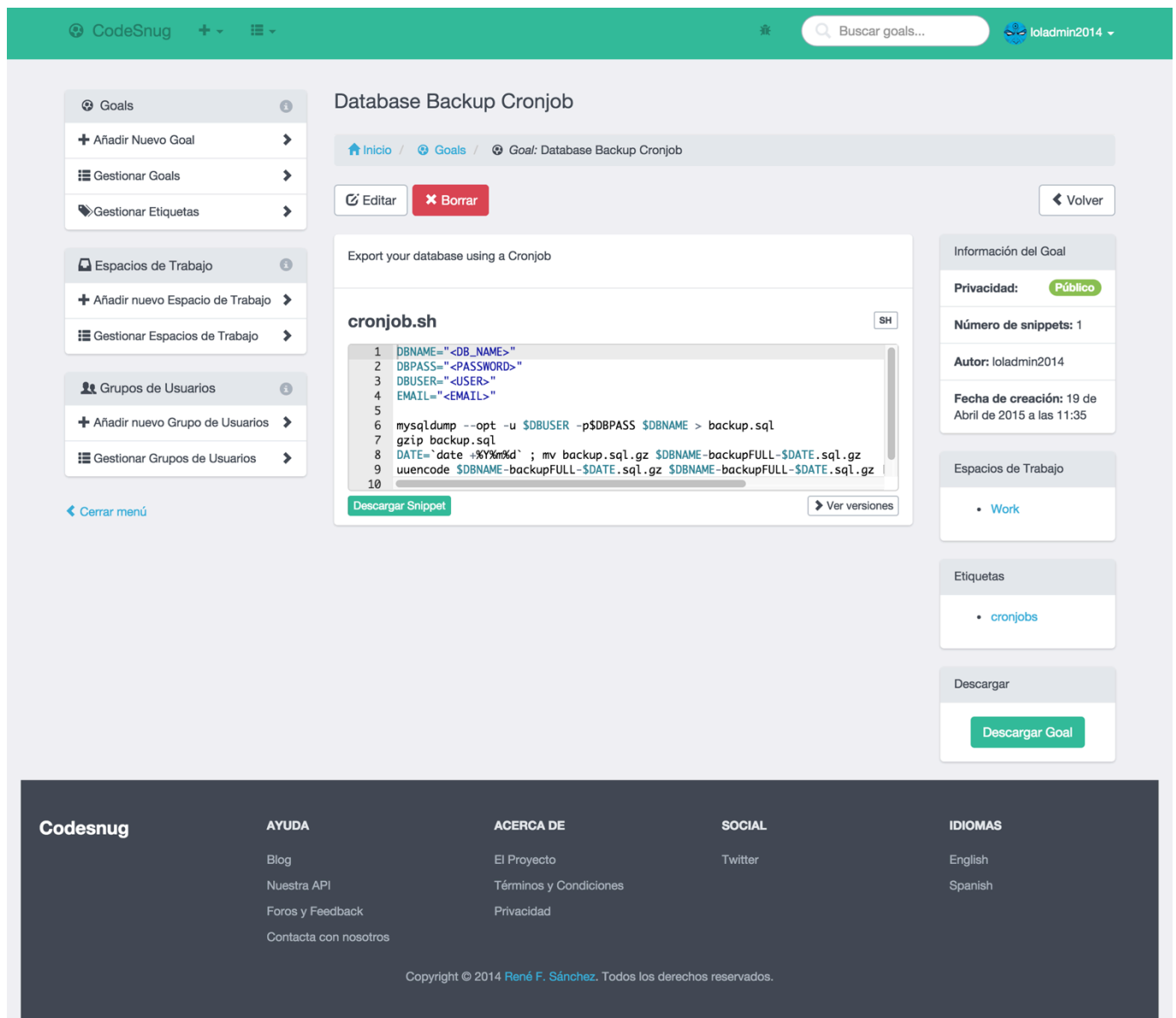


Ilustración 66. Página de detalles de goal

En la página de detalles de goal se puede ver su contenido e información complementaria (número de snippets, fecha de creación, workspaces y tags asignados, etc), descargar el código de sus snippets, ver sus etiquetas, lenguajes de programación y acceder a sus distintas revisiones.

Se permite descargar el código de dos formas diferentes:

- Del goal entero en formato txt.
- De cada snippet individual en formato txt.

### 7.3.6 Página de edición de goal

Ilustración 67. Página de edición de goals

Desde esta página se puede editar todo el contenido de un goal, añadir nuevos snippets, asignar categorías y etiquetas, etc.

En la parte central es donde se incluye la información principal del Goal y sus snippets. A un goal se le pueden añadir tantos snippets como se desee pulsando el botón "Añadir otro snippet", que añadirá nuevos campos al formulario mediante JQuery.

Ilustración 68. Sección de cabecera de Snippet

Los campos de edición de cada snippet permiten establecerle un nombre, un tamaño de fuente temporal (que será útil en el caso de que se desee ver el texto del snippet más grande o más pequeño y un selector de lenguajes de

programación. Cuando se modifique el valor en este selector, se cambiará automáticamente el resaltado del código del campo contenido del snippet correspondiente.

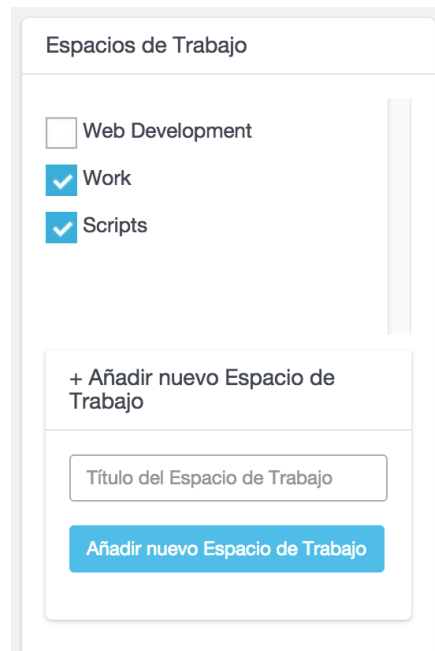


Ilustración 69. Barra lateral derecha de la pantalla de edición de goals

La barra lateral derecha permanece siempre estática, es colapsable mediante JQuery para permitir su correcta visualización en resoluciones pequeñas y está compuesta por tres zonas diferentes: la primera para seleccionar los workspaces a los que pertenecerá el goal, permitiendo añadir nuevos workspaces sin salir de la página mediante AJAX. La segunda funciona de manera similar, pero en este caso será para las etiquetas.

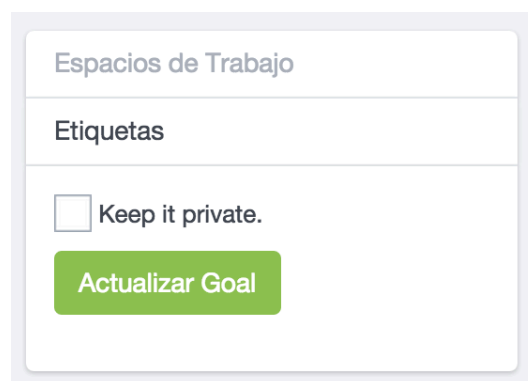


Ilustración 70. Zona inferior de la barra lateral derecha de la pantalla de edición de goals

La última zona permite establecer un goal como privado (solamente el creador del mismo puede verlo, aunque esté asignado a varios workspaces) o público (será visible en todos los workspaces que esté asignado).

Por último, está el botón de actualizar/crear el goal.

### 7.3.7 Página de listado de etiquetas

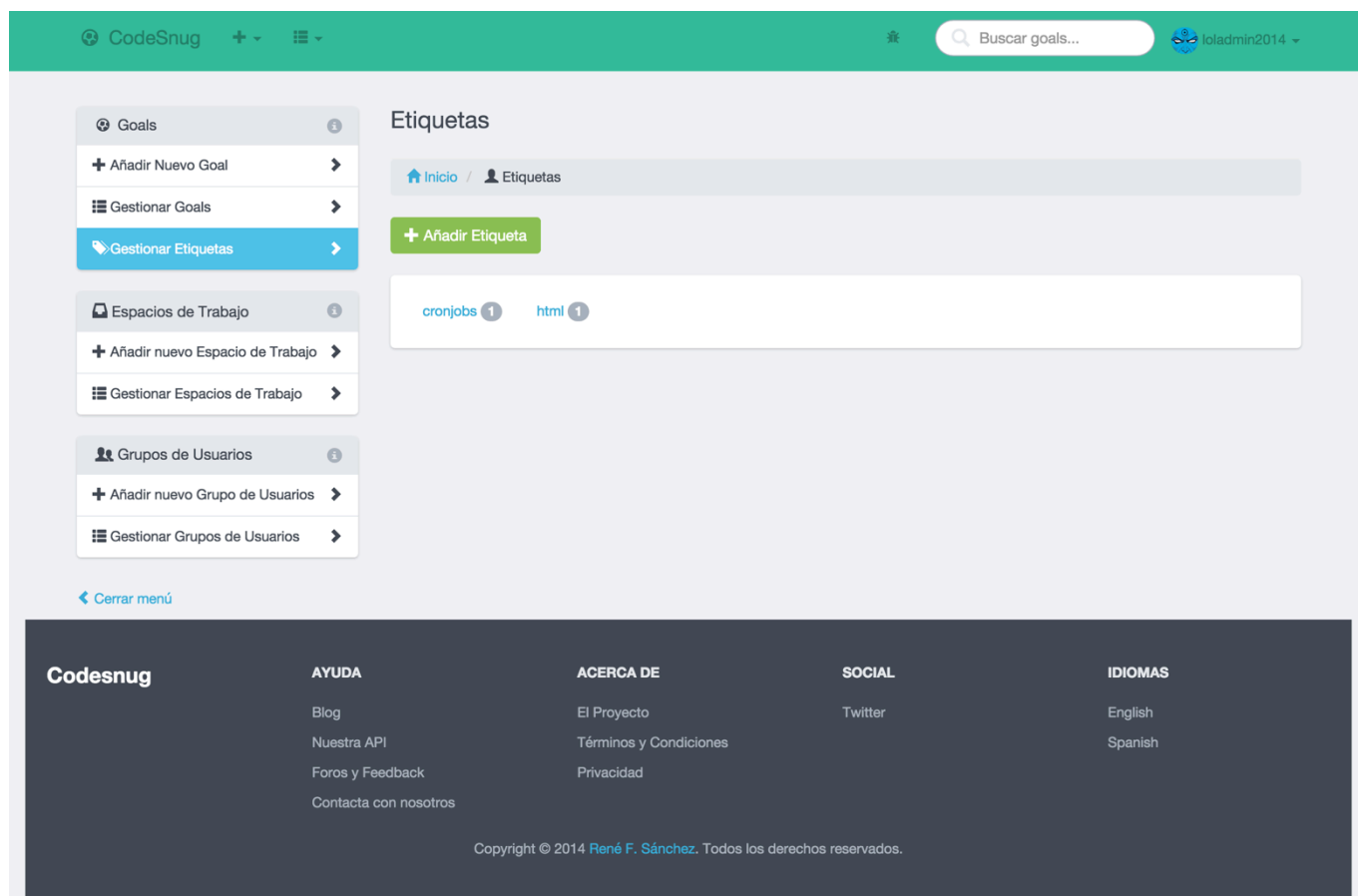


Ilustración 71. Página de listado de etiquetas

En esta página se muestran todas las etiquetas del usuario ordenadas alfabéticamente e indicando el número de goals de cada una. Si se pulsa sobre una de las etiquetas se accederá a su contenido (se mostrará un listado con todos los goals que hay en ella).

### 7.3.8 Página de creación de workspaces

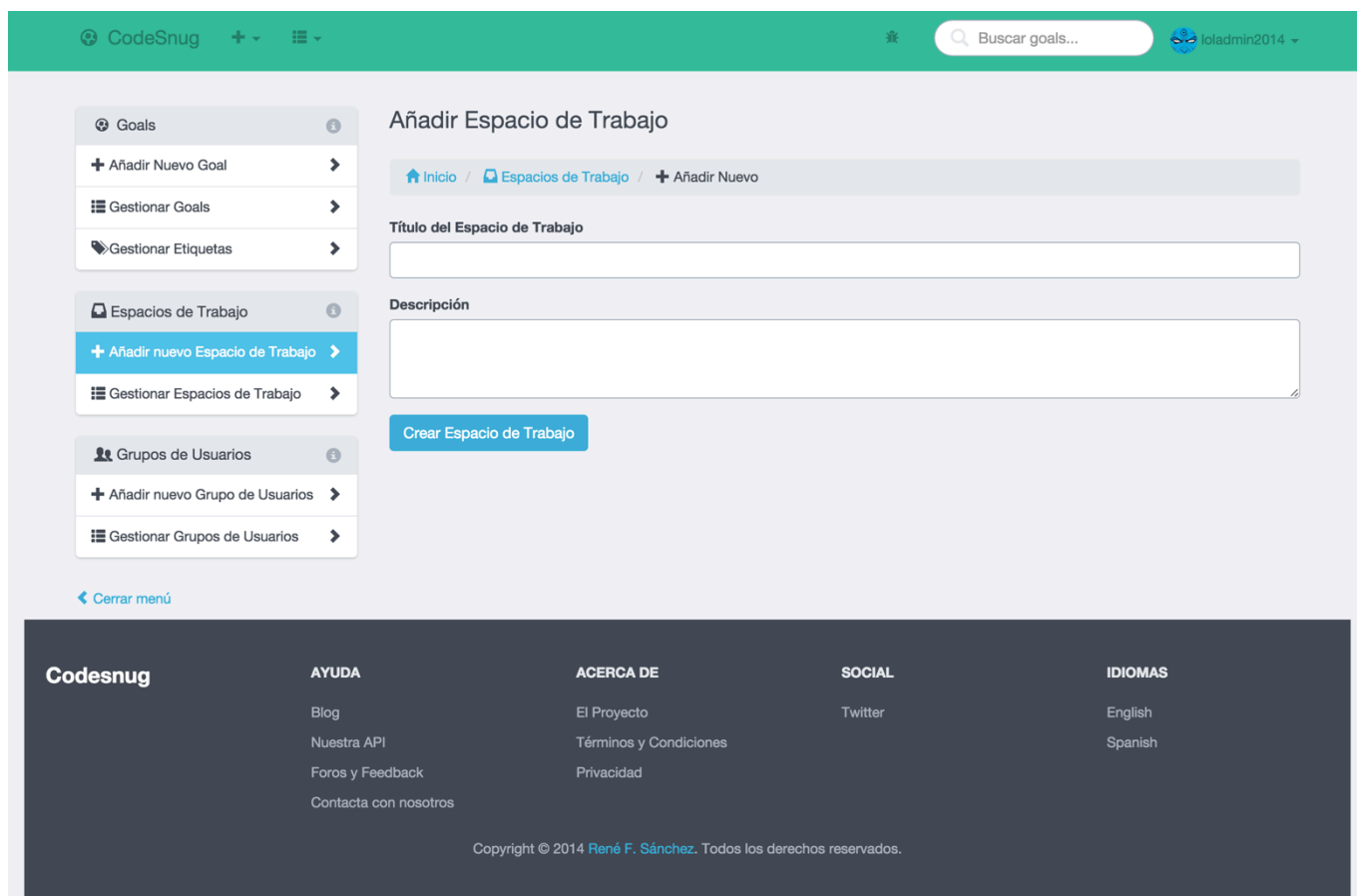


Ilustración 72. Página de creación de workspaces

Esta página muestra el formulario de creación de un nuevo workspace. Es similar a la página de creación de grupos de usuario.

