

GRADO EN INGENIERÍA INFORMÁTICA DE  
GESTIÓN Y SISTEMAS DE INFORMACIÓN  
**TRABAJO FIN DE GRADO**

***GESTIÓN DE FANFICS***

**Alumno/Alumna:** de Jaime, de Toro, Paula

**Director/Directora:** Blanco, Jauregi, Begoña

**Curso:** 2018-2019

**Fecha:** Bilbao, 17, 06, 2019



## 0. Resumen

Mediante este proyecto se ha conseguido desarrollar una aplicación web que permite gestionar *fanfics* o historias ficticias. Los usuarios podrán llevar la cuenta de los capítulos que han leído y que les quedan por leer de forma sencilla. Se ha utilizado el *framework* de *Python* llamado *Django*, entre otras herramientas.

Se podrán añadir nuevos *fanfics*, simplemente importando estos desde las páginas web compatibles debidamente indicadas, y crear listas para organizar aquellas historias en las que se tiene interés.

La aplicación web también cuenta con un sistema de notificaciones en tiempo real, para que los usuarios puedan estar al tanto de las últimas actualizaciones de los *fanfics* y de otros miembros de la aplicación.

Por último, se ha implementado un sistema de recomendaciones para poder ofrecer *fanfics* similares a uno visitado, además de ayudar a visualizar los datos obtenidos de los usuarios para ayudar a tomar decisiones de negocio.

# 1. Laburpena

Proiektu honen bidez web aplikazio bat lortu da, zehazki, fanfics edo fikziozko istorioak kudeatzen laguntzen duena. Aplikazio honen erabiltzaileek irakurritako (eta irakurri gabeko) kapituluak modu erraz batean kontrolatu ditzakete.

Hau lortzeko, besteak beste, *Django* izeneko *framework*-a erabili da. Honek, *Python*-en bidez eginda dago.

Bestalde, erabiltzaileek *fanfic* berriak gehitu ditzakete adierazitako bate-ragarri diren web orrietatik hauek inportatuz, besterik ez. Gainera, arreta deitzen dieten istorioek zerrendetan antolatzeko aukera izango dute ere.

Web aplikazio honek bi sistema garrantzitsu ditu barne: abisuak denbo-  
ra errealean jasotzeko sistema eta gomendio sistema bat. Lehenengoa era-  
biltzaileek azken aktualizazioen eta beste erabiltzaileen mugimenduen berri  
izan ditzaten burutu egin da. Eta, bigarrenaren implementazioaren bitartez,  
bakoitzak irakurritako kapituluaren antzekoak diren beste *fanfic*-ak errazago  
aurki ditzakete.

Amaitzeko, azken sistema honek lortutako datuak erakuzten ditu negozio  
erabakiak hartzen laguntzeko.

## 2. Abstract

The aim of this project was to develop a web application for the users to manage the *fanfics* or fiction stories they liked. Users can see the chapters they have read and the ones they have not yet. And for the development of this application, the *Django framework* has been used.

Users are able to import *fanfics* too. The site hosting the story they want to import has to be one of the specified ones. In other words, not all the stories can be added. Lists can also be created to organize the *fanfics* a user is interested in.

Users are notified in real time of their *fanfics* updates and other users actions.

They can get similar stories thanks to the recommendation system implemented inside the application. Lastly, graphs will be generated to visualize user data.

# Índice

<b>0. Resumen</b>	<b>2</b>
<b>1. Laburpena</b>	<b>3</b>
<b>2. Abstract</b>	<b>4</b>
<b>3. Definiciones y Vocabulario Técnico</b>	<b>18</b>
<b>4. Introducción</b>	<b>20</b>
4.1. Descripción y situación del trabajo . . . . .	20
4.2. Planteamiento del problema . . . . .	22
4.3. Justificación del TFG . . . . .	24
4.3.1. Aprendizaje . . . . .	24
4.3.2. Curiosidad . . . . .	25
4.3.3. Interés personal . . . . .	25
4.3.4. Diversión . . . . .	26
<b>5. Planteamiento Inicial</b>	<b>27</b>
5.1. Objetivos . . . . .	27
5.1.1. Objetivos técnicos . . . . .	27
5.2. Arquitectura . . . . .	29
5.2.1. Django . . . . .	29
5.2.2. WebSockets . . . . .	30
5.2.3. Tareas asíncronas . . . . .	31
5.2.4. Resumen . . . . .	31
5.3. Herramientas . . . . .	32
5.4. Alcance . . . . .	37
5.4.1. Organización . . . . .	38
5.4.2. Aprendizaje . . . . .	41
5.4.3. Captura de requisitos . . . . .	44
5.4.4. Análisis y Diseño . . . . .	47
5.4.5. Implementación y Desarrollo . . . . .	48

5.4.6.	Pruebas . . . . .	55
5.4.7.	Documentación . . . . .	62
5.4.8.	Resumen de la planificación realizada . . . . .	64
5.5.	Planificación Temporal . . . . .	67
5.6.	Evaluación económica . . . . .	70
5.6.1.	Recursos humanos . . . . .	70
5.6.2.	Software . . . . .	71
5.6.3.	Hardware . . . . .	72
5.6.4.	Libros . . . . .	73
5.6.5.	Otros . . . . .	73
5.6.6.	Gastos totales . . . . .	73
5.6.7.	Rentabilidad . . . . .	74
5.7.	Gestión de Riesgos . . . . .	75
5.7.1.	Enfermedad o lesión . . . . .	76
5.7.2.	Error en la planificación temporal . . . . .	78
5.7.3.	Falta de conocimientos . . . . .	79
5.7.4.	Perdida de información . . . . .	80
5.7.5.	Trabajo menos complejo de lo esperado . . . . .	81
5.7.6.	Problemas con el equipo informático . . . . .	82
5.7.7.	Cambio en la estructura de las páginas web . . . . .	83
<b>6.</b>	<b>Análisis de antecedentes</b>	<b>85</b>
<b>7.</b>	<b>Captura de requisitos</b>	<b>88</b>
7.1.	Jerarquía de actores . . . . .	88
7.2.	Casos de Uso . . . . .	89
7.2.1.	Cualquiera . . . . .	91
7.2.2.	Usuario identificado . . . . .	91
7.2.3.	Bibliotecario . . . . .	93
7.2.4.	Administrador . . . . .	94
7.3.	Modelo de Dominio . . . . .	95
<b>8.</b>	<b>Análisis y Diseño</b>	<b>99</b>