### 7.3.9 Página de listado de workspaces

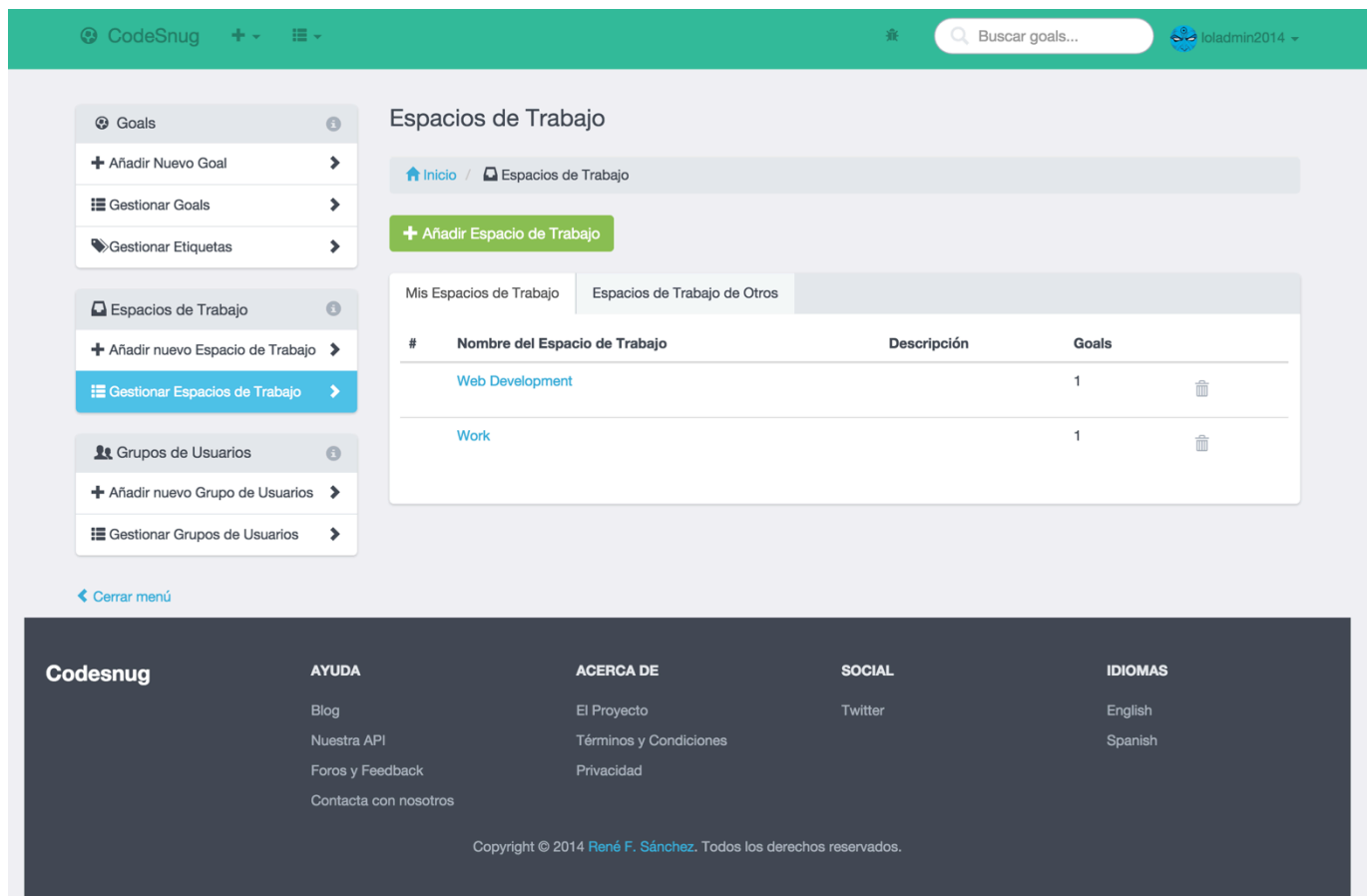


Ilustración 73. Página de listado de workspaces

Esta página tiene dos pestañas. La primera de ellas muestra los workspaces del usuario y la segunda los workspaces de otros usuarios a los que el usuario actual tiene acceso.

En ambos casos se muestra el nombre, la descripción el número de goals en cada workspace y un botón para borrarlo.

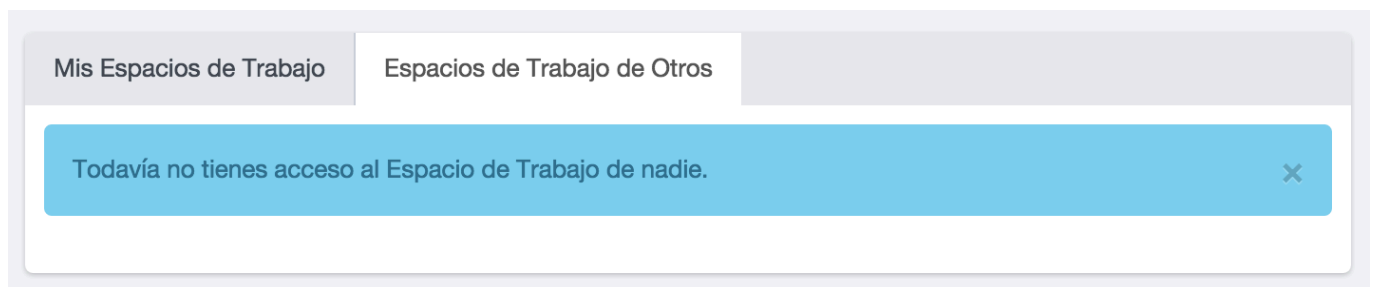


Ilustración 74. Sección de selección de workspaces

En el caso de que no se haya concedido al usuario acceso a ningún espacio de trabajo, se le notificará mediante un mensaje.

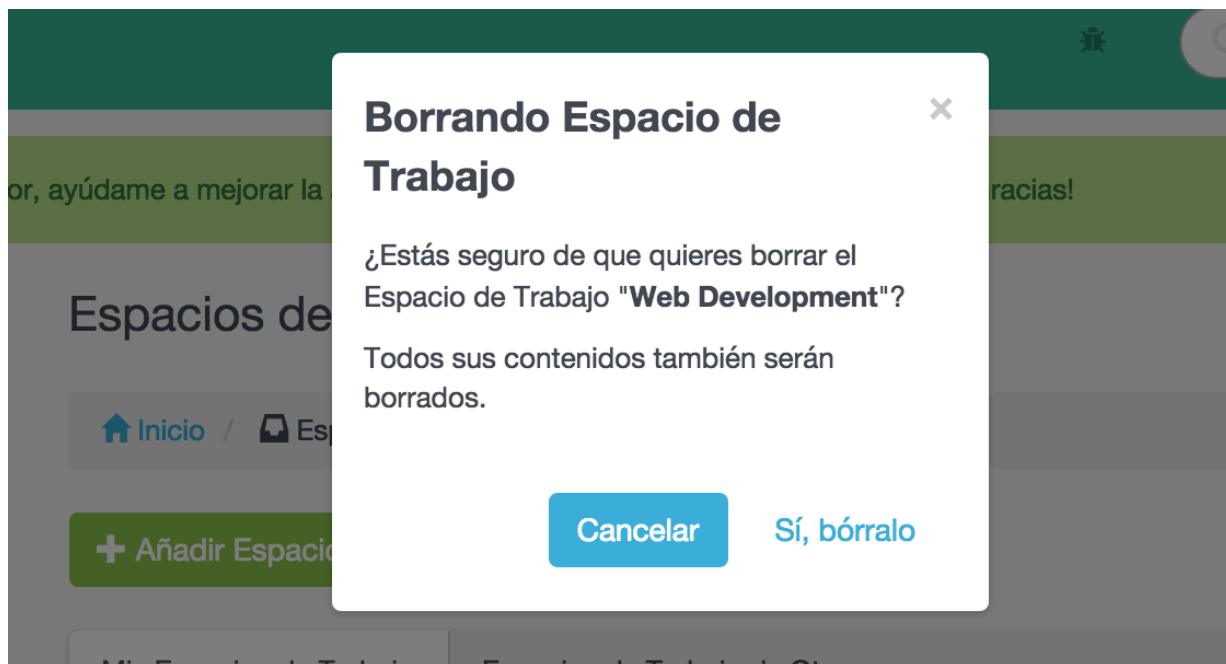


Ilustración 75. Mensaje de confirmación de borrado

En el caso de pulsar el botón de borrado se solicitará la confirmación del usuario mediante un popup realizado con JQuery y Bootstrap. En este popup se le otorga más importancia visual al botón de cancelar que al de borrar para que el usuario tenga menos probabilidades de cometer un error.

### 7.3.10 Página de detalles de grupo de usuarios

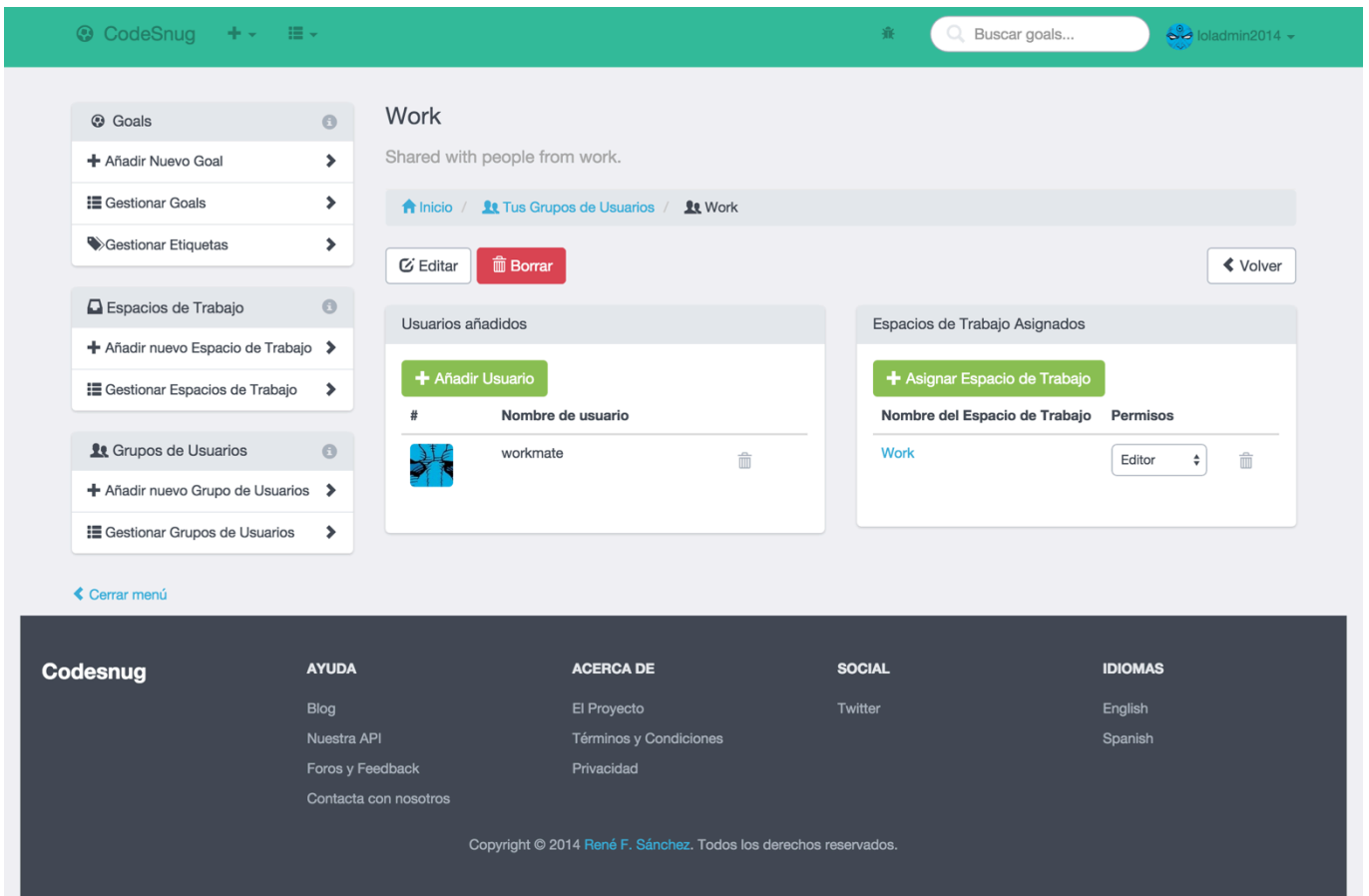


Ilustración 76. Página de detalles de grupo de usuarios

En esta página se muestra la información relevante a los grupos de usuario, como los workspaces asignados y los permisos de los usuarios sobre dichos workspaces.

### 7.3.11 Página de añadir usuario a grupo de usuarios



Ilustración 77. Página para añadir usuarios a un grupo de usuarios



Cuando se desea añadir un usuario a un grupo de usuarios, solamente se muestra un campo para introducir el nombre del usuario. Este campo dispone de autocompletado sobre los nombres de usuario para permitir encontrar al usuario deseado de una manera sencilla, lo que también se consigue mostrando el avatar del usuario en los resultados.

### 7.3.12 Página de listado de grupos de usuarios

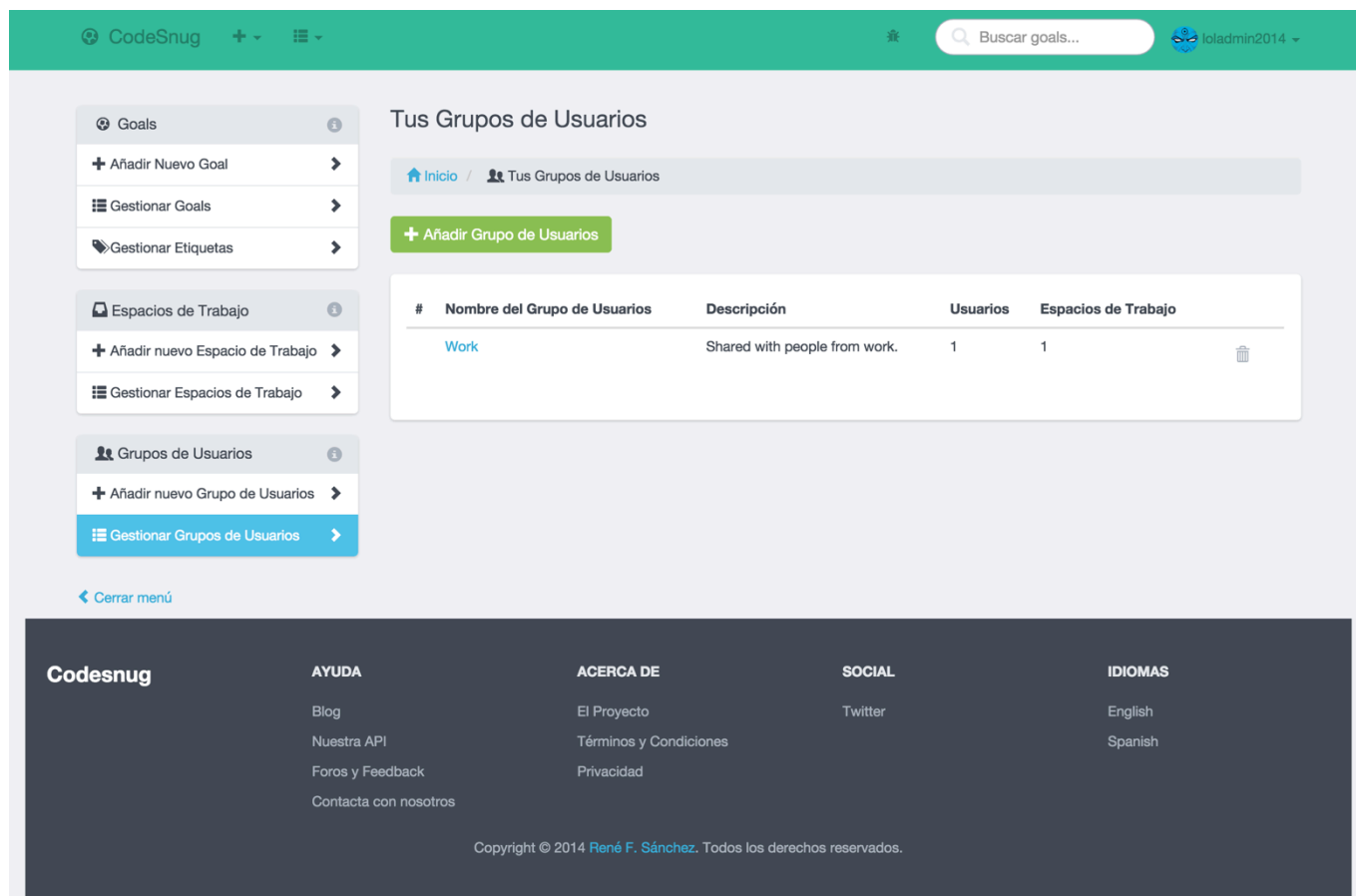


Ilustración 78. Página de listado de grupos de usuarios

En esta página se muestra un listado de todos los grupos de usuario del usuario.

## 7.3.13 Página de perfil de usuario

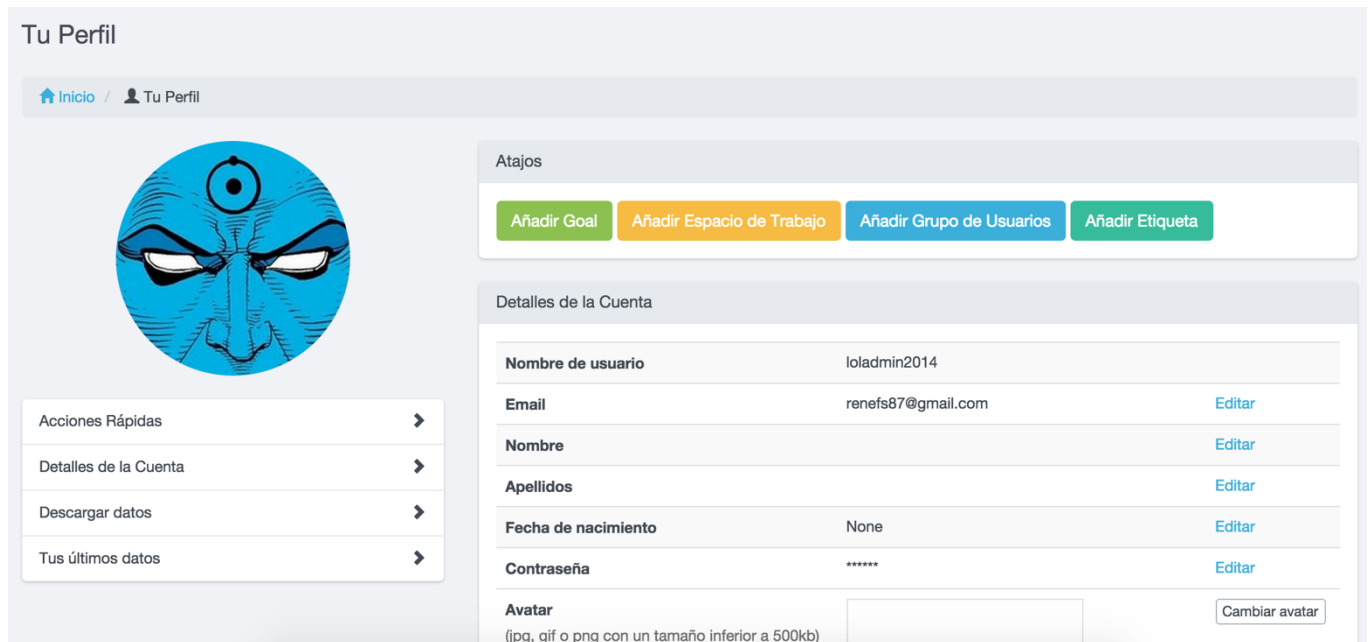


Ilustración 79. Zona de datos de usuario de la página de perfil

La página de perfil de usuario está organizada en varias zonas. En primer lugar, en la zona superior se encuentra una lista de botones con enlaces a las diferentes partes del sitio web (Atajos o acciones rápidas).

Desde el menú izquierdo se puede acceder a las distintas zonas del perfil mediante enlaces “ancla” que llevan a la zona correspondiente.

En la zona “Detalles de la cuenta” se pueden editar los datos del usuario. Para editar estos datos se utilizan peticiones AJAX, de tal forma que el usuario no necesita actualizar la página para ver sus cambios reflejados.

**Contraseña**

Old password	
New password	
Repeat password	

[Cancel](#) [Save](#)

Ilustración 80. Ejemplo de edición de campo

Cuando se pulsa el botón “Editar” de una columna, se muestra un formulario para que el usuario introduzca los nuevos datos.

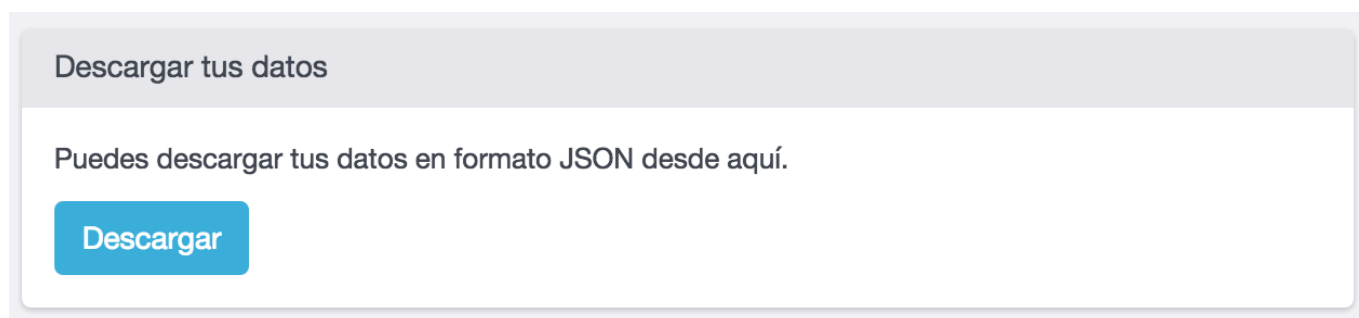


Ilustración 81. Zona de descarga de datos de usuario de la página de perfil

La siguiente zona es la que permite a un usuario descargar todos sus datos en formato JSON para ser procesados por él mismo a su antojo o utilizando una aplicación de terceros.

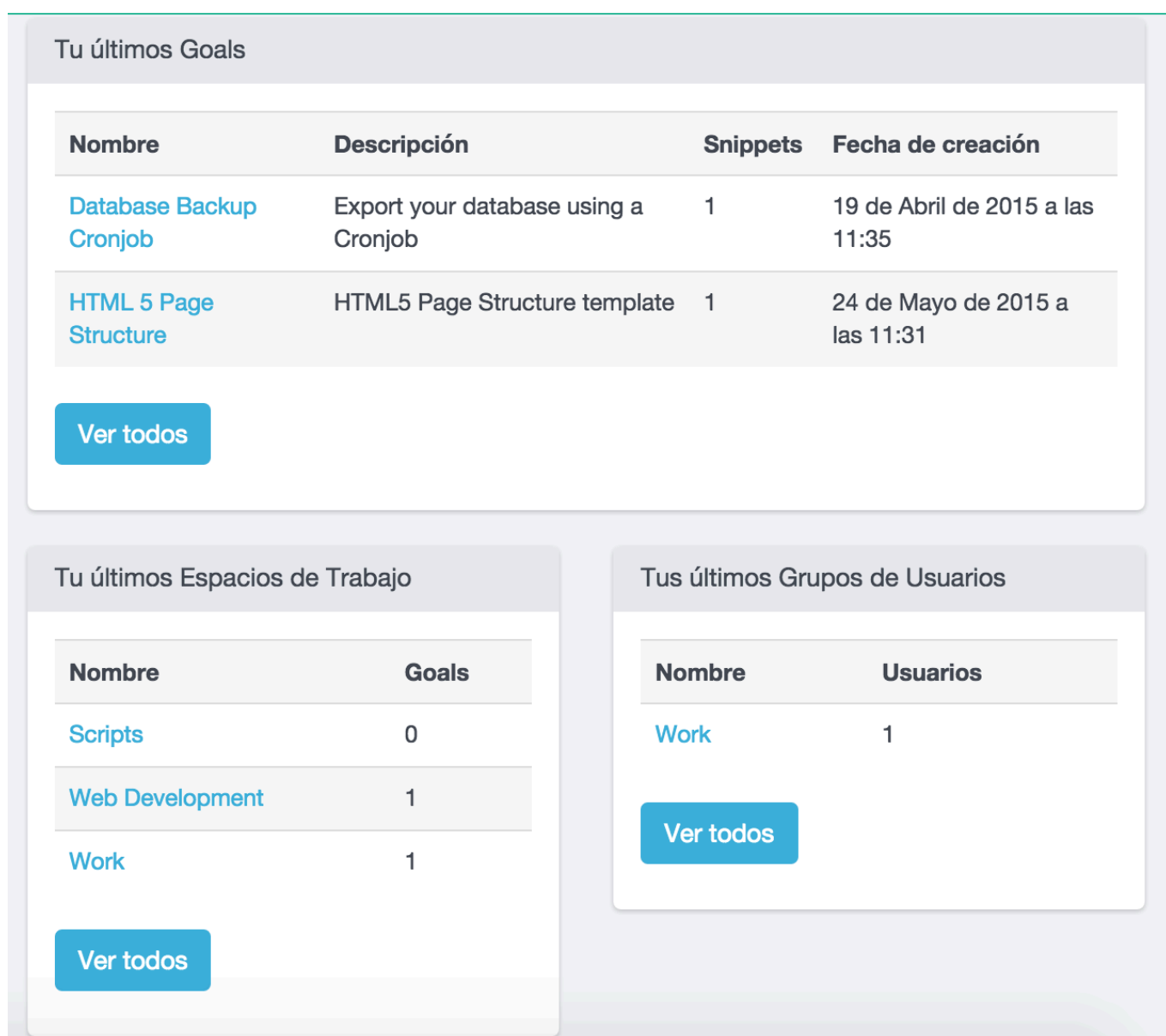


Ilustración 82. Zona de resumen de la página de perfil

Por último, en la última zona puede observarse un resumen de las últimas acciones realizadas por el usuario: últimos goals añadidos, últimos workspaces y grupos de usuarios.

## 7.3.14 Página de texto estándar

CodeSnug + -

Buscar goals... loladmin2014

## Nuestra API

Inicio / La API de Codesnug

### Nuestra API

Codesnug dispone de una API REST de consulta para acceder a los datos de un usuario.

Formatos de retorno de la API:

1. application/json
2. text/html

Entre las operaciones que permite la API se encuentran las siguientes operaciones:

1. **GET /api/goals**: Obtiene la lista de goals del usuario actual.
2. **GET /api/goals/{uuid}**: Obtiene la información del Goal identificado mediante su uuid.
3. **GET /api/workspaces**: Obtiene un listado con todos los Espacios de trabajo del usuario actual.
4. **GET /api/workspaces/{uuid}**: Obtiene la información de un Espacio de trabajo identificado por su uuid.
5. **GET /api/snippets**: Obtiene un listado de los snippets del usuario actual.
6. **GET /api/snippets/{uuid}**: Obtiene la información de un snippet en concreto dado su uuid.
7. **GET /api/tags**: Obtiene un listado de las etiquetas del usuario actual.
8. **GET /api/tags/{uuid}**: Obtiene la información de una etiqueta en concreto dado su uuid.

Todas las peticiones deben incluir las credenciales del usuario, que serán cifradas utilizando SSL.

Ejemplo de petición:

```
https://codesnug.com/api/goals -u user:pass
```

Ejemplo de respuesta:

```
HTTP 200 OK
Content-Type: application/json
Vary: Accept
Allow: GET, HEAD, OPTIONS
{
  "count": 6,
  "next": null,
  "previous": null,
  "results": [ {
    "title": "Goal 1",
    "description": "Maecenas laoreet orci justo, ut sagittis neque varius eu. Nullam risus sapien, cursus id magna vitae, bland
it semper nisi.",
    "uuid": "8b5eb590d0094c529f547ea7f17dcf86",
    "created_at": "2014-07-21T13:18:31.506Z",
    "is_private": false,
    "owner": "nombre_usuario",
    "snippets": [
      "4c46864b9f3c4a89b1aee6bf50f04a5f",
      "572d5029abb54d1fa225f0c44e9fe44d"
    ],
    "workspaces": [
      "Workspace 1",
      "Workspace 2"
    ],
    "tags": [
      "Tag 1",
      "Tag 2"
    ]
  }
],
  ...
}
```

Para hacer pruebas con la información ofrecida por la API, está disponible una interfaz web para realizar las consultas.

[Browsable API](#)

Debes estar registrado para acceder a la API navegable.

<b>Codesnug</b>	<b>AYUDA</b>	<b>ACERCA DE</b>	<b>SOCIAL</b>	<b>IDIOMAS</b>
	<a href="#">Blog</a>	<a href="#">El Proyecto</a>	<a href="#">Twitter</a>	<a href="#">English</a>
	<a href="#">Nuestra API</a>	<a href="#">Términos y Condiciones</a>		<a href="#">Spanish</a>
	<a href="#">Foros y Feedback</a>	<a href="#">Privacidad</a>		
	<a href="#">Contacta con nosotros</a>			

Copyright © 2014 René F. Sánchez. Todos los derechos reservados.

Ilustración 83. Ejemplo de página de texto

Página de texto genérica de un solo bloque. Similar a las páginas de política de privacidad, términos y condiciones, etc.

## 7.4 Funcionalidades de la API

Codesnug dispone de una API REST de consulta para acceder a los datos de un usuario.

Formatos de retorno de la API:

- application/json
- text/html

Entre las operaciones que permite la API se encuentran las siguientes operaciones:

- **GET /api/goals**: Obtiene la lista de goals del usuario actual.
- **GET /api/goals/{uuid}**: Obtiene la información del Goal identificado mediante su uuid.
- **GET /api/workspaces**: Obtiene un listado con todos los Espacios de trabajo del usuario actual.
- **GET /api/workspaces/{uuid}**: Obtiene la información de un Espacio de trabajo identificado por su uuid.
- **GET /api/snippets**: Obtiene un listado de los snippets del usuario actual.
- **GET /api/snippets/{uuid}**: Obtiene la información de un snippet en concreto dado su uuid.
- **GET /api/tags**: Obtiene un listado de las etiquetas del usuario actual.
- **GET /api/tags/{uuid}**: Obtiene la información de una etiqueta en concreto dado su uuid.