8.1. Diagrama de la Base de Datos . . . . .	99
8.2. Diagrama de clases . . . . .	101
8.3. Diagramas de secuencia . . . . .	105
<b>9. Desarrollo</b>	<b>106</b>
9.1. Desarrollo de prototipos . . . . .	106
9.2. Framework Django . . . . .	112
9.2.1. Peticiones . . . . .	113
9.2.2. Tareas asíncronas . . . . .	115
9.2.3. Notificaciones . . . . .	119
9.2.4. Panel de administración . . . . .	122
9.2.5. Tests . . . . .	126
9.2.6. Otros . . . . .	127
9.3. Ciencia de datos . . . . .	130
9.3.1. Sistema de recomendaciones . . . . .	130
9.3.2. Visualización de datos de los usuarios . . . . .	134
9.3.3. Compatibilidad de navegadores . . . . .	135
<b>10.Verificación y evaluación</b>	<b>136</b>
<b>11.Conclusiones</b>	<b>146</b>
11.1. Análisis entre planificación estimada y real . . . . .	146
11.1.1. Objetivos . . . . .	146
11.1.2. Herramientas . . . . .	146
11.1.3. Planificación temporal . . . . .	146
11.1.4. Evaluación económica . . . . .	152
11.2. Líneas futuras . . . . .	152
11.2.1. Identificación externa . . . . .	152
11.2.2. Aplicación multidioma . . . . .	152
11.2.3. Chat entre usuarios . . . . .	152
11.2.4. Extensión de Google Chrome . . . . .	153
11.2.5. Listas privadas de <i>fanfics</i> . . . . .	153
11.2.6. Tests de usabilidad . . . . .	153



11.3. Conclusión personal . . . . .	153
<b>Referencias</b>	<b>155</b>
<b>12.Anexo I: Casos de uso extendidos</b>	<b>156</b>
12.1. Cualquiera . . . . .	156
12.2. Bibliotecario . . . . .	161
12.3. Administrador . . . . .	175
12.4. Usuario identificado . . . . .	176
<b>13.Anexo II: Diagramas de secuencia</b>	<b>210</b>
<b>14.Anexo III: Prototipos baja fidelidad</b>	<b>217</b>
<b>15.Anexo IV: Prototipos alta fidelidad</b>	<b>225</b>
<b>16.Anexo V: Manual de instalación</b>	<b>238</b>
16.1. Configuración de la base de datos . . . . .	238
16.2. Instalación de librerías necesarias . . . . .	238
16.3. Credenciales . . . . .	241

## Índice de figuras

1.	Página principal de <i>Fanfiction.net</i> . . . . .	23
2.	Página principal de <i>Wattpad.com</i> . . . . .	23
3.	Arquitectura de la aplicación web . . . . .	31
4.	Grupos principales de las tareas . . . . .	37
5.	Diagrama Gantt de las tareas . . . . .	69
6.	Página principal de <i>FicTrack.com</i> . . . . .	86
7.	Página principal de <i>TvTime.com</i> . . . . .	87
8.	Jerarquía de actores . . . . .	89
9.	Diagrama casos de uso . . . . .	90
10.	Modelo de dominio . . . . .	96
11.	Diagrama relacional de la Base de Datos . . . . .	100
12.	Diagrama de clases . . . . .	102
13.	Diagrama de clases “añadir un <i>fanfic</i> ” . . . . .	103
14.	Diagrama de clases “sistema de recomendación” . . . . .	104
15.	Clase de las tareas . . . . .	104
16.	Prototipo baja fidelidad de la página de aterrizaje . . . . .	107
17.	Prototipo alta fidelidad de la identificación . . . . .	108
18.	Gama de colores . . . . .	109
19.	Fuentes utilizadas . . . . .	109
20.	Diseño <i>responsive</i> (I) . . . . .	110
21.	Diseño <i>responsive</i> (II) . . . . .	110
22.	Página principal de <i>Bulma.io</i> . . . . .	111
23.	Página principal de <i>Undraw.co</i> . . . . .	111
24.	Estructura del proyecto . . . . .	113
25.	Formulario para petición <i>DELETE</i> . . . . .	114
26.	Clase que implementa <i>DELETE</i> . . . . .	114
27.	Clase creada para la redirección de <i>DELETE</i> y <i>PUT</i> . . . . .	115
28.	Bloque excepción de la tarea asíncrona . . . . .	116
29.	Inicializar tarea asíncrona . . . . .	116
30.	El <i>worker</i> recibe una tarea . . . . .	118

31.	Notificación de <i>fanfic</i> añadido . . . . .	118
32.	Petición conexión <i>WebSocket</i> . . . . .	120
33.	Estructura de un <i>consumer</i> . . . . .	121
34.	Señal de las notificaciones . . . . .	122
35.	Una página del panel de administración . . . . .	123
36.	Código de una página del panel de administración . . . . .	124
37.	Filtro creado para mostrar los <i>fandoms</i> . . . . .	125
38.	<i>Widgets</i> personalizados . . . . .	125
39.	<i>Mocks</i> en un método a probar . . . . .	127
40.	Validador creado para comprobar la edad de un usuario . . . . .	128
41.	Decorador creado . . . . .	129
42.	Autenticación y obtención del código HTML de una página web	130
43.	Datos juntos de los <i>fanfics</i> . . . . .	132
44.	Método para preprocesar los datos . . . . .	133
45.	Recomendaciones de un <i>fanfic</i> . . . . .	133
46.	Gráfico . . . . .	134
47.	Gráfico . . . . .	135
48.	Planificación temporal real que ha cambiado . . . . .	151
49.	Botón “registrarse” . . . . .	157
50.	Varios campos del formulario de registro . . . . .	157
51.	Botón “iniciar sesión” . . . . .	158
52.	Formulario de inicio de sesión . . . . .	159
53.	Pantalla de recuperar contraseña . . . . .	160
54.	Opciones de los usuarios . . . . .	162
55.	Panel de administración . . . . .	162
56.	Panel de administración . . . . .	164
57.	Añadir capítulo . . . . .	164
58.	Panel de administración . . . . .	167
59.	Añadir personaje a un <i>fanfic</i> . . . . .	167
60.	Añadir personaje . . . . .	168
61.	Panel de administración . . . . .	170
62.	Añadir <i>fandom</i> . . . . .	170