Todas las peticiones deben incluir las credenciales del usuario, que serán cifradas utilizando SSL.

Ejemplo de petición:

```
https://codesnug.com/api/goals -u user:pass
```

Ejemplo de respuesta:

```
HTTP 200 OK
Content-Type: application/json
Vary: Accept
Allow: GET, HEAD, OPTIONS {
  "count": 6,
  "next": null,
  "previous": null,
  "results": [ {
    "title": "Goal 1",
    "description": "Maecenas laoreet orci justo, ut sagittis neque varius eu. Nullam
    risus sapien, cursus id magna vitae, blandit semper nisi.",
    "uuid": "8b5eb590d0094c529f547ea7f17dcf86",
    "created_at": "2014-07-21T13:18:31.506Z",
    "is_private": false,
    "owner": "nombre_usuario",
    "snippets": [
      "4c46864b9f3c4a89b1aee6bf50f04a5f",
```

```

    "572d5029abb54d1fa225f0c44e9fe44d"
  ],
  "workspaces": [
    "Workspace 1",
    "Workspace 2"],
  "tags": [
    "Tag 1",
    "Tag 2"
  ]}, ...

```

### 7.4.1 API Navegable

The screenshot shows the CodeSnug REST API v1.0 interface. The breadcrumb navigation is 'Api Root > / Snippet > / Snippet'. The method is 'GET' and the endpoint is '/api/snippets/89d6f7c964784fc180fd94592bf939b3'. The response is 'HTTP 200 OK' with headers: 'Content-Type: application/json', 'Vary: Accept', and 'Allow: GET, HEAD, OPTIONS'. The response body is a JSON object:

```

{
  "title": "cronjob.sh",
  "goal": "071197e2b3b34e268a0e377bff57f9d8",
  "author": "loladmin2014",
  "text": "DBNAME=\"<DB_NAME>\"\\r\\nDBPASS=\"<PASSWORD>\"\\r\\nDBUSER=\"<USER>\"\\r\\nEMAIL=\"<EMAIL>\"\\r\\n \\r\\nmysqldump --opt -u $DBUSER -p$DBPASS $DBNAME > backup.sql\\r\\ngzip backup.sql\\r\\nDATE=`date +%Y%m%d` ; mv backup.sql.gz $DBNAME-backupFULL-$DATE.sql.gz\\r\\nuuencode $DBNAME-backupFULL-$DATE.sql.gz $DBNAME-backupFULL-$DATE.sql.gz | mail -s \"<SUBJECT>\" $EMAIL\\r\\nrm $DBNAME-backupFULL-$DATE.sql.gz",
  "created_at": "2015-04-19T09:36:01.517Z",
  "uuid": "89d6f7c964784fc180fd94592bf939b3",
  "language": "sh",
  "version_number": 1,
  "versions": []
}

```

Ilustración 84. Página de la API navegable

Para facilitar la tarea de los desarrolladores que deseen extender Codesnug o crear aplicaciones a partir de ella, Codesnug dispone también de una API navegable accesible desde el navegador web, que permite probar la respuesta de la API a los distintos métodos de la misma sin necesidad de ningún plugin en el navegador ni programa externo.

## 7.5 Especificación técnica del plan de pruebas

### 7.5.1 Pruebas unitarias

Para la creación de un set de pruebas se ha utilizado el módulo incluido dentro del framework Django. Este módulo permite realizar pruebas de casos de uso y también simular un navegador web para realizar pruebas de las distintas vistas de la aplicación.

#### 7.5.1.1 Pruebas de usuarios

Método	Descripción	Resultado esperado	Resultado obtenido
<b>test_add_user</b>	Prueba que simula la creación e inclusión de un usuario en el sistema.	El número de usuarios añadidos es 2.	Correcto
<b>test_add_user_with_error</b>	Simula la creación de usuarios erróneos en el sistema. Intenta crear usuarios con identificadores repetidos.	Se lanza una excepción por cada uno de los usuarios que se intenta añadir. Los usuarios no son añadidos.	Correcto
<b>test_remove_user</b>	Borra usuarios del sistema y comprueba que el borrado ha tenido éxito.	Disminuye el número de usuarios por cada uno de los usuarios borrados.	Correcto
<b>test_check_user_data</b>	Comprueba que tras la creación de un usuario, todos sus datos son correctos y no dispone de permisos de administrador por defecto.	Todos los datos coinciden con los establecidos y esperados.	Correcto
<b>test_add_user_group</b>	Verifica que los grupos de usuarios se crean correctamente en el sistema.	El número de grupos de usuarios en el sistema es el esperado	Correcto
<b>test_add_same_group_twice</b>	Intenta añadir un grupo de usuarios con datos duplicados.	Debe lanzar una excepción y el grupo de usuarios no debe ser añadido.	Correcto
<b>test_add_same_group_different_users</b>	Crea dos grupos de usuarios con la misma información y los asigna a diferentes usuarios.	Cada usuario debe disponer de su propio grupo y ambos deben existir en el sistema	Correcto.
<b>test_remove_user_group</b>	Añade y elimina varios grupos de usuarios del sistema.	El número de grupos de usuarios en el sistema deben disminuir al ser borrados los grupos de usuarios.	Correcto
<b>test_user_has_group</b>	Comprueba si un grupo de usuarios pertenece a un usuario.	El grupo debe pertenecer al usuario	Correcto



## 7.5.1.2 Pruebas de goals

Método	Descripción	Resultado esperado	Resultado obtenido
<b>test_tags_add</b>	Verifica que los tags pueden ser añadidos correctamente.	El número de tags añadidos es correcto y los tags son asignados a a los usuarios correspondientes.	Correcto
<b>test_user_has_no_tags</b>	Comprueba que un usuario no tiene tags después de su creación.	El número de tags es cero.	Correcto
<b>test_tags_with_same_name</b>	Comprueba que se produce un error de integridad cuando se intenta añadir un tag con el mismo nombre dos veces.	Se produce un error.	Correcto
<b>test_tags_remove</b>	Verifica que los tags pueden ser borrados correctamente tanto del sistema como de sus usuarios.	El tag deja de pertenecer al usuario y ya no se encuentra en el sistema.	Correcto
<b>test_workspace_add</b>	Verficia que los workspaces pueden ser añadidos correctamente al sistema y asignados a usuarios.	El número de workspaces en el sistema es correcto, al igual que el número de workspaces del usuario.	Correcto
<b>test_user_has_no_workspaces</b>	Comprueba que un usuario no dispone de workspaces tras su creación.	El número de workspaces es cero.	Correcto
<b>test_workspaces_with_same_name</b>	Verifica que no se pueden crear dos workspaces con el mismo nombre para el mismo usuario.	Debe producirse un error al intentar añadir el segundo workspace.	Correcto
<b>test_workspaces_remove</b>	Comprueba que los workspaces pueden ser borrados correctamente.	Los workspaces son borrados del sistema y del usuario correspondiente.	Correcto

## 7.5.1.3 Pruebas de vistas y permisos

Método	Descripción	Resultado esperado	Resultado obtenido
<b>test_add_goal_admin_error</b>	Trata de añadir un goal con datos erróneos.	Debe devolverse una respuesta 404.	Correcto
<b>test_add_goal_admin</b>	Trata de añadir un goal con datos correctos.	Debe añadir el goal, realizar las redirecciones esperadas y devolver una respuesta correcta (código 200)	Correcto
<b>test_add_goal_guest_error</b>	Comprueba que un usuario no registrado no puede añadir goals.	El goal no es añadido al sistema.	Correcto
<b>test_remove_goal_owner</b>	Comprueba que el dueño de unos goals es capaz de borrarlos.	Las redirecciones son las esperadas, los goals son borrados del sistema y los códigos de respuesta son los esperados.	Correcto
<b>test_remove_goal_not_owner</b>	Comprueba que solamente el dueño de un goal es capaz de eliminarlo.	El goal no debe ser borrado del sistema y la página debe devolver las respuestas esperadas.	Correcto
<b>test_workspace_access_admin</b>	Comprueba que el creador de un workspace puede acceder a su página de detalles.	El código de respuesta al acceder a la página es 200.	Correcto
<b>test_workspace_access_suscriber</b>	Comprueba que un usuario con permisos de suscriptor es capaz de acceder a un workspace al que se le ha proporcionado acceso.	El código de respuesta de la página es 200.	Correcto
<b>test_workspace_access_editor</b>	Comprueba que un usuario con permisos de editor es capaz de acceder a un workspace al que se le ha proporcionado acceso.	El código de respuesta de la página es 200.	Correcto
<b>test_workspace_access_none</b>	Comprueba que un usuario sin registrar no es capaz de acceder a ningún workspace.	El código de respuesta de la página es un 404.	Correcto
<b>test_workspace_access_guest</b>	Comprueba que un usuario registrado que no dispone de acceso a un workspace no puede acceder a él.	El código de respuesta es 302 y el usuario es redirigido.	Correcto
<b>test_workspace_update_admin</b>	Comprueba que un usuario con permisos	Los códigos de respuesta son los esperados, las	Correcto

	de administrador es capaz de editar sus propios workspaces.	redirecciones son las esperadas, los mensajes en pantalla son los esperados y el workspace es actualizado	
<b>test_workspace_update_subscriber</b>	Verifica que un usuario con acceso a un workspace y permisos de suscriptor no es capaz de editar un workspace	El workspace no es actualizado ni por petición post ni get y el código de respuesta es 404.	Correcto
<b>test_workspace_update_editor</b>	Comprueba que un usuario con acceso a un workspace y permisos de editor no puede editar los datos del workspace.	Tanto mediante petición post como get el código de respuesta es 404.	Correcto
<b>test_workspace_update_none</b>	Verifica que un usuario no registrado no puede editar los datos de un workspace.	Las respuestas son siempre 404 y el workspace no es actualizado.	Correcto
<b>test_workspace_update_guest</b>	Comprueba que un usuario registrado no puede editar un workspace al que no tiene acceso.	El usuario es redirigido y no se muestran mensajes de error ni de confirmación.	Correcto
<b>test_workspace_delete_admin_get</b>	Comprueba que un workspace no puede ser borrado utilizando un método GET.	La respuesta devuelta debe ser un código 405 (método no permitido).	Correcto
<b>test_workspace_delete_admin_post</b>	Comprueba que un workspace puede ser borrado por su administrador utilizando un método POST.	El workspace debe ser borrado, las redirecciones deben ser las esperadas, los códigos de respuesta deben ser correctos y los mensajes en pantalla deben mostrar éxito en la operación.	Correcto
<b>test_workspace_delete_subscriber_get</b>	Comprueba que un workspace no puede ser borrado utilizando un método GET por un suscriptor.	La respuesta devuelta debe ser un código 405 (método no permitido).	Correcto
<b>test_workspace_delete_subscriber_post</b>	Comprueba que un workspace no puede ser borrado utilizando un método POST por un suscriptor.	El workspace no debe ser borrado y el código de respuesta debe ser 404.	Correcto
<b>test_workspace_delete_editor_get</b>	Comprueba que un workspace no puede ser borrado utilizando un método GET por un suscriptor.	La respuesta devuelta debe ser un código 405 (método no permitido).	Correcto

<b>test_workspace_delete_editor_post</b>	Comprueba que un workspace no puede ser borrado utilizando un método POST por un editor.	El workspace no debe ser borrado y el código de respuesta debe ser 404.	Correcto
<b>test_workspace_delete_none_get</b>	Comprueba que un workspace no puede ser borrado utilizando un método GET por un usuario no registrado.	La respuesta devuelta debe ser un código 405 (método no permitido).	Correcto
<b>test_workspace_delete_none_post</b>	Comprueba que un workspace no puede ser borrado utilizando un método POST por un usuario no registrado.	El workspace no debe ser borrado y el código de respuesta debe ser 404.	Correcto
<b>test_workspace_delete_guest_get</b>	Comprueba que un workspace no puede ser borrado utilizando un método GET por un usuario registrado.	La respuesta devuelta debe ser un código 405 (método no permitido).	Correcto
<b>test_workspace_delete_guest_post</b>	Comprueba que un workspace no puede ser borrado utilizando un método POST por un usuario registrado.	El workspace no debe ser borrado y el código de respuesta debe ser 302 para redirigir al usuario.	Correcto
<b>test_add_goal_admin</b>	Comprueba que un goal puede ser añadido por un administrador	El goal es añadido correctamente, se muestran los mensajes de éxito por pantalla y se redirige al usuario.	Correcto
<b>test_update_goal_admin</b>	Comprueba que un goal puede ser actualizado por un administrador.	El goal es actualizado correctamente, se muestran los mensajes de éxito por pantalla y se redirige al usuario.	Correcto

#### 7.5.1.4 Pruebas de vistas

Método	Descripción	Resultado esperado	Resultado obtenido
<b>test_home</b>	Comprueba que la página de inicio es accesible a través del servidor	Debe ser accesible y devolver una respuesta 200.	Correcto
<b>test_home_register</b>	Comprueba que el formulario de registro de la página de inicio funciona y redirige correctamente a la página de conversación.	Debe redirigir correctamente a la página de confirmación	Correcto

Método	Descripción	Resultado esperado	Resultado obtenido
<b>test_login</b>	Comprueba que la página de login es accesible y funciona correctamente.	Las respuestas y redirecciones deben ser correctas, el usuario no se debe crear como activo.	Correcto
<b>test_home_register_error</b>	Verifica que se produce un error cuando el usuario introduce campos erróneos en el formulario de registro.	Debe devolverse al usuario a la misma página en la que se encontraba.	Correcto
<b>test_add_workspace_to_group</b>	Verifica si los resultados son los esperados al asignar un workspace a un grupo de usuarios.	El usuario es añadido correctamente. No se muestran mensajes de error en pantalla y se muestra un mensaje de confirmación	Correcto
<b>test_remove_workspace_from_group</b>	Verifica si los resultados son los esperados al quitar un workspace de un grupo de usuarios.	Se muestran mensajes de confirmación, no se muestran errores, el número de workspaces que el grupo tiene asignado disminuye.	Correcto
<b>test_add_user_to_group</b>	Verifica si un usuario es añadido correctamente a un grupo de usuarios.	Muestra en pantalla los mensajes esperados, realiza las redirecciones esperadas y el número de usuarios del grupo de usuarios aumenta.	Correcto
<b>test_remove_user_from_group</b>	Verifica que un usuario es eliminado correctamente de un grupo de usuarios.	Muestra en pantalla los mensajes esperados, realiza las redirecciones correctas, el número de usuarios del grupo de usuarios es el correcto.	Correcto
<b>test_get_usergroup_list</b>	Comprueba que la vista de listado de grupos de usuarios es accesible.	La página es accesible.	Correcto
<b>test_usergroup_add</b>	Verifica que la página de creación de grupo de usuario funciona correctamente.	Permite añadir grupos de usuarios, la respuesta de la página es la esperada, las redirecciones son correctas.	Correcto
<b>test_usergroup_remove_success</b>	Verifica que un grupo de usuarios puede ser borrado correctamente.	Los mensajes mostrados en pantalla son los esperados, las respuestas son correctas y los grupos de usuarios son borrados.	Correcto
<b>test_usergroup_remove_error</b>	Comprueba que las respuestas son las esperadas cuando se trata de borrar un grupo de usuarios que no existe.	La respuesta del sitio debe devolver un error 404.	Correcto
<b>test_usergroup_update</b>	Verifica que un grupo de usuarios puede ser actualizado correctamente y puede ser encontrado por sus nuevos datos.	Los datos son modificados correctamente.	Correcto

## 7.5.2 Pruebas de integración y del sistema

### 7.5.2.1 Cuestionario

♥ Thank you for using Codesnug. Please, help me to improve the application by answering this anonymous survey. Thank you!

Ilustración 85. Mensaje de solicitud de participación en la encuesta

Para la evaluación de la aplicación por parte de los usuarios se ha creado un formulario utilizando Google Docs para permitir a aquellos usuarios que lo deseen, transmitir sus opiniones sobre la aplicación y determinar cuales son sus puntos fuertes y débiles.

A continuación se indican las preguntas del formulario con sus posibles respuestas. Aquellas respuestas abiertas (es decir, que permiten al usuario introducir el texto que desea, están indicadas con un guion).

Pregunta	Respuestas
For how long have you been programming?	<ol style="list-style-type: none"> <li>1. Less than a year</li> <li>2. From 2 to 5 years</li> <li>3. From 5 to 10 years</li> <li>4. More than 10 years</li> </ol>
How do you usually store your code snippets?	<ol style="list-style-type: none"> <li>1. Text files</li> <li>2. Evernote</li> <li>3. Github Gists</li> <li>4. Other</li> </ol>
Do you use any tool to keep your code synced between your devices?	<ol style="list-style-type: none"> <li>1. Yes</li> <li>2. No, I don't need it</li> <li>3. No, there is anything I like</li> </ol>
Does this app seem useful to you?	<ol style="list-style-type: none"> <li>1. A lot</li> <li>2. Something</li> <li>3. A bit</li> <li>4. Nothing</li> </ol>
What is the feature you liked the most?	<ol style="list-style-type: none"> <li>1. Code storage</li> <li>2. Sharing code with other users</li> <li>3. UI</li> <li>4. Code Export</li> <li>5. Search</li> <li>6. REST API</li> </ol>

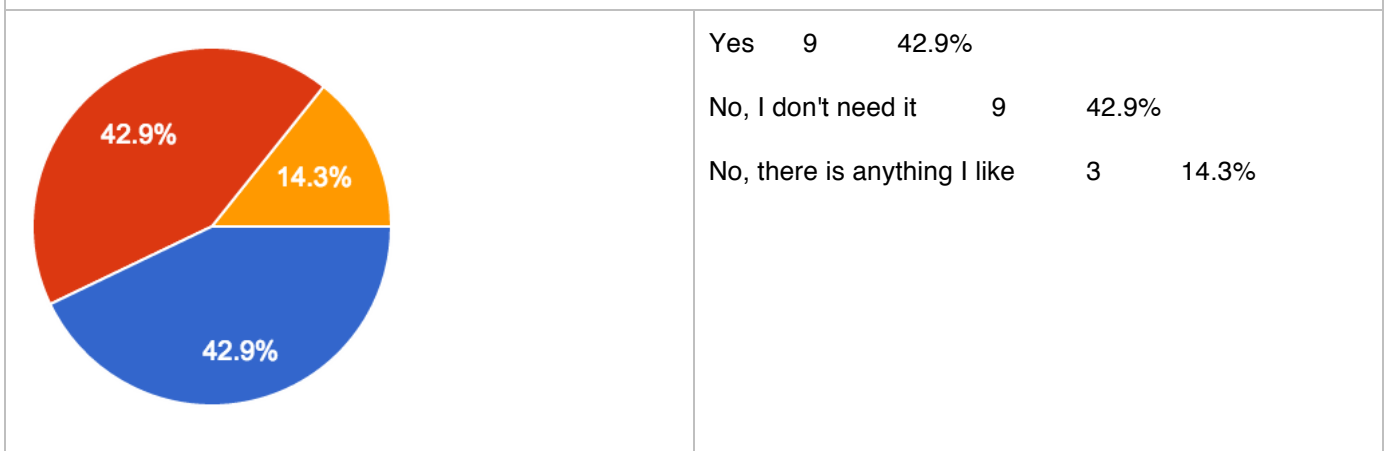
Pregunta	Respuestas
What is the feature you liked the less?	<ol style="list-style-type: none"> <li>Code Storage</li> <li>Sharing code with other users</li> <li>UI</li> <li>Code export</li> <li>Search</li> <li>REST API</li> </ol>
What features would you like to be added to the app?	-
Would you use the app to manage your code snippets?	<ol style="list-style-type: none"> <li>Yes</li> <li>No</li> </ol>
If you would not use the app, can you tell a few words telling why?	-
Any other thoughts about the app?	-

#### 7.5.2.1.1 Resultados de la encuesta

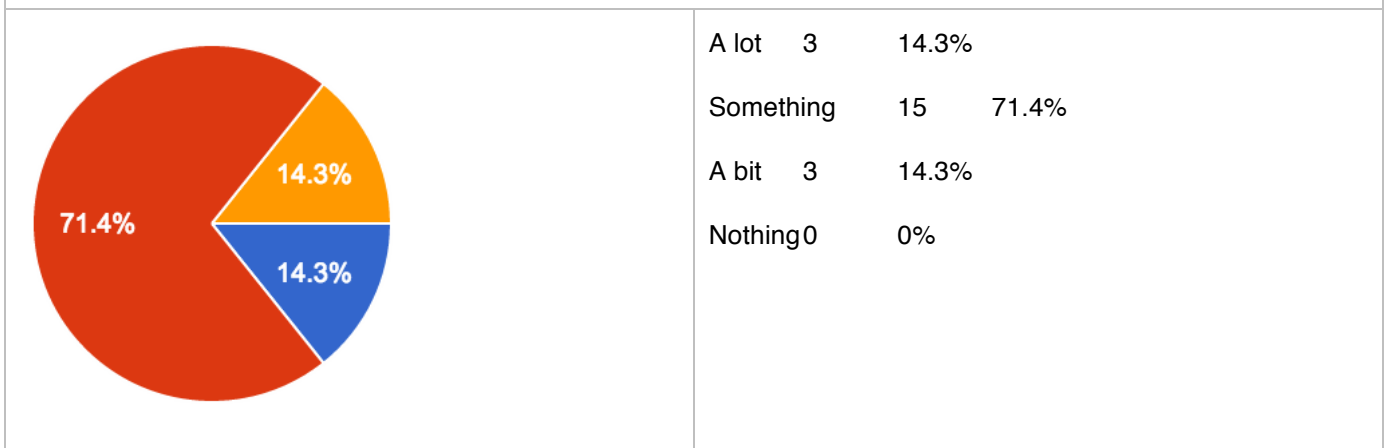
For how long have you been programming?													
	<table border="1"> <tbody> <tr> <td>Less than a year</td> <td>6</td> <td>28.6%</td> </tr> <tr> <td>From 2 to 5 years</td> <td>6</td> <td>28.6%</td> </tr> <tr> <td>From 5 to 10 years</td> <td>6</td> <td>28.6%</td> </tr> <tr> <td>More than 10 years</td> <td>3</td> <td>14.3%</td> </tr> </tbody> </table>	Less than a year	6	28.6%	From 2 to 5 years	6	28.6%	From 5 to 10 years	6	28.6%	More than 10 years	3	14.3%
Less than a year	6	28.6%											
From 2 to 5 years	6	28.6%											
From 5 to 10 years	6	28.6%											
More than 10 years	3	14.3%											
How do you usually store your code snippets?													



**Do you use any tool to keep your code synced between your devices?**

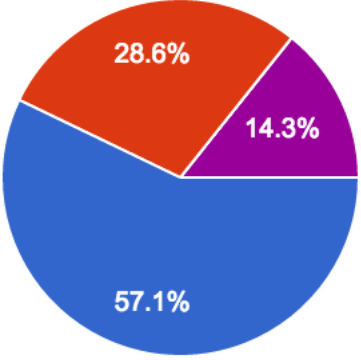
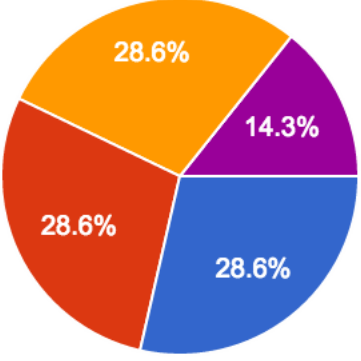



**Does this app seem useful to you?**



**What is the feature you liked the most?**



	<table border="0"> <tr> <td>Code storage</td> <td>12</td> <td>57.1%</td> </tr> <tr> <td>Sharing code with other users</td> <td>6</td> <td>28.6%</td> </tr> <tr> <td>UI</td> <td>0</td> <td>0%</td> </tr> <tr> <td>Code export</td> <td>0</td> <td>0%</td> </tr> <tr> <td>Search</td> <td>3</td> <td>14.3%</td> </tr> <tr> <td>REST API</td> <td>0</td> <td>0%</td> </tr> </table>	Code storage	12	57.1%	Sharing code with other users	6	28.6%	UI	0	0%	Code export	0	0%	Search	3	14.3%	REST API	0	0%
Code storage	12	57.1%																	
Sharing code with other users	6	28.6%																	
UI	0	0%																	
Code export	0	0%																	
Search	3	14.3%																	
REST API	0	0%																	
<p><b>What is the feature you liked the less?</b></p>																			
	<table border="0"> <tr> <td>REST API</td> <td>6</td> <td>28.6%</td> </tr> <tr> <td>Sharing code with other users</td> <td>6</td> <td>28.6%</td> </tr> <tr> <td>UI</td> <td>6</td> <td>28.6%</td> </tr> <tr> <td>Code export</td> <td>0</td> <td>0%</td> </tr> <tr> <td>Search</td> <td>3</td> <td>14.3%</td> </tr> <tr> <td>Code storage</td> <td>0</td> <td>0%</td> </tr> </table>	REST API	6	28.6%	Sharing code with other users	6	28.6%	UI	6	28.6%	Code export	0	0%	Search	3	14.3%	Code storage	0	0%
REST API	6	28.6%																	
Sharing code with other users	6	28.6%																	
UI	6	28.6%																	
Code export	0	0%																	
Search	3	14.3%																	
Code storage	0	0%																	
<p><b>What features would you like to be added to the app?</b></p>	<p>-</p>																		
<p><b>Would you use the app to manage your code snippets?</b></p>																			
	<table border="0"> <tr> <td>Yes</td> <td>21</td> <td>100%</td> </tr> <tr> <td>No</td> <td>0</td> <td>0%</td> </tr> </table>	Yes	21	100%	No	0	0%												
Yes	21	100%																	
No	0	0%																	
<p><b>If you would not use the app, can you tell a few words telling why?</b></p>	<p>-</p>																		
<p><b>Any other thoughts about the app?</b></p>	<p>- Develop Google Sign in</p>																		

## 8 Implementación del sistema

### 8.1 Lenguajes de programación y tecnologías usadas

#### 8.1.1 Python

Python es un lenguaje de programación interpretado cuyo principal objetivo es ofrecer un código legible. Se trata de un lenguaje de programación multiplataforma que ofrece orientación a objetos, programación imperativa y funcional.

Los administradores de Python son la Python Software Foundation y posee una licencia de código abierto.

##### 8.1.1.1 Utilización en el proyecto

Python será el lenguaje de programación principal en el que estará desarrollado el proyecto en el lado del servidor.

#### 8.1.2 Javascript

Javascript es un lenguaje de programación de código libre comúnmente utilizado en el lado del cliente (aunque también existen herramientas y lenguajes de programación para ser utilizados en el lado del servidor, como NodeJS). Javascript se utiliza normalmente para interactuar con el código HTML de los sitios web para generar estructuras y partes de las páginas dinámicas.

##### 8.1.2.1 Utilización en el proyecto

En el lado del cliente, la aplicación necesita efectuar ciertas operaciones, como por ejemplo la generación de los editores de texto para los snippets. También se utiliza AJAX para realizar ciertas operaciones (creación de workspaces, etiquetas, búsqueda de usuarios con autocompletado, etc).

#### 8.1.3 HTML5

HTML5 (Siglas de HyperText Markup Language, versión 5) es la quinta versión del lenguaje de marcado de la World Wide Web regulado por el W3C cuya versión definitiva se publicó en octubre de 2014. Incluye numerosas etiquetas nuevas y funcionalidades con respecto a su versión anterior.

##### 8.1.3.1 Utilización en el proyecto

HTML5 será el lenguaje de marcado utilizado en las páginas de la aplicación.

#### 8.1.4 CSS3

CSS3 o Cascade Style Sheet 3 es un lenguaje creado para definir la presentación de documentos HTML. Sus especificaciones son gestionadas por el W3C.

##### 8.1.4.1 Utilización en el proyecto

Los estilos del sitio web serán definidos utilizando CSS3.

#### 8.1.5 Django

Django es un framework de Python de alto nivel enfocado en el desarrollo rápido de aplicaciones web opensource. Permite programación orientada a objetos e incluye gran parte de las herramientas necesarias y módulos reutilizables para no tener que reescribir código.

### 8.1.5.1 *Utilización en el proyecto*

La aplicación estará escrita utilizando este framework en el backend.

## 8.1.6 Twitter Bootstrap

Bootstrap es un framework HTML, CSS y JS (Javascript) que permite desarrollar de manera sencilla y rápida sitios web adaptables (*responsive*) compatible con dispositivos de escritorio y móviles.

### 8.1.6.1 *Utilización en el proyecto*

La presentación de la aplicación utiliza este framework puesto que proporciona maneras sencillas y cómodas de implementar todos los elementos necesarios por la aplicación.

## 8.1.7 pip

pip es el principal gestor de paquetes de Python.

### 8.1.7.1 *Utilización en el proyecto*

Para el funcionamiento de la aplicación será necesario instalar distintos paquetes y módulos utilizando pip.

## 8.1.8 Git

### 8.1.8.1 *Descripción*

Git es un sistema de control de versiones diseñado por Linus Torvalds. Está pensado para gestionar proyectos con una gran cantidad de código gestionado por mucha gente de manera distribuida.

Entre sus características se encuentran que los “commits” se realizan de manera local, lo que le proporciona una enorme rapidez, mientras que cuando se desea subir el código al repositorio se debe utilizar el comando “push”.

### 8.1.8.2 *Utilización en el proyecto*

Para llevar un control de las versiones del proyecto se ha decidido utilizar Git debido a su integración con el IDE, así como a su sencillez de uso. Habría sido prácticamente imposible desarrollar el proyecto sin un control de versiones ya que las pérdidas de información habrían sido constantes.

## 8.1.9 PostgreSQL

### 8.1.9.1 *Descripción*

PostgreSQL es un sistema de bases de datos relacionales Opensource y multiplataforma. Soporta los tipos de datos, cláusulas, funciones y dispone de herramientas gráficas de administración. Una ventaja importante es que su rendimiento es óptimo cuando la cantidad de datos almacenada en la base de datos es grande.

### 8.1.9.2 *Utilización en el proyecto*

Para la persistencia de datos del sistema se ha optado por utilizar PostgreSQL debido a su alto rendimiento, escalabilidad y funcionalidades, así como por su integración con Django.

## 8.2 Herramientas y programas usados para el desarrollo

### 8.2.1 PyCharm

#### 8.2.1.1 Descripción

PyCharm es un editor de código con soporte para Python, Javascript, HTML, CSS etc. Dispone de detección de errores y dispone de una gran integración con Django, pip, etc.