63.	Panel de administración . . . . .	172
64.	Añadir <i>fanfic</i> . . . . .	172
65.	Panel de administración . . . . .	173
66.	Panel de administración . . . . .	174
67.	Opciones de los usuarios . . . . .	175
68.	Opciones de los usuarios . . . . .	176
69.	Opciones de los usuarios . . . . .	178
70.	Formulario para añadir una nueva lista . . . . .	178
71.	Borrar listas . . . . .	178
72.	Cambiar el nombre de listas . . . . .	179
73.	Icono notificaciones . . . . .	181
74.	Marcar como leído . . . . .	181
75.	Marcar como leído . . . . .	181
76.	Opciones de los usuarios . . . . .	182
77.	Panel lateral del perfil de un usuario . . . . .	183
78.	Opciones de los usuarios . . . . .	185
79.	Configuración de la cuenta del usuario . . . . .	185
80.	Crear copia de seguridad . . . . .	186
81.	Reseña de un miembro de la aplicación . . . . .	187
82.	Panel lateral del perfil de un usuario . . . . .	188
83.	Panel lateral del perfil de un usuario . . . . .	189
84.	Botón seguir a un usuario . . . . .	190
85.	Botón dejar de seguir a un usuario . . . . .	190
86.	Botón para reportar error . . . . .	191
87.	Formulario para notificar de un error en el <i>fanfic</i> . . . . .	192
88.	Categorías <i>fanfics</i> . . . . .	194
89.	<i>Fandoms</i> de una categoría . . . . .	194
90.	Filtros para buscar <i>fanfics</i> . . . . .	195
91.	Parte del menú superior . . . . .	196
92.	Página mis fanfics . . . . .	197
93.	Cuadro de búsqueda . . . . .	198
94.	Opciones de búsqueda . . . . .	199

95.	Añadir <i>fanfic</i> . . . . .	200
96.	Un <i>fanfic</i> . . . . .	202
97.	Gestionar capítulos de un <i>fanfic</i> . . . . .	203
98.	Notas privadas en los capítulos . . . . .	204
99.	Formulario nueva reseña . . . . .	206
100.	Una reseña . . . . .	206
101.	Añadir <i>fanfic</i> a una lista . . . . .	207
102.	Cambiar <i>fanfic</i> de lista . . . . .	208
103.	Un <i>fanfic</i> . . . . .	209
104.	Añadir tarea para importar <i>fanfic</i> . . . . .	210
105.	IU_Fig1 . . . . .	211
106.	IU_Fig2 . . . . .	211
107.	Iniciar importación de <i>fanfic</i> externo . . . . .	212
108.	Subdiagrama scrapear y añadir <i>fanfic</i> externo . . . . .	213
109.	Comprobar si hay capítulos nuevos . . . . .	214
110.	Entrenar sistema de recomendación de <i>fanfics</i> . . . . .	215
111.	Subdiagrama de entrenar sistema de recomendación de <i>fanfics</i> . . . . .	216
112.	Prototipo de registrarse . . . . .	217
113.	Prototipo de iniciar sesión y recuperar contraseña . . . . .	218
114.	Prototipo de configuración del perfil . . . . .	219
115.	Prototipo de dashboard y perfil . . . . .	220
116.	Prototipo de <i>fanfics</i> propios . . . . .	221
117.	Prototipo de explorar <i>fanfics</i> y ver página de <i>fanfic</i> . . . . .	222
118.	Prototipo de añadir <i>fanfic</i> externo . . . . .	223
119.	Prototipo de la página de aterrizaje . . . . .	224
120.	Prototipo de la identificación . . . . .	225
121.	Prototipo de recuperar contraseña . . . . .	226
122.	Prototipo de registrarse . . . . .	227
123.	Prototipo del dashboard . . . . .	228
124.	Prototipo de añadir <i>fanfics</i> . . . . .	229
125.	Prototipo de explorar <i>fanfics</i> . . . . .	230
126.	Prototipo de explorar <i>fanfics</i> de un <i>fandom</i> . . . . .	231

127. Prototipo de ver la página de un <i>fanfic</i> . . . . .	232
128. Prototipo de los <i>fanfics</i> propios . . . . .	233
129. Prototipo de la página del perfil . . . . .	234
130. Prototipo de la configuración de la cuenta . . . . .	235
131. Prototipo de la página de aterrizaje (I) . . . . .	236
132. Prototipo de la página de aterrizaje (II) . . . . .	237
133. Instalación de Visual Studio . . . . .	239

## Índice de tablas

1.	Tarea Planteamiento de la aplicación . . . . .	38
2.	Tarea Nombrar la aplicación . . . . .	38
3.	Tarea Planificación de Tareas . . . . .	39
4.	Tarea Planificación temporal . . . . .	39
5.	Tarea Elegir lenguaje de programación . . . . .	39
6.	Tarea Elegir software . . . . .	40
7.	Tarea Preparación entorno de desarrollo . . . . .	40
8.	Tarea Reuniones con el director del proyecto . . . . .	40
9.	Tarea Aprendizaje de modularidad en Python . . . . .	41
10.	Tarea Aprendizaje de Django . . . . .	41
11.	Tarea Aprendizaje de CSS Flexbox . . . . .	42
12.	Tarea Aprendizaje de CSS Grid . . . . .	42
13.	Tarea Aprendizaje de UI . . . . .	43
14.	Tarea Aprendizaje de UX . . . . .	43
15.	Tarea Aprendizaje de Ficwad.com . . . . .	43
16.	Tarea Aprendizaje de AvengersFanfiction.com . . . . .	44
17.	Tarea Aprendizaje de ArchiveOfOurOwn.org . . . . .	44
18.	Tarea Listar funcionalidades . . . . .	45
19.	Tarea Plantear esquema web . . . . .	45
20.	Tarea Diseño prototipos low-res . . . . .	45
21.	Tarea Diseño prototipos high-res . . . . .	46
22.	Tarea Jerarquía de actores . . . . .	46
23.	Tarea Casos de uso . . . . .	46
24.	Tarea Modelo de dominio . . . . .	47
25.	Tarea Diagramas de uso . . . . .	47
26.	Tarea Diagramas de secuencia . . . . .	48
27.	Tarea Diagramas de la base de datos . . . . .	48
28.	Tarea Registro e identificación de usuarios . . . . .	49
29.	Tarea Página de aterrizaje . . . . .	49
30.	Tarea Añadir fanfic externo . . . . .	49