#### 8.2.1.2 Utilización en el proyecto

PyCharm será el IDE sobre el que se programará la aplicación.

### 8.2.2 Visual Paradigm

#### 8.2.2.1 Descripción

Visual Paradigm es un programa de modelado UML multiplataforma escrito en Java con soporte para diagramas de casos de uso, diagramas de requisitos SysML y diseño de bases de datos. Está desarrollado por la empresa Visual Paradigm International y dispone de una versión gratuita denominada Community Edition. También dispone de versiones de pago que van desde los 99 dólares hasta los 1399.

#### 8.2.2.2 Utilización en el proyecto

Visual Paradigm ha sido utilizado para desarrollar todos los diagramas del proyecto.

### 8.2.3 GitHub

#### 8.2.3.1 Descripción

GitHub es una plataforma de alojamiento de proyectos creada por la empresa GitHub Inc. en 2008. La principal característica de GitHub es que utiliza el control de versiones GIT<sup>14</sup> para almacenar los proyectos. Por defecto, el almacenamiento es público aunque existen versiones de pago para mantener repositorios privados.

El 21 de Septiembre de 2011 GitHub anunció que había llegado al millón de usuarios y había superado la barrera de los 2 millones de repositorios.

#### 8.2.3.2 Utilización en el proyecto

Para mantener un repositorio con el código del proyecto se decidió utilizar GIT como tecnología y GitHub como alojamiento debido a la compatibilidad con XCode y facilidad de uso de la misma.

### 8.2.4 Microsoft Project

#### 8.2.4.1 Descripción

Microsoft Project es una herramienta de planificación de proyectos creada por la empresa Microsoft desarrollada para Windows.

Permite establecer tareas, agruparlas, establecer su fecha y duración y asignarles recursos. También permite determinar el horario laboral entre muchas otras cosas.

---

<sup>14</sup> Sistema distribuido de control de versiones. Dispone de su propia descripción: Git

#### 8.2.4.2 *Utilización en el proyecto*

Microsoft Project ha sido la herramienta utilizada para realizar la planificación de toda la ejecución del proyecto.

### 8.2.5 Balsamiq Mockups

#### 8.2.5.1 *Descripción*

Balsamiq Mockups es una herramienta que permite realizar bocetos de interfaces de usuario de una forma rápida y sencilla.

#### 8.2.5.2 *Utilización en el proyecto*

Los bocetos de la interfaz de usuario se realizarán utilizando este programa.

### 8.2.6 PgCommander

#### 8.2.6.1 *Descripción*

PgCommander es una aplicación para Mac OS X que permite visualizar los datos almacenados en las tablas de una base de dato PostgreSQL.

#### 8.2.6.2 *Utilización en el proyecto*

Se ha utilizado en el proyecto para comprobar que la información almacenada en la base de datos tiene la estructura esperada.

### 8.2.7 PostgresApp

#### 8.2.7.1 *Descripción*

PostgresApp es una aplicación para Mac OS X que permite iniciar un servidor de base de datos PostgreSQL de una manera muy sencilla, simplemente ejecutando la aplicación. Dispone de una documentación suficiente en su página oficial.

#### 8.2.7.2 *Utilización en el proyecto*

Para la utilización de un entorno de desarrollo local en Mac OS X se ha utilizado PostgresApp debido a su sencillez de uso y a sus características.

### 8.2.8 MAMP

#### 8.2.8.1 *Descripción*

MAMP (My Apache, MySQL, PHP) es una aplicación disponible para Mac OS X y Windows que permite la ejecución de un un servidor Apache de manera sencilla y sin afectar al resto del sistema.

#### 8.2.8.2 *Utilización en el proyecto*

Se ha utilizado durante el desarrollo para efectuar la función de servidor de datos estáticos utilizando el protocolo HTTPS y un certificado autofirmado en local.

## 8.3 Creación del sistema

### 8.3.1 Librerías pip utilizadas

A continuación se enumeran y describen las librerías de pip utilizadas en el proyecto. Estas librerías se instalan automáticamente por medio del comando:

```
pip install -r pip-requirements.txt
```

Nombre	Descripción
<b>sslserver</b>	Módulo que permite hacer funcionar el servidor de desarrollo de Django utilizando SSL.
<b>south</b>	Módulo que proporciona una manera sencilla de gestionar los cambios de los atributos de los modelos en la base de datos, generando automáticamente las migraciones necesarias.
<b>djangosecure</b>	Añade un Middleware de seguridad para las peticiones realizadas al sitio web, previniendo ataques XSS o detectando proxies maliciosos entre otras cosas.
<b>registration</b>	Proporciona los recursos necesarios para permitir registro y gestión de usuarios en el frontend, incluyendo sistema de recuperación de contraseñas por medio de email.
<b>django_bootstrap_breadcrumbs</b>	Permite personalizar las migas de Django para hacerlas compatibles con Bootstrap.
<b>widget_tweaks</b>	Añaden diversas funcionalidades a las etiquetas de las plantillas para añadir clases CSS a diversos atributos (por ejemplo campos de formularios)
<b>uuidfield</b>	Añade el campo UUIDField para posibilitar su utilización en los modelos de Django.
<b>django_ajax</b>	Módulo que añade funcionalidades AJAX al sitio web.
<b>selectable</b>	Añade un campo de selección con funciones de autocompletado por medio de AJAX.
<b>awesome_avatar</b>	Añade un campo de avatar a los modelos.
<b>cookieclaw</b>	Módulo que añade un mensaje de privacidad de cookies al sitio.

<b>braces</b>	Añade distintos mixins (clases que cumplen unas funciones determinadas, como comprobar que un usuario está registrado) para beneficiarse de la herencia múltiple y no tener que repetir código.
<b>captcha</b>	Añade un campo de captcha de Google a los modelos de formulario.
<b>haystack</b>	Añade un sistema de búsqueda al sitio web.
<b>rest_framework</b>	Añade un sistema de API REST al sitio web.

### 8.3.2 Elementos ajenos al sitio web

Además de la aplicación web principal se han creado cuentas en otros servicios para proporcionar funcionalidades adicionales. A continuación se enumeran los servicios ajenos utilizados:

- **Blog:** Se ha utilizado Wordpress para crear el blog oficial de la aplicación. Se ha aprovechado también el blog para añadir una sección de preguntas frecuentes.
- **Google Analytics:** Se ha creado una cuenta en este servicio de Google para llevar la analítica web de la aplicación.
- **Twitter<sup>15</sup>:** Se ha creado la cuenta de Twitter @codesnug para estar en contacto con los usuarios de la aplicación en las redes sociales y publicar noticias.
- **Foro Uservice<sup>16</sup>:** Se ha creado también un foro para que los usuarios puedan sugerir nuevas funcionalidades para la aplicación y notificar errores.

### 8.3.3 Política de privacidad y condiciones legales

Se ha añadido al sitio web una página de política de privacidad y condiciones legales basándose en la legislación española.

#### 8.3.3.1 *Contenido de la política de privacidad*

##### 8.3.3.1.1 *Política de Privacidad y Datos*

Según lo establecido en la Ley Orgánica 15/1999 de protección de datos de carácter personal (LOPD), informamos que mediante la cumplimentación del formulario de registro en el sitio, sus datos pasarán a formar parte de un fichero con información de usuarios propiedad de "Codesnug.com". Dicho fichero no será compartido con terceros bajo ningún concepto.

La finalidad de dicho fichero es la gestión de usuarios registrados, envío de noticias y recordatorios de contraseñas a la dirección que ha introducido el usuario.

Si un usuario no desea recibir más información deberá pulsar el enlace correspondiente que se incluye en todos los emails que se le envían y desactivar la opción en su cuenta de usuario.

<sup>15</sup> Twitter de Codesnug: <https://twitter.com/codesnug>

<sup>16</sup> Página web del foro de Codesnug: <https://codesnug.uservice.com/forums/260358-general>

Codesnug.com garantiza la confidencialidad de todos los datos aportados y asegura que bajo ningún caso serán cedidos a terceros sin el consentimiento expreso del usuario. Solamente se solicitará al usuario los datos mínimos necesarios para la utilización del servicio y únicamente serán utilizados para el fin principal.

Se garantiza que para el almacenamiento de datos se han adoptado los niveles de seguridad adecuados a dichos datos, establecidos por la legislación vigente.

#### 8.3.3.1.1.1 Política de cookies

El acceso a este sitio puede implicar la utilización de cookies y se notifica correspondiente al usuario cuando accede al mismo. Las cookies son pequeñas cantidades de información que se almacenan en el navegador utilizado por cada usuario para que el servidor recuerde cierta información que posteriormente pueda utilizar. Esta información permite identificarle a usted como un usuario concreto y permite guardar sus preferencias personales, así como información técnica como puedan ser visitas o páginas concretas que visite.

Aquellos usuarios que no deseen recibir cookies o quieran ser informados antes de que se almacenen en su ordenador, pueden configurar su navegador a tal efecto.

La mayor parte de los navegadores de hoy en día permiten la gestión de las cookies de 3 formas diferentes:

- Las cookies no se aceptan nunca.
- El navegador pregunta al usuario si se debe aceptar cada cookie.
- Las cookies se aceptan siempre.

El navegador que utilice también puede incluir la posibilidad de especificar mejor qué cookies tienen que ser aceptadas y cuáles no. En concreto, el usuario puede normalmente aceptar alguna de las siguientes opciones: rechazar las cookies de determinados dominios; rechazar las cookies de terceros; aceptar cookies como no persistentes (se eliminan cuando el navegador se cierra); permitir al servidor crear cookies para un dominio diferente. Además, los navegadores pueden también permitir a los usuarios ver y borrar cookies individualmente.

##### 8.3.3.1.1.1.1 Cookies propias

- **cookielaw\_accepted**: Cookie que almacena si el usuario ha aceptado las cookies del sitio web.
- **csrftoken**: Cookie de los formularios del sitio.
- **sessionid**: Session cookie id.

\* Dispone de más información sobre las Cookies en: <http://es.wikipedia.org/wiki/Cookie>

#### 8.3.3.1.1.2 Web Beacons

Este sitio puede albergar también web beacons (también conocidos por web bugs). Los web beacons suelen ser pequeñas imágenes de un pixel por un pixel, visibles o invisibles colocados dentro del código fuente de las páginas web de un sitio. Los Web beacons sirven y se utilizan de una forma similar a las cookies. Además, los web beacons suelen utilizarse para medir el tráfico de usuarios que visitan una página web y poder sacar un patrón de los usuarios de un sitio.

\* Dispone de más información sobre los web beacons en: [http://es.wikipedia.org/wiki/Web\\_bug](http://es.wikipedia.org/wiki/Web_bug)

#### 8.3.3.1.1.3 Terceros

En algunos casos, compartimos información sobre los visitantes del sitio de forma anónima o agregada con terceros como puedan ser anunciantes, patrocinadores o auditores con el único fin de mejorar nuestros servicios. Todas estas tareas de procesamiento serán reguladas según las normas legales y se respetarán todos sus derechos en materia de protección de datos conforme a la regulación vigente.



Este sitio mide el tráfico con diferentes soluciones que pueden utilizar cookies o web beacons para analizar lo que sucede en nuestras páginas. Actualmente utilizamos las siguientes soluciones para la medición del tráfico de este sitio. Puede ver más información sobre la política de privacidad de cada una de las soluciones utilizadas para tal efecto:

- Google (Analytics): [http://www.google.com/intl/es\\_ALL/privacypolicy.html](http://www.google.com/intl/es_ALL/privacypolicy.html)
  - **\_ga**: Cookie de Google Analytics.

El sitio también puede albergar publicidad propia, de afiliados, o de redes publicitarias. Esta publicidad se muestra mediante servidores publicitarios que también utilizan cookies para mostrar contenidos publicitarios afines a los usuarios. Cada uno de estos servidores publicitarios dispone de su propia política de privacidad, que puede ser consultada en sus propias páginas web.

### 8.3.3.2 *Contenido de los Términos y Condiciones*

#### 8.3.3.2.1 *Términos y condiciones*

Como usuario de "Codesnug" debe proceder a la lectura del siguiente aviso legal. El acceso a los servicios proporcionados por el sistema supone la aceptación expresa de todo lo recogido en este texto. En caso de no aceptarlos, el sitio debe ser abandonado inmediatamente.

El acceso al sitio web de Codesnug.com implica la aceptación sin reservas de las presentes condiciones generales de uso que el usuario afirma comprender en su totalidad.

El usuario se compromete a no utilizar el sitio web y los servicios que se ofrecen en el mismo para la realización de actividades contrarias a la ley y a respetar en todo momento las presentes condiciones generales.

##### **8.3.3.2.1.1 Responsabilidad Legal**

Codesnug es un servicio que permite a sus usuarios almacenar, organizar y acceder a fragmentos de código. El autor del código mantiene todos los derechos de propiedad intelectual del mismo, aunque los usuarios que tengan acceso a él podrían copiarlo o compartirlo.

##### **8.3.3.2.1.2 Información de contacto**

Correo electrónico: [info@codesnug.com](mailto:info@codesnug.com)

##### **8.3.3.2.1.3 Licencia de contenidos y propiedad intelectual**

Salvo que se indique expresamente lo contrario, la propiedad intelectual sobre todo el contenido correspondiente a la construcción del sitio web (código fuente, javascript, hojas de estilos, etc.) del sitio pertenece a los autores del mismo, así como los distintivos, nombre y marcas comerciales del sitio web.

Se excluyen de este término los paquetes y librerías de terceros utilizados, que pueden disponer de sus propias licencias.

Toda la información publicada en el sitio por los usuarios pertenece exclusiva y únicamente a los propios usuarios y no está permitido su uso por terceras partes para cualquier fin ajeno a este sitio web bajo ningún pretexto y sin consentimiento del usuario.

##### **8.3.3.2.1.4 Responsabilidad de los enlaces de los usuarios**

Según lo establecido en los artículos 16 y 17 de la Ley 34/2002 (LSSICE), los autores del sitio web no se hacen responsables de la licitud ni la información almacenada por los usuarios, siendo estos los únicos responsables de su información, de los enlaces que hayan añadido y los terceros de quienes provengan estos enlaces.

En ningún caso la existencia de un enlace a terceros implica la existencia de una relación con Codesnug.com ni su conocimiento, aceptación y aprobación.

En el caso de que un órgano competente declare como ilícita cierta información contenida en este sitio, ordenando su retirada o solicitando información sobre el usuario autor de la misma, se le proporcionarán dichos datos según lo establecido en el artículo 36 de la ley 34/2002.

Codesnug.com y todo su contenido está sujeto a la legislación española vigente.

## 8.3.4 Problemas encontrados

### 8.3.4.1 Seguridad en la transmisión de los datos



#### **codesnug.com**

Emitido por: COMODO RSA Domain Validation Secure Server CA  
Caduca: sábado, 6 de junio de 2015, 1:59:59 (Hora de verano de Europa central)

✓ Este certificado es válido

Ilustración 86. Captura del mensaje del certificado del sitio web

Para asegurar que la transmisión de datos privados como contraseñas o tokens de creación de usuarios se transmiten de manera segura se utilizan conexiones seguras en todo el sitio web. Para ello se ha comprado un certificado firmado por la empresa COMODO S.A. y se ha instalado en el servidor.

### 8.3.4.2 Seguridad en Formularios

La verificación de los formularios se realiza tanto en el cliente como en el servidor. Para ello, Django implementa diversas medidas de seguridad.