31.	Tarea Cambiar configuración de la cuenta . . . . .	50
32.	Tarea Ver perfil de usuario . . . . .	50
33.	Tarea Ver página de fanfic . . . . .	50
34.	Tarea Comentar en fanfics . . . . .	51
35.	Tarea Añadir notas a capítulos de los fanfics . . . . .	51
36.	Tarea Explorar fanfics . . . . .	51
37.	Tarea Filtrar Fanfics . . . . .	52
38.	Tarea Clasificar Fanfics . . . . .	52
39.	Tarea Ver mis fanfics . . . . .	52
40.	Tarea Gestionar listas del usuario . . . . .	53
41.	Tarea Dashboard . . . . .	53
42.	Tarea Registrar capítulos leídos . . . . .	53
43.	Tarea Buscar fanfics y usuarios . . . . .	54
44.	Tarea Avisar de nuevos capítulos . . . . .	54
45.	Tarea Crear copias de seguridad . . . . .	54
46.	Tarea Funciones administrativas . . . . .	55
47.	Tarea Ciencia de datos . . . . .	55
48.	Tarea Pruebas del registro y log in de usuarios . . . . .	56
49.	Tarea Pruebas de la página de aterrizaje . . . . .	56
50.	Tarea Pruebas de Añadir fanfic externo . . . . .	56
51.	Tarea Cambiar configuración de la cuenta . . . . .	57
52.	Tarea Ver perfil de usuario . . . . .	57
53.	Tarea Ver página de fanfic . . . . .	57
54.	Tarea Pruebas de Comentar en fanfics . . . . .	58
55.	Tarea Pruebas de añadir notas a capítulos de los fanfics . . . . .	58
56.	Tarea Explorar fanfics . . . . .	58
57.	Tarea Pruebas de Filtrar fanfics . . . . .	59
58.	Tarea Pruebas de clasificar fanfics . . . . .	59
59.	Tarea Pruebas de ver mis fanfics . . . . .	59
60.	Tarea Pruebas de gestionar listas del usuario . . . . .	60
61.	Tarea Pruebas del dashboard . . . . .	60
62.	Tarea Pruebas de Registrar capítulos leídos . . . . .	60



63.	Tarea Pruebas de Buscar fanfics y usuarios . . . . .	61
64.	Tarea Pruebas de Avisar de nuevos capítulos . . . . .	61
65.	Tarea Pruebas de Crear copia de Seguridad . . . . .	61
66.	Tarea Pruebas de Funciones administrativas . . . . .	62
67.	Tarea Pruebas de ciencia de datos . . . . .	62
68.	Tarea Redacción de la Memoria del Proyecto . . . . .	63
69.	Tarea Redacción de Manuales . . . . .	63
70.	Tarea Preparación de la defensa . . . . .	63
71.	Lista de Tareas resumidas . . . . .	64
72.	Gastos totales . . . . .	74
73.	Lista de Tests . . . . .	136
74.	Lista de Tareas resumidas . . . . .	147
75.	Caso de uso extendido “Registrarse” . . . . .	156
76.	Caso de uso extendido “Identificarse” . . . . .	158
77.	Caso de uso extendido “Recuperar contraseña” . . . . .	160
78.	Caso de uso extendido “Ejecutar funciones administrativas” .	161
79.	Caso de uso extendido “Gestionar capítulos” . . . . .	163
80.	Caso de uso extendido “Gestionar personajes” (I) . . . . .	165
81.	Caso de uso extendido “Gestionar personajes” (II) . . . . .	166
82.	Caso de uso extendido “Gestionar <i>fandoms</i> ” . . . . .	169
83.	Caso de uso extendido “Gestionar <i>fanfics</i> ” . . . . .	171
84.	Caso de uso extendido “Administrar reseñas” . . . . .	173
85.	Caso de uso extendido “Gestionar errores” . . . . .	174
86.	Caso de uso extendido “Ver estadísticas e información de Cien- cia de Datos” . . . . .	175
87.	Caso de uso extendido “Cerrar sesión” . . . . .	176
88.	Caso de uso extendido “Gestionar listas” . . . . .	177
89.	Caso de uso extendido “Leer notificaciones” . . . . .	180
90.	Caso de uso extendido “Ver perfil propio” . . . . .	182
91.	Caso de uso extendido “Ver fanfics de un usuario” . . . . .	183
92.	Caso de uso extendido “Cambiar la configuración de la cuenta”	184
93.	Caso de uso extendido “Exportar copia de seguridad” . . . . .	186

94. Caso de uso extendido “Ver perfil usuario” . . . . .	187
95. Caso de uso extendido “Ver usuarios seguidos” . . . . .	188
96. Caso de uso extendido “Ver seguidores” . . . . .	189
97. Caso de uso extendido “Gestionar seguimiento” . . . . .	190
98. Caso de uso extendido “Enviar error” . . . . .	191
99. Caso de uso extendido “Explorar fanfics” . . . . .	193
100. Caso de uso extendido “Ver mis fanfics” . . . . .	196
101. Caso de uso extendido “Buscar fanfics” . . . . .	198
102. Caso de uso extendido “Añadir fanfic externo” . . . . .	200
103. Caso de uso extendido “Ver página de fanfic individual” . . . . .	201
104. Caso de uso extendido “Gestionar capítulos leídos” . . . . .	203
105. Caso de uso extendido “Gestionar notas privadas” . . . . .	204
106. Caso de uso extendido “Gestionar reseñas” . . . . .	205
107. Caso de uso extendido “Gestionar <i>fanfic</i> en lista” . . . . .	207
108. Caso de uso extendido “Marcar último capítulo de un fanfic como leído” . . . . .	209

### 3. Definiciones y Vocabulario Técnico

- **Fanfic o Fanfiction:** Historias creadas a partir de obras de ficción, es decir, historias que se centran en los personajes y mundos de otras novelas. Este tipo de historias suelen estar escritas por seguidores de la obra original, que desean darle un giro a la trama, o explorar el mundo creado.
- **Fandom:** Representa la obra original en la que está basado el *fanfic*. *Harry Potter* y *El señor de los anillos* son ejemplos de *fandom*.
- **Crossover:** *Fanfic* que cuenta con elementos de distintos *fandoms*. Ejemplo: historia sobre *Harry Potter* en *Narnia*.
- **Pairing:** Se refiere a la relación amorosa o de amistad entre dos personajes de un *fanfic*.
- **Framework:** Estructura básica implementada sobre un lenguaje de programación para facilitar el desarrollo de aplicaciones.
- **Ciencia de datos o data science:** La habilidad de obtener datos, entenderlos, procesarlos, visualizarlos y extraer información de ellos.
- **Reseña:** Opinión de un usuario respecto de un producto, en este caso, los *fanfics*.
- **Scrape or Scraping:** Proceso de extraer datos de las páginas web utilizando su código fuente.
- **Panel de administración:** Página web que cuenta con todas las herramientas de administración centralizadas.
- **Business Intelligence:** Proceso que ayuda a analizar e interpretar datos históricos y a tomar decisiones de negocio.
- **HTTP:** También *Hyper Text Transfer Protocol*, es un protocolo de comunicación entre clientes y servidores.

- **Ajax:** Técnica que permite que una página web se actualice de forma asíncrona intercambiando información con el servidor en segundo plano. Su acrónimo significa “Asynchronous JavaScript and Xml”.
- **Websocket:** Protocolo que permite crear un canal bidireccional de comunicación entre un servidor y cliente. El servidor podrá enviar datos al cliente sin que este último haya enviado nada.
- **Editor WYSIWYG:** También *What You See Is What You Get*, se refiere a aquellos editores que permiten ver el resultado final de lo escrito.
- **Diseño responsive:** Una manera de diseñar para que las páginas web sean fácilmente vistas y usadas en cualquier tipo de dispositivo sin importar el tamaño de la pantalla [6].
- **Rating:** Sistema de clasificación por edad para conocer la audiencia recomendada para ver o leer cierta obra.
- **UI:** También conocido como “User Interface”, hace referencia a la parte gráfica de la aplicación.
- **UX:** También “User Experience”, se refiere a la experiencia que tiene el usuario al interactuar con la aplicación web.

## 4. Introducción

El Trabajo de Fin de Grado (también llamado TFG) aquí presentado fue una idea propia. La falta de interés en el resto de trabajos disponibles propició la elección de un TFG distinguido, además de útil en lo que se refiere al ámbito personal.

Mediante este proyecto se planea unir dos elementos que aparentemente solo tienen el *entretenimiento* en común, cómo bien son la lectura y la tecnología. La tecnología puede aplicarse a la lectura de formas impredecibles. Por ejemplo, un uso predecible serían los libros electrónicos (*ej. Amazon Kindle, Nook,...*) como medio para leer.

La realización de este proyecto supone una forma de ampliar exponencialmente lo estudiado durante el *Grado en Ingeniería Informática de Gestión y Sistemas de Información* respecto al diseño de aplicaciones web, y ver de forma práctica el resultado de aplicar conceptos relacionados con la *ciencia de datos*.

### 4.1. Descripción y situación del trabajo

Los *fanfics* son historias creadas a partir de obras de ficción, es decir, son historias que se centran en los personajes y mundos de otras novelas. Este tipo de historias suelen estar escritas por seguidores de la obra original, que desean darle un giro a la trama, o explorar el mundo creado.

Conociendo el significado de esta palabra, el objetivo principal de este trabajo consiste en crear una aplicación web donde los usuarios puedan realizar un seguimiento de los *fanfics* que planean leer, que han terminado de leer o están leyendo actualmente entre otros. Se dispondrá de listas en las que poder clasificar éstos para una mejor organización.

Los usuarios serán avisados de nuevos capítulos de sus *fanfics* favoritos en una sola página sin tener que visitar historia por historia. Es decir, la información de los relatos estará centralizada a disposición de los miembros del sitio web.

Los *fanfics* podrán ser organizados según la obra literaria en la que estén basados y los personajes en los que estén centrados. Se tendrá la posibilidad de añadir notas privadas a los capítulos, y los usuarios tendrán conocimiento en cada momento de los capítulos que les faltan por leer o cual fue el último que leyeron. Además, se podrá opinar dejando comentarios o reseñas, y puntuando.

Se recuerda que dichas historias no se encontrarán almacenadas en la aplicación web pensada, sino que ésta actuará de gestor. Por lo que se utilizarán técnicas como el *scraping* para recolectar la información de los *fanfics* que los usuarios añadan y planeen seguir. Se obtendrán datos como el título y el autor de las historias; el número y nombre de los capítulos actuales; y el nombre de los personajes que forman parte. Asimismo, esta técnica será de utilidad en la posterior comprobación de la existencia de nuevos capítulos.

Cuando un nuevo capítulo esté disponible, se avisará al usuario, siempre y cuando tenga la historia añadida en una de sus listas. También se usarán notificaciones para alertar al usuario de la ocurrencia de ciertas acciones.

Toda la información de lectura aportada por el usuario podrá ser exportada fácilmente de modo que en ningún momento se tema la pérdida de datos.

Los administradores y un grupo especial de bibliotecarios contarán con acciones extra que les permitirán editar ciertos aspectos de la aplicación no disponibles para los miembros normales. Estos usuarios tendrán acceso a un panel administrativo en el que poder aplicar estas acciones extra mencionadas.

Y por último, se usarán técnicas de la *Ciencia de Datos* para ofrecer recomendaciones de *fanfics* similares a los usuarios, así como para explorar los datos recolectados y poder ofrecer:

- Estimaciones de los *fandoms* más populares del momento.
- Estimaciones de los *fandoms* más populares del momento según el género.

Dichos análisis de datos hacen un guiño al campo *Business Intelligence* y serán de utilidad para todas aquellas personas que les interese conocer cual es la “moda” del momento.

## 4.2. Planteamiento del problema

Con tanto contenido multimedia a disposición del público es cada vez más difícil controlar las series que se han visto, los libros pendientes de leer y los que se han leído.

A consecuencia de esto, en las últimas décadas han aparecido herramientas como *Tvtime.com* que posibilitan el poder gestionar contenido multimedia, permitiendo marcar como vistos los episodios de las series. De esta forma se consigue un mayor control sobre lo que se ve.