Cuando se define un formulario, debe crearse un objeto que incluya los campos y sus tipos (CharField, IntegerField, EmailField, DateField, etc). Esto permite que cuando se crea la instancia del formulario en el frontend y se envía al servidor, es Django el encargado de realizar la validación de estos campos y devolver errores o confirmaciones en el caso de que sea necesario. Se pueden definir atributos obligatorios y opcionales de tal forma que se añada validación en el frontend mediante HTML5 cuando se renderiza el formulario.

Otras medida de seguridad que implementa Django es la verificación CSFR para evitar la suplantación de peticiones cross-site y el cifrado de las cookies de sesión.

### 8.3.4.3 Seguridad en URLS

Una de las cuestiones más importantes a lo largo del diseño del sistema fue la de la organización de las URL de los distintos datos del sistema. Por defecto Django utiliza los id de los objetos en las URL, pero como medida de seguridad se ha optado por utilizar un atributo diferente para ello. El atributo elegido es el UUID puesto que es un atributo único, para lo cual se ha instalado el módulo correspondiente que genera este tipo de campo en los modelos utilizando pip.

De esta forma, una URL quedará de la siguiente forma:

<https://codesnug.com/app/usuario/goals/071197e2b3b34e268a0e377bff57f9d8/>

El precio que se paga con esta medida es que la URL queda menos limpia, pero se oculta a los usuarios los id de los objetos para dificultar los posibles ataques.

### 8.3.4.4 Validación del HTML y accesibilidad

Se han realizado pruebas de accesibilidad al sitio web (WCAG 1.0) y también al HTML del mismo (HTML5) y se ha determinado que el sitio web tiene algunos errores fácilmente subsanables fácilmente pero hay otros que debido a los frameworks y tecnologías utilizadas no se pueden corregir de una manera trivial.

#### 8.3.4.4.1 Accesibilidad

Estado de los puntos de control				
Prioridad	Verificar	Bien	Mal	N/A
 <b>P1</b> WCAG 1.0	6 🔍	--	3 ✖	8 ✔
 <b>P2</b> WCAG 1.0	19 🔍	2 ✔	5 ✖	3 ✔
 <b>P3</b> WCAG 1.0	11 🔍	2 ✔	2 ✖	4 ✔

Ilustración 87. Resultados del test de accesibilidad Hera

Según los resultados obtenidos en el test Hera, existen errores de nivel 1 referentes a la activación de enlaces mediante Javascript, pero al ser una característica interna de Django a la que hacen referencia (cambio de idioma del sitio web), estos no se pueden corregir de una manera sencilla.

Los errores notificados en el nivel 2 se corresponden con algunos de los iconos mostrados (como por ejemplo, el icono del menú superior) y también a errores debidos a los estándares utilizados en el CSS, que al ser producidos por librerías externas también resultarían tener una difícil solución.

#### 8.3.4.4.2 Validación HTML

En cuanto a la validación HTML5, el validador oficial del W3C devuelve 8 errores referentes a los textos alternativos en algunas imágenes, que en este caso sí podrían tener una solución relativamente sencilla aunque habría que observar cada una de las páginas internas para disponer de un análisis verdaderamente exhaustivo.

### 8.3.4.5 Visualización de los elementos en pantallas pequeñas

Llegado a un punto del diseño de las interfaces de usuario de la aplicación se determinó que quizás es espacio en pantalla dedicado al código era demasiado pequeño, por lo que se establecieron algunos elementos podían variar su tamaño vertical y horizontal bajo demanda del usuario.

En el caso de la barra lateral izquierda principal de la aplicación, se puede pulsar el botón “Cerrar menú”, que mediante Javascript minimizará este menú y ampliará el espacio disponible en la zona principal en la que se muestra el código.

## 8.3.4.5.1 Ejemplo de vista expandida

The screenshot shows the CodeSnug interface for a goal titled "Database Backup Cronjob". The left sidebar is expanded, showing categories like "Goals", "Espacios de Trabajo", and "Grupos de Usuarios". The main content area displays the goal's details, including a description "Export your database using a Cronjob" and a code snippet for "cronjob.sh". The right sidebar shows "Información del Goal" with details like "Privacidad: Público", "Número de snippets: 1", and "Autor: loladmin2014".

**Database Backup Cronjob**

Inicio / Goals / Goal: Database Backup Cronjob

Editar **Borrar** [Volver](#)

Export your database using a Cronjob

**cronjob.sh** SH

```

1 DBNAME="<DB_NAME>"
2 DBPASS="<PASSWORD>"
3 DBUSER="<USER>"
4 EMAIL="<EMAIL>"
5
6 mysqldump --opt -u $DBUSER -p$DBPASS $DBNAME > backup.sql
7 gzip backup.sql
8 DATE=`date +%Y%m%d` ; mv backup.sql.gz $DBNAME-backupFULL-$DATE.sql.gz
9 uuencode $DBNAME-backupFULL-$DATE.sql.gz $DBNAME-backupFULL-$DATE.sql.gz |
10

```

Descargar Snippet [Ver versiones](#)

**Información del Goal**

Privacidad: **Público**

Número de snippets: 1

Autor: loladmin2014

Fecha de creación: 19 de Abril de 2015 a las 11:35

**Espacios de Trabajo**

- Work

Ilustración 88. Vista de la barra lateral izquierda expandida

## 8.3.4.5.2 Ejemplo de vista con el menú cerrado

The screenshot shows the CodeSnug interface for the same goal, but with the left sidebar collapsed. The main content area and right sidebar are visible, showing the goal's details and the "cronjob.sh" code snippet.

**Database Backup Cronjob**

Inicio / Goals / Goal: Database Backup Cronjob

Editar **Borrar** [Volver](#)

Export your database using a Cronjob

**cronjob.sh** SH

```

1 DBNAME="<DB_NAME>"
2 DBPASS="<PASSWORD>"
3 DBUSER="<USER>"
4 EMAIL="<EMAIL>"
5
6 mysqldump --opt -u $DBUSER -p$DBPASS $DBNAME > backup.sql
7 gzip backup.sql
8 DATE=`date +%Y%m%d` ; mv backup.sql.gz $DBNAME-backupFULL-$DATE.sql.gz
9 uuencode $DBNAME-backupFULL-$DATE.sql.gz $DBNAME-backupFULL-$DATE.sql.gz | mail -s "<SUBJECT>" $!
10

```

Descargar Snippet [Ver versiones](#)

**Información del Goal**

Privacidad: **Público**

Número de snippets: 1

Autor: loladmin2014

Fecha de creación: 19 de Abril de 2015 a las 11:35

**Espacios de Trabajo**

- Work

Ilustración 89. Vista con la barra lateral izquierda cerrada

En cuanto a la variación vertical, se encuentra en las página de creación/edición de un goal. La barra lateral derecha dispone de elementos colapsables para permitir que sus diferentes apartados se encuentren siempre disponibles sin imponer la resolución y espacio vertical del medio en el que se están visualizando:

### 8.3.4.5.3 Ejemplo de vista expandida

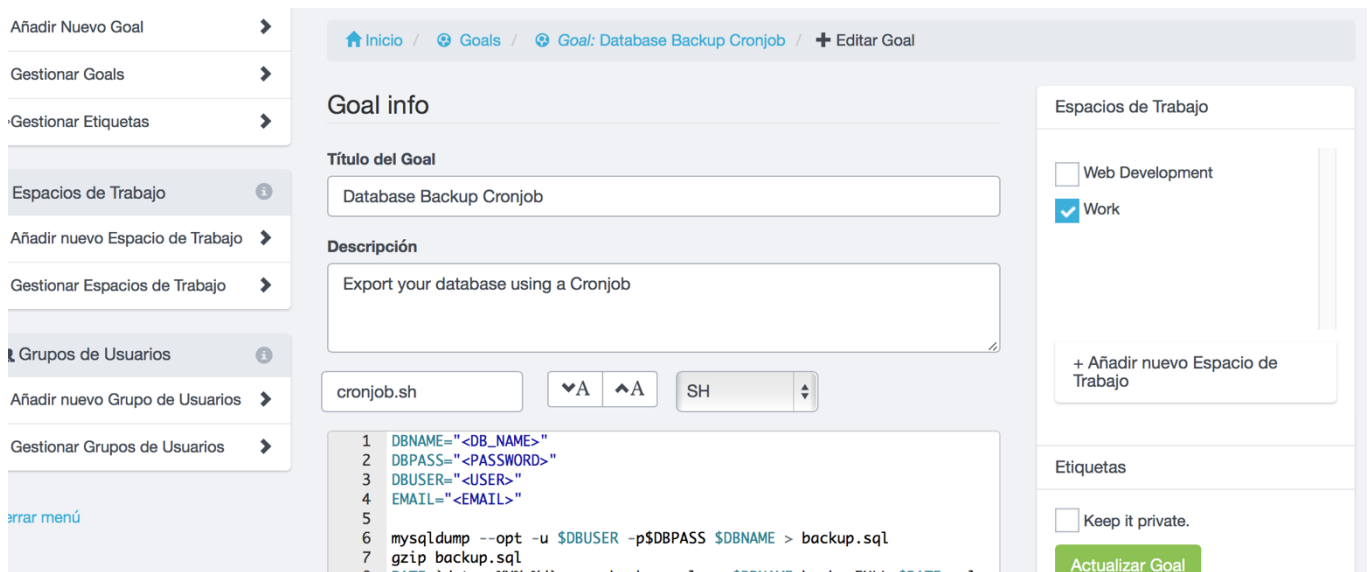


Ilustración 90. Vista de la barra lateral derecha expandida

### 8.3.4.5.4 Ejemplo de vista minimizada

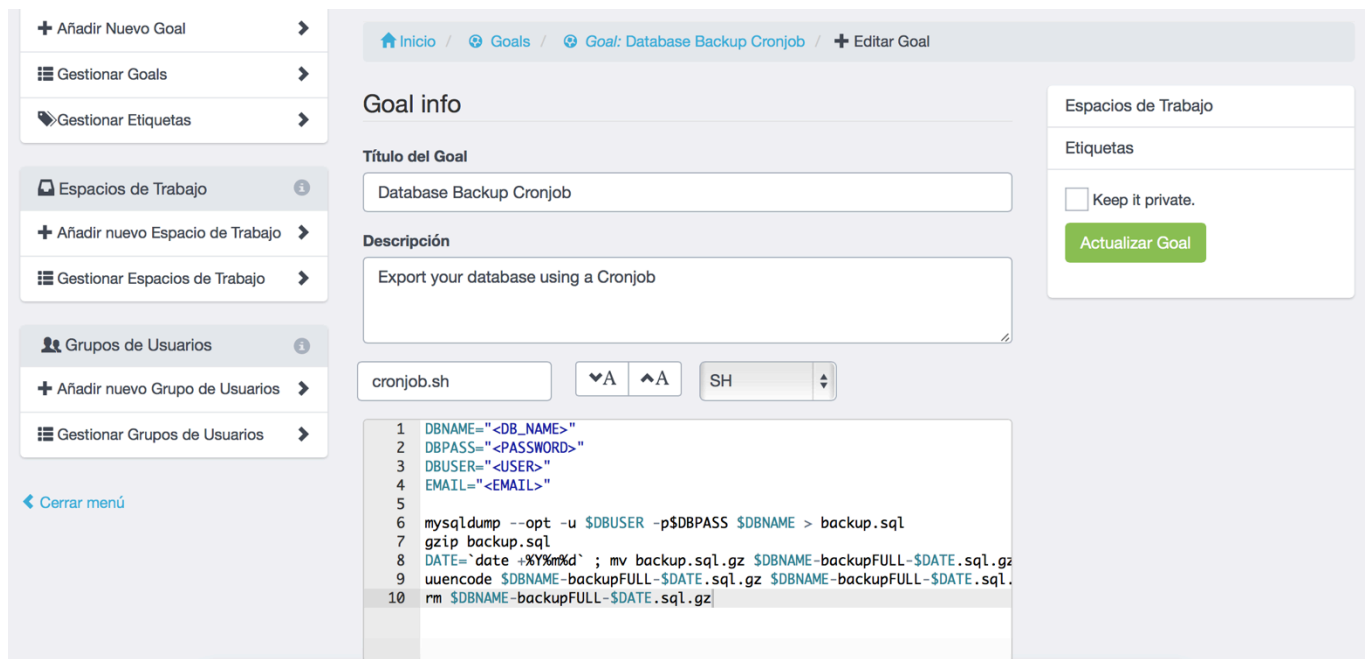
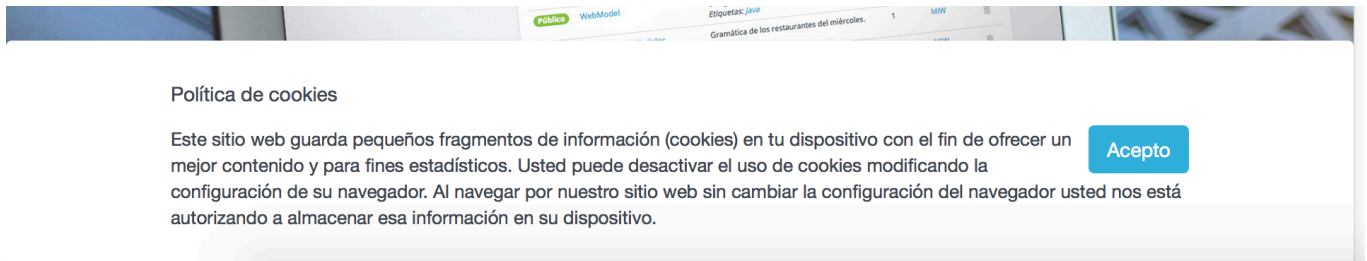


Ilustración 91. Vista de la barra lateral cerrada

### 8.3.4.6 *Política de Cookies*



**Ilustración 92. Vista del mensaje de cookies del sitio web**

Debido a que el sitio web se encuentra alojado en un servidor europeo ha sido necesario implementar un mensaje de aviso de cookies. Para ello se ha utilizado el módulo de pip cookieLaw.

### 8.3.4.7 *Solicitar la edad de los usuarios en el registro*

Debido a la legislación española, solamente los mayores de 14 años pueden registrarse en una red social sin consentimiento paterno. Debido a la posible evolución de la aplicación a lo largo del tiempo y sus posibles variaciones en cuanto a funcionalidades sociales, adelantando una posible necesidad se solicita a los usuarios durante su registro su fecha de nacimiento.

## 9 Manuales del sistema

### 9.1 Manual de usuario

La aplicación dispone de una página web con FAQs en su blog oficial (<http://blog.codesnug.com/faqs/>). En esta página se irán añadiendo nuevas preguntas y respuestas sobre las tareas que puedan provocar más dudas a los usuarios del sitio web.

### 9.2 Manual del programador

En este apartado se encuentran sendos tutoriales sobre la instalación, gestión y organización de distintos aspectos del proyecto y de los distintos elementos que lo conforman.

#### 9.2.1 Entorno de desarrollo local

Para la ejecución de la aplicación en un entorno local es necesaria la ejecución de dos servidores. Por un lado el que se encargará de ejecutar la aplicación y por otro el encargado de servir los archivos estáticos. Para la ejecución del proyecto se utilizará el propio servidor de desarrollo de la aplicación con una conexión https utilizando el módulo "sslserver", mientras que para el servidor de archivos estáticos se utilizará MAMP.