Con este aumento de contenido multimedia se ha hecho más popular la aparición del *fanfic* o *fanfiction*. Estos relatos ficticios suelen encontrarse en páginas como *Fanfiction.net* (Figura 1) y *Wattpad.com* (Figura 2) aunque no existe ninguna centralización de las historias. Administrar y estar al tanto de su actualización puede ser más que complicado si el volumen seguido de fanfics es alto.

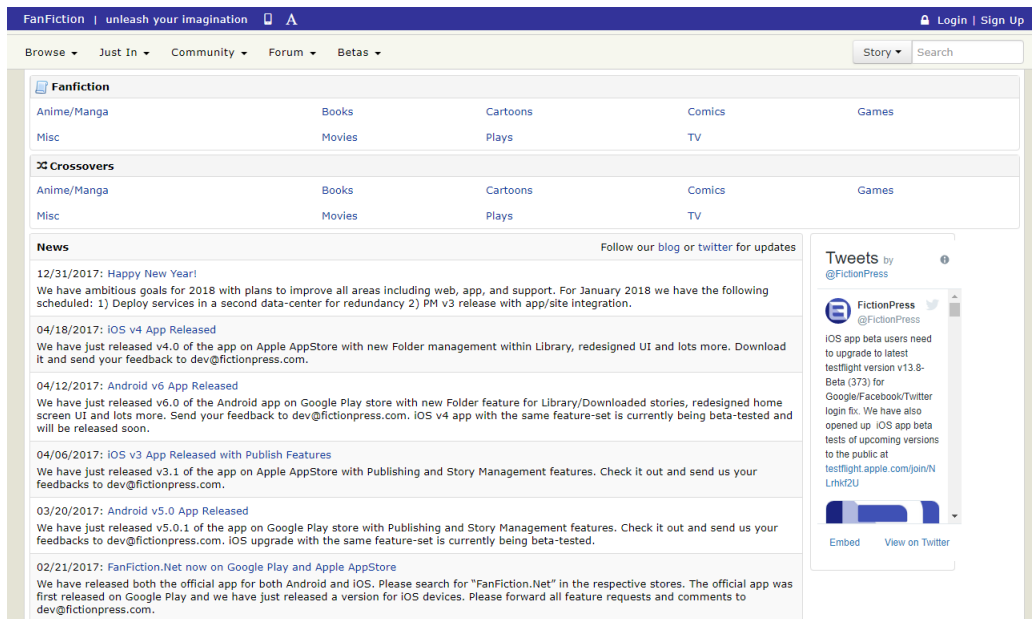


Figura 1: Página principal de *Fanfiction.net*



Figura 2: Página principal de *Wattpad.com*



Desde notebooks hasta listas en editores de texto pasando por el uso de la página de favoritos en los navegadores. Cualquier herramienta sirve para apuntar las historias ficticias que planeamos leer o estamos leyendo, pero en grandes cantidades y sin ninguna herramienta específica para ello, esto se puede volver abrumador.

### 4.3. Justificación del TFG

Han sido varias las razones que me han llevado a elegir este proyecto como *Trabajo de Fin de Grado*.

#### 4.3.1. Aprendizaje

Es una oportunidad tener tiempo suficiente dentro del año escolar universitario para aprender nuevas tecnologías como bien son *Python* y *Django*. Este proyecto me permite lo anterior sin albergar ningún sentimiento de culpa respecto al uso de tiempo en su realización.

Por otro lado, este trabajo se va a realizar de forma paralela a unas prácticas de empresa en las que utilizaré *Python* y *Django*. El haber elegido estas tecnologías será beneficioso a lo largo de la realización del proyecto, ya que todas las mañanas se pasarán al menos 4 horas utilizando dicho framework.

También es un paso importante en la adquisición de nuevos conocimientos hasta ahora no dados o vistos muy brevemente. Además de demostrar la capacidad de aprendizaje sin la necesidad constante de un docente.

### 4.3.2. Curiosidad

Siempre he tenido curiosidad por los *frameworks* porque me han parecido herramientas sofisticadas y complejas. Después de investigar en internet las distintas posibilidades (*ej. Django, Laravel, Ruby On Rails,...*) me empecé a hacer preguntas: ¿Puedes programar tareas en un framework? Y la base de datos, ¿cómo funciona? ¿Se pueden enviar mensajes específicamente a un usuario?

Después de decantarme por Django realicé una breve investigación que solo consiguió que me surgieran más preguntas y más ganas de comprender esta herramienta.

### 4.3.3. Interés personal

Esta no es una aplicación que planeo dejar en el cajón una vez terminada. Es más, es probable que la termine utilizando día a día y la propague por ciertos grupos de Internet para que personas con las mismas necesidades puedan beneficiarse de ella.

Ahora mismo utilizo los marcadores del navegador *Google Chrome* para realizar un seguimiento de los *fanfics* que leo. La situación se está volviendo insostenible debido al gran volumen de historias que sigo y que he seguido.

Es imposible realizar una comprobación diaria de las historias para ver si han sido actualizadas, por dos razones: falta de tiempo, no se puede estar como mínimo una hora comprobando si los autores han publicado nuevos capítulos; y de ganas, después de un día ajetreado no apetece ponerse a buscar qué *fanfics* puedes leer por la existencia de nuevos capítulos.

#### **4.3.4. Diversión**

Realizar una aplicación web siempre me ha parecido divertido. Juntando eso con la lectura, una actividad que realizo diariamente; y el aprendizaje de nuevas tecnologías, se podría decir que es el proyecto perfecto para mi.

## 5. Planteamiento Inicial

En esta sección del documento se expondrán los objetivos, se mostrará el alcance, la planificación temporal, las herramientas a usar, la arquitectura a seguir, la gestión de riesgos y la evaluación económica.

### 5.1. Objetivos

El objetivo principal del proyecto consiste en diseñar e implementar una aplicación web que permita al usuario organizar *fanfics* y visualizar de forma sencilla la existencia de nuevos capítulos publicados.

Otros objetivos son:

- Ampliar los conocimientos sobre redes y sistemas web obtenidos a lo largo de la carrera.
- Aprender a desarrollar aplicaciones web mediante el uso de *frameworks* como Django.
- Coger soltura en el uso de Python como lenguaje de programación.
- Obtener un conocimiento básico sobre los mecanismos de diseño de interfaces (*UI - User Interface*) y diseño de experiencias de usuario (*UX - User Experience*).
- Aprender a analizar de forma práctica un conjunto de datos mediante el uso de herramientas de *Ciencia de Datos*.

#### 5.1.1. Objetivos técnicos

Los objetivos técnicos principales serán los siguientes:

- Contar con la opción de ser bibliotecario para poder ejecutar funciones administrativas que ayuden a una buena gestión y organización de la información de la aplicación.
- Posibilidad de añadir, modificar y eliminar elementos como capítulos, personajes, *fanfics* y *fandoms*, entre otros, siempre y cuando se cuente con el rango de bibliotecario.
- Ver estadísticas e información a través de los datos recopilados por la aplicación haciendo uso de herramientas de *ciencia de datos*.
- Poder crear, modificar y eliminar listas para mantener los *fanfics* seguidos bien organizados.
- Recibir notificaciones cada vez que se de un evento relacionado con el usuario.
- Posibilidad de cambiar el nivel de privacidad desde la configuración de la cuenta.
- Proporcionar una manera de realizar copias de seguridad de los datos del usuario.
- Posibilidad de explorar los distintos *fandoms* y *fanfics*, además de poder visualizar estos junto con sus correspondientes reseñas. También se podrá llevar un seguimiento de los capítulos leídos, así como añadir notas personales a estos.
- Automatizar la importación de *fanfics* al sistema, así como la comprobación de nuevos capítulos.
- Implementar un sistema de recomendación de *fanfics* que permita a usuario descubrir de forma sencilla *fanfics* similares a los leídos.

## 5.2. Arquitectura

En este apartado, se explica cómo es la arquitectura que usará la aplicación web, y el proceso que se lleva a cabo desde que un cliente realiza una petición hasta que recibe una respuesta.

Antes de nada, es necesario saber que se ha utilizado un *framework* de *Python* llamado *Django*, y herramientas como *Celery* y *Django Channels*. Este conjunto de herramientas afectará notablemente a la arquitectura final del sistema.

Cuando un usuario intenta acceder a una de las páginas de la aplicación, un servidor de tipo **ASGI** interceptará la petición. ASGI (*Asynchronous Server Gateway Interface*) es una interfaz posicionada entre un servidor, y una aplicación de *Python*, que permite manejar diferentes tipos de protocolos (*HTTP*, *HTTP2* y *WebSocket*). [3]

### 5.2.1. Django

El usuario al intentar acceder a una página estará usando el protocolo *HTTP*, y *Django* será el encargado de gestionar y procesar las peticiones con este protocolo.

Este *framework* utiliza el patrón **MTV (Modelo-Plantilla-Vista)**, “*un patrón de software, similar en estilo a MVC (Modelo-Vista-Controlador), pero una mejor descripción de la forma en que Django funciona.*” [2]

El anterior patrón explicado:

- El término **Modelo** (*Model* en inglés), hace referencia a los datos persistentes de la aplicación. Esta capa permite añadir, modificar y eliminar información de una base de datos de forma sencilla.

- El segundo término **Plantilla** (*Template* en inglés), se refiere a los archivos en formato *HTML* utilizados para mostrar una respuesta al cliente. Estos ficheros pueden contener código en formato *Python*, gracias al soporte de Django.
- Por último, el término **Vista** (*View* en inglés), se encarga de ejecutar la lógica correspondiente asignada a cada URL, de decidir la plantilla a utilizar, así como los datos a mostrar en esta. En resumen, actúa de enlace entre el Modelo y la Plantilla.

El primer paso que realiza *Django*, es traducir la dirección web pedida para saber que funcionalidad debe ejecutar. Aquí entrará el módulo **url resolver**, encargado de leer las *URLs* y llamar a la Vista correspondiente.

### 5.2.2. WebSockets

Anteriormente se ha comentado la existencia de un sistema de notificaciones en nuestra aplicación. Hasta ahora era común realizar este tipo de funcionalidad mediante llamadas *Ajax* y un método llamado *Long-Polling* que consiste en realizar peticiones cada cierto tiempo a un servidor para comprobar si se han producido cambios.

Se va a intercambiar ese método por el uso de **WebSockets**. Este protocolo permite mantener abierta una conexión entre un servidor y un cliente, consintiendo una conexión bidireccional, en la que un servidor puede enviar datos a un cliente, y viceversa.

Las peticiones de tipo *WebSocket* recibidas por el servidor ASGI serán gestionadas por la librería *Django Channels*, a través de *consumers*. La equivalencia de un *consumer* son las vistas en *Django*. Responden a peticiones, o envían datos, puesto que la conexión es bidireccional.

### 5.2.3. Tareas asíncronas

Las tareas que se quieran ejecutar de forma asíncrona serán inicializadas por alguna vista, y después serán gestionadas por un *broker*. Un *broker* es simplemente una cola. Además, los *workers* serán los encargados de seleccionar las tareas del *broker* a ejecutar, y ejecutarlas.

La librería *Celery* implementa el proceso anterior, asimismo permite programar las tareas asíncronas para que se ejecuten a cierta hora, y permite tareas periódicas.

### 5.2.4. Resumen

Lo comentado a lo largo de este apartado se puede resumir en la Figura 3.