##### 9.2.1.1 Creación de un entorno de desarrollo virtual

Antes de instalar las dependencias de la aplicación es conveniente la creación de un entorno virtual (virtualenv) para evitar incompatibilidades con otras aplicaciones por medio de la creación de un entorno independiente.

Para ello se instalará el módulo virtualenv utilizando pip:

```
pip install virtualenv
```

Y a continuación se instalará virtualenvwrapper, un envoltorio que simplificará su uso:

```
pip install virtualenvwrapper
```

Tras esto deberemos configurar virtualenvwrapper para decirle dónde se almacenarán nuestros entornos virtuales y nuestros proyectos añadiendo las siguientes líneas al archivo .bashrc o .zshrc de nuestro directorio (utilizando Mac o Linux).

```
export WORKON_HOME=$HOME/.virtualenvs
export PROJECT_HOME=$HOME/<Directorio con los proyectos>
source /usr/local/bin/virtualenvwrapper.sh
```

Una vez hecho esto, podremos crear nuestro entorno virtual:

```
mkvirtualenv <Nombre del entorno>
```

Y comenzar a trabajar en él:

```
workon <Nombre del entorno>
```

Para dejar de trabajar en él utilizaremos el comando:

```
deactivate
```

### 9.2.1.2 Puesta en funcionamiento del servidor local

Para poner en funcionamiento la aplicación se utilizará el comando (dentro del directorio de la misma):

```
./manage.py runsslserver
```

## 9.2.2 Creación y gestión de la base de datos y de sus migraciones

Para la gestión de la base de datos del proyecto se utilizan las herramientas integradas de Django, que incluyen su propio ORM para el acceso a la información de la base de datos y también se encargan de su creación y mantenimiento.

Para la migración y actualización de los modelos se utiliza el módulo South, que se encarga de manera automática de actualizar las tablas de la base de datos en el caso de que se realicen modificaciones en los modelos.

Para la creación y actualización por primera vez de la base de datos en el caso de que esta no requiera de migraciones debe utilizarse el comando:

```
./manage.py syncdb
```

Posteriormente debe utilizarse el siguiente comando para migrar las distintas apps que conforman la aplicación según se hayan modificado:

```
./manage.py migrate <goals, users>
```

## 9.2.3 Instalación de las dependencias del proyecto

Para instalar automáticamente los módulos necesarios para el funcionamiento de la aplicación utilizando PIP, debe ejecutarse el siguiente comando dentro del directorio de la aplicación:

```
pip install -r pip-requirements.txt
```

## 9.2.4 Generación de traducciones

Por defecto Django es capaz de analizar cadenas de traducción, generar los archivos de traducción (archivos po) y posteriormente compilarlos para ser utilizados dentro de la misma (archivos mo) Para generar los archivos de traducción es necesario utilizar los siguientes comandos cada vez que se realicen cambios en las cadenas de la aplicación.

En primer lugar es necesario generar el archivo de traducción con todas las cadenas de texto:

```
django-admin.py makemessages -a
```

Una vez introducido este comando, se generará un archivo po por cada uno de los idiomas soportados por la aplicación. Estos archivos generados deberán ser procesados un programa de traducción de cadenas como Poedit.

Una vez realizados todos los cambios, se introducirá el siguiente comando para compilar las traducciones:

```
django-admin.py compilemessages
```



## 9.2.5 Creación e instalación de un certificado de servidor para generar un SSL

Para utilizar SSL en el servidor ha sido necesario crear un certificado CSR con los datos del dominio que se utilizará para el proyecto (codesnug.com).

Para ello ha sido necesario conectarse mediante SSH al servidor e introducir los siguientes comandos:

En primer lugar creamos una clave privada RSA de 2048 bits:

```
openssl genrsa -out codesnug_com.key 2048
```

A continuación creamos el certificado a partir de la clave que acabamos de crear. Debemos introducir la información que se nos pida por pantalla teniendo en cuenta que es muy importante el campo “Common Name”, que deberá ser el dominio al cual aplicaremos el certificado SSL (codesnug.com):

```
openssl req -new -key codesnug_com.key -out codesnug_com.csr
```

Una vez hecho esto, mostraremos el certificado, lo copiaremos y lo pegaremos donde nos indique la empresa con la que contratemos el certificado SSL:

```
more codesnug.csr
```

## 9.2.6 Redireccionamiento de direcciones sin SSL y sin www

Como todo el sitio será accesible mediante SSL, se deberán redireccionar todas aquellas direcciones que pretendan acceder de manera no cifrada al sitio.

También se redireccionará todos aquellos accesos al sitio utilizando el prefijo www (www.codesnug.com) al sitio sin el prefijo (codesnug.com).

Para realizar estas redirecciones, se ha creado un archivo .htaccess en la raíz del sitio con el siguiente contenido:

```
Options +FollowSymLinks
RewriteEngine on
RewriteCond %{HTTP_HOST} ^www.codesnug.com$ [NC,OR]
RewriteCond %{HTTP_HOST} ^codesnug.com$ [NC]
RewriteRule ^(.*)$ https://codesnug.com/$1 [R=301,L]
```

## 9.2.7 Organización de un proyecto Git

### 9.2.7.1 Organización de las ramas

- **Develop:** En ella se irán haciendo todos los desarrollos en cuanto a funcionalidades.
- **Master:** La rama principal. Cuando haya suficientes funcionalidades, se irán añadiendo a esta rama.

### 9.2.7.2 Crear ramas

Crea la rama en local:

```
git branch develop
```

Envía la rama local al repositorio:

```
git push origin develop
```

### 9.2.7.3 Tags

Versión inicial: 0.1 (Asignar la primera vez que se hace commit del proyecto)

Por cada versión con funcionalidades en el Master, se crea una versión nueva (tag). Las tags son asignadas a la rama por defecto.

Para ver las ramas y cuál es la que está por defecto:

```
git branches
```

Para cambiar entre ramas:

```
git co [branch you want to switch to]
```

Para crear el tag:

```
git tag '0.1'
```

Para asignarle un tag a la rama:

```
git push --tags
```

## 9.2.8 Configuración de la búsqueda dentro de la aplicación

Para configurar la búsqueda dentro de la aplicación debe crearse un índice de búsqueda.

Para ello, debe ejecutarse siguiente comando:

```
python2.7 ./manage.py rebuild_index
```

Si deseamos actualizar el índice actual sin borrar el existente, el comando a utilizar debe ser el siguiente:

```
python2.7 ./manage.py update_index
```

Esto solo debería ser necesario la primera vez, puesto que la aplicación está configurada de tal forma que cuando se añade un nuevo modelo a la base de dato, los cambios se apliquen automáticamente.

## 9.2.9 Despliegue en producción

El despliegue en producción de la aplicación se realiza utilizando git. Para ello, es necesario disponer de un servidor con Git instalado y configurar el usuario en el servidor. Al estar alojado el proyecto en Github, es necesario generar una clave privada en el servidor y posteriormente agregarla a la cuenta de Github dueña del repositorio, lo que servirá para identificar al usuario en el servidor cuando se realicen operaciones.

En este caso, la única operación que se realizará será:

```
git pull
```

Para descargar y actualizar todos los datos del servidor desde el repositorio.

Existen archivos que no se sobrescribirán y que deberán ser añadidos manualmente, como la carpeta de los avatares de los usuarios y también el archivo de configuración de la aplicación, que incluye contraseñas, clave secreta, etc.

Utilizar Git para realizar despliegues garantiza que la información sensible no será transmitida en ninguna ocasión.

Cada vez que se incluyan archivos compilables de python en el proyecto deberá reiniciarse el servidor web para que los cambios tengan efecto.

## 10 Presupuesto detallado

Se han establecido varios roles diferentes con sus propios precios por hora para presupuestar las distintas fases del proyecto:

- **Alumno:** 25€/hora
- Programador: 35€/hora
- **Sysadmin:** 35€/hora
- **Analista:** 55€/hora
- **Community Manager:** 25€/hora
- Director del proyecto: 70€/hora

Item	Recurso	Coste unitario	Cantidad	Total
1	Ordenador	1.500 €	1	1.500 €
2	Hosting (1 año)	93€	1	93 €
3	Certificado SSL	10€	1	10€
4	Aprendizaje de las tecnologías	5€	120	600€
5	Análisis y diseño	55 €	132	7.260 €
6	Desarrollo	35 €	378	13.230 €
7	Administración de sistemas	35€	18	630 €
8	Dirección	70 €	90	6.300 €
9	Mantenimiento	35 €	144	5.040€
10	Gestión de la comunidad y analítica web	25 €	15	375 €
11	Conexión a internet	40€	12	480 €
			Subtotal	35.198 €
			IVA 21%	7.391 €
			<b>TOTAL</b>	<b>42.560 €</b>

# 11 Conclusiones y ampliaciones

## 11.1 Conclusiones

Analizando los requisitos iniciales y el resultado final, se puede concluir que se ha conseguido alcanzar el objetivo marcado desde un inicio al crear la aplicación web.

Se ha creado un proyecto útil y con capacidades de explotación futura si se sigue trabajando en él y añadiéndole funcionalidades como las indicadas en el apartado siguiente. Resulta evidente que se trata de un nicho del mercado con cierta competencia de empresas potentes como Bitbucket o Github, por lo que habría que saber diferenciarse y desarrollar las futuras ampliaciones centrándose en aquellas características que verdaderamente marquen la diferencias y ofreciendo un alto grado de fiabilidad al usuario.

El proyecto creado utiliza tecnologías que le permitirán ser fácilmente mantenible en el futuro y también escalable. Gracias a su modularidad podrán sustituirse elementos del mismo sin afectar al resto del sistema.

Por otro lado, el desarrollo del presente proyecto ha servido al alumno a aprender nuevos lenguajes de programación, frameworks, entornos de desarrollo y metodologías de despliegues a producción que serán útiles en el presente y futuro.

## 11.2 Ampliaciones

### 11.2.1 Creación de una aplicación de escritorio

Una posible ampliación para la aplicación es la creación de una aplicación de escritorio para gestionar los goals utilizando su API.

### 11.2.2 Crear una aplicación móvil

De la misma forma que se podría crear una aplicación de escritorio, podría crearse una aplicación para el móvil

### 11.2.3 Extender las funcionalidades de la API

Se podrían también añadir más métodos a la API para gestionar los goals. Por ejemplo, añadiendo métodos de edición y borrado para los datos.

### 11.2.4 Autenticación Social

En vez de ser necesario crear una cuenta en la aplicación, podría utilizarse algún servicio externo como Google o Facebook. Esto permitiría evitar el efecto “pereza” a la hora de crear una cuenta en el servicio y agilizar el proceso.

### 11.2.5 Modificar los colores de la página para hacerlos más accesibles

Podría realizarse un estudio exhaustivo de la accesibilidad de la página web y establecerse una paleta de colores alternativa para usuarios con discapacidades visuales.

### 11.2.6 Permitir goals públicos

En un futuro, para favorecer la indexación en buscadores podrían establecerse unos nuevos niveles de privacidad que permitiesen establecer los goals como completamente públicos.

### 11.2.7 Permitir pantalla completa al visualizar los snippets

Una ampliación interesante podría ser la de añadir un botón que permitiese extraer la ventana del editor de un Snippet o visualizarlo en pantalla completa. Esto resultaría útil para snippets de un tamaño considerable.

### 11.2.8 Mejorar la accesibilidad y el código HTML utilizado

Para que los usuarios con discapacidades puedan utilizar la aplicación de manera lo más cómoda posible, se debería mejorar la accesibilidad de la aplicación para validar las pautas de accesibilidad marcadas por el WCAG y la validación HTML5 del W3C.

## 12 Referencias bibliográficas

### 12.1 Páginas Web

Título	Descripción	URL
<b>Which database should I use for a killer web application: MongoDB, PostgreSQL, or MySQL?</b>	Pregunta en Quora sobre qué sistema de base de datos es más recomendable utilizar.	<a href="http://www.quora.com/Which-database-should-I-use-for-a-killer-web-application-MongoDB-PostgreSQL-or-MySQL">http://www.quora.com/Which-database-should-I-use-for-a-killer-web-application-MongoDB-PostgreSQL-or-MySQL</a>
<b>Comparison of JavaScript-based source code editors</b>	Comparación de editores de código en Javascript.	<a href="http://en.wikipedia.org/wiki/Comparison_of_JavaScript-based_source_code_editors">http://en.wikipedia.org/wiki/Comparison_of_JavaScript-based_source_code_editors</a>
<b>Página oficial de Django Haystack</b>	-	<a href="http://haystacksearch.org/">http://haystacksearch.org/</a>
<b>Página oficial de Woosh</b>	-	<a href="https://bitbucket.org/mchaput/whoosh/wiki/Home">https://bitbucket.org/mchaput/whoosh/wiki/Home</a>
<b>Python – Wikipedia</b>	Definición de Python en la Wikipedia	<a href="http://es.wikipedia.org/wiki/Python">http://es.wikipedia.org/wiki/Python</a>
<b>Javascript – Wikipedia</b>	Definición de Javascript en la Wikipedia	<a href="http://es.wikipedia.org/wiki/JavaScript">http://es.wikipedia.org/wiki/JavaScript</a>
<b>CSS3 – Wikipedia</b>	Definición de CSS3 en la Wikipedia	
<b>HTML5 – Wikipedia</b>	Definición de HTML5 en la Wikipedia	<a href="http://es.wikipedia.org/wiki/HTML5">http://es.wikipedia.org/wiki/HTML5</a>
<b>Github Gist</b>	Página oficial de Github Gists	<a href="https://gist.github.com/">https://gist.github.com/</a>
<b>Gistbox</b>	Página oficial de Gistbox	<a href="http://www.gistboxapp.com">http://www.gistboxapp.com</a>
<b>Snipplr</b>	Página oficial de Snipplr	<a href="http://snipplr.com">http://snipplr.com</a>
<b>PgCommander</b>	Página oficial de PgCommander	<a href="https://eggerapps.at/pgcommander/">https://eggerapps.at/pgcommander/</a>

Título	Descripción	URL
<b>PostgresApp</b>	Página oficial de PostgresApp	<a href="http://postgresapp.com">http://postgresapp.com</a>
<b>MAMP</b>	Página oficial de MAMP	<a href="https://www.mamp.info/en/">https://www.mamp.info/en/</a>
<b>Django Rest framework</b>	Página oficial de Django REST framework	<a href="http://www.django-rest-framework.org">http://www.django-rest-framework.org</a>
<b>Which database should I use for a killer web application: MongoDB, PostgreSQL, or MySQL?</b>	Debate sobre qué tecnologías de bases de datos	<a href="http://www.quora.com/Which-database-should-I-use-for-a-killer-web-application-MongoDB-PostgreSQL-or-MySQL">http://www.quora.com/Which-database-should-I-use-for-a-killer-web-application-MongoDB-PostgreSQL-or-MySQL</a>
<b>Successful git branching</b>	Página con información sobre cómo organizar en git un proyecto	<a href="http://nvie.com/posts/a-successful-git-branching-model/">http://nvie.com/posts/a-successful-git-branching-model/</a>
<b>Django 1.7 Docs</b>	Página con la documentación de Django	<a href="https://docs.djangoproject.com/en/1.7/">https://docs.djangoproject.com/en/1.7/</a>
<b>Whoosh Docs</b>	Cómo definir un esquema de búsqueda	<a href="http://whoosh.readthedocs.org/en/latest/schema.html">http://whoosh.readthedocs.org/en/latest/schema.html</a>