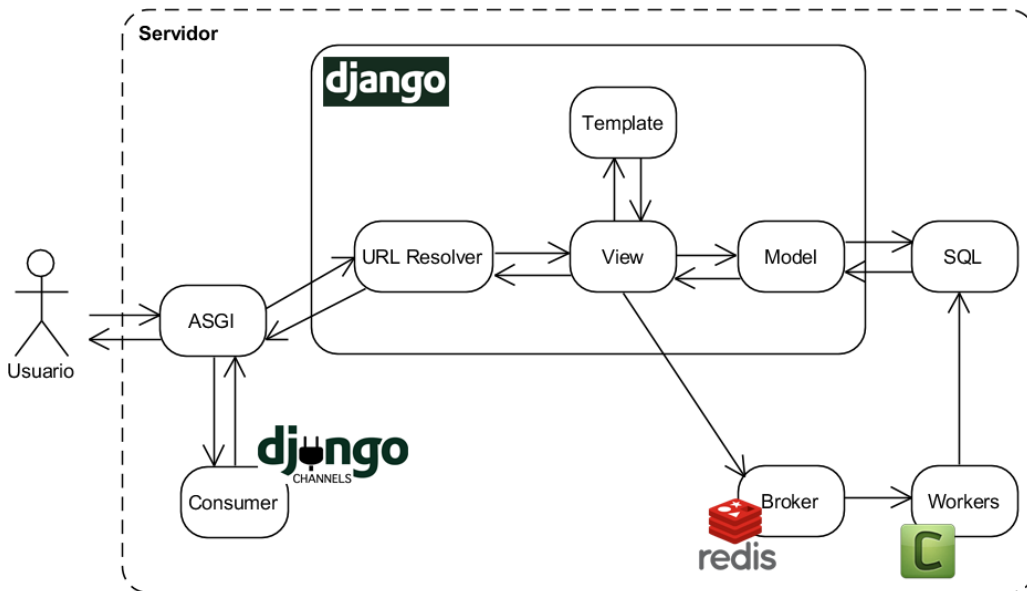


Figura 3: Arquitectura de la aplicación web



### 5.3. Herramientas

En este apartado se detallan las herramientas a usar en el proyecto, así como una breve descripción del lugar donde se utilizarán.

- **GitLab:** Herramienta online que implementa el sistema de control de versiones, además de permitir la creación de repositorios privados gratuitos. Se creará un repositorio privado para almacenar el código del proyecto. Página oficial: <https://gitlab.com/>
- **GitHub Desktop:** Interfaz gráfica para “comunicarse” con *GitLab*, y poder guardar cambios en el código. Página oficial: <https://desktop.github.com/>
- **Django:** Es un *framework* de *Python* creado para simplificar la implementación de aplicaciones web. Usa el patrón de diseño conocido como Modelo-Plantilla-Vista (MTV). La aplicación web se realizará con la versión 2.2 de esta herramienta. Página oficial: <https://www.djangoproject.com/>
- **Python:** Lenguaje de programación de alto nivel en el que está basado el *framework* de Django. Se usará a lo largo de la implementación.
- **Django Channels:** Librería adicional de *Django* que acepta peticiones de tipo *WebSocket*. Se utilizará para la implementación del sistema de notificaciones de la aplicación. Página oficial: <https://channels.readthedocs.io/en/latest/>
- **Django Extensions:** Librería adicional de *Django* que incluye varias funcionalidades útiles, como poder ver todas las URLs en activo de la aplicación. Página oficial: <https://django-extensions.readthedocs.io/en/latest/>
- **Django Countries:** Extensión creada para *Django* que facilita un listado de países. Se usará en varios formularios de la aplicación. Página oficial: <https://github.com/SmileyChris/django-countries>

- **Widget Tweaks:** Permite la manipulación de los campos que *Django* genera automáticamente para los formularios. Se podrán añadir estilos y atributos a elementos HTML. Página oficial: <https://github.com/jazzband/django-widget-tweaks>
- **Celery:** Esta herramienta permite poner en cola múltiples tareas, para que sean ejecutadas de forma automática y asíncrona respecto de los usuarios. Es interesante para ser capaces de ejecutar ciertas acciones de forma periódica. Página oficial: [Página oficial: http://www.celeryproject.org/](http://www.celeryproject.org/)
- **Redis:** Estructura de datos utilizada como base de datos, caché y “message broker”. Se utilizará junto con *Celery* para poder ejecutar tareas asíncronas. Página oficial: <https://redis.io/>
- **Django Celery Results:** Extensión que permitirá guardar y visualizar el resultado de las tareas asíncronas realizadas. Página oficial: <https://github.com/celery/django-celery-results>
- **WSL (Windows Subsystem for Linux):** Este subsistema permite ejecutar herramientas de GNU/Linux mediante una consola de comandos directamente desde Windows. Se utilizará para instalar *Redis*. Página oficial: <https://docs.microsoft.com/es-es/windows/wsl>
- **HTML:** También llamado *Hyper Text Markup Language*, es un lenguaje estándar para la creación de páginas web. Mediante etiquetas o *tags* se forman elementos que son interpretados por los navegadores. Este lenguaje se usará para la creación de las distintas páginas que formarán la aplicación final.
- **CSS:** Su nombre completo es *Cascading Style Sheets*. Permite describir como los elementos creados en *HTML* son visualizados, es decir, define sus estilos y su apariencia en pantalla.
- **Bulma:** *Framework* de CSS basado en *Flexbox* utilizado para dar estilo a componentes HTML. Página oficial: <https://bulma.io/>

- **FontAwesome:** Paquetería de iconos usada para obtener iconos de libre uso. Los iconos se utilizarán en varios elementos HTML. Página oficial: <https://fontawesome.com/>
- **unDraw:** Aplicación web que contiene un gran número de ilustraciones de libre uso. Página oficial: <https://undraw.co/illustrations>
- **jQuery:** Librería de *Javascript* utilizada en varios archivos del *frontend*. Página oficial: <https://jquery.com/>
- **Javascript:** Lenguaje de programación usado en el lado del cliente. Se utilizará junto con la librería *jQuery*.
- **jQuery SlimScroll:** Componente realizado con *jQuery* que permite transformar el estilo de una barra de desplazamiento. Se usará en el contenedor de las notificaciones. Página oficial: <https://github.com/rochal/jquery-slimScroll>
- **Pell:** Simple editor WYSIWYG web para la creación y edición de reseñas de un *fanfic*. Página oficial: <https://github.com/jaredreich/pell>
- **Selectize.js:** Librería que permite personalizar los elementos *select* de HTML, permitiendo así realizar búsquedas. Se utilizará en varios puntos del panel de administración, así como en múltiples formularios de la aplicación. Página oficial: <https://selectize.github.io/selectize.js/>
- **Tippy:** Librería que proporciona *tooltips* y *popovers*. Se usará para aportar información adicional al usuario cuando este coloque el ratón en ciertas partes de la aplicación. Página oficial: <https://atomiks.github.io/tippyjs/>
- **Toastr:** Librería usada para generar mensajes emergentes por pantalla. Usada generalmente para mostrar *feedback* al usuario en respuesta a llamadas *Ajax*. Página oficial: <https://codeseven.github.io/toastr/>

- **Sklearn:** Librería que contiene múltiples herramientas para la realización de tareas de *data science*. Se utilizará en la realización del sistema de recomendaciones de *fanfics*. Página oficial: <https://scikit-learn.org/stable/>
- **Pandas:** Facilita la manipulación de datos mediante estructuras llamadas *dataframes* y *series*. Se utilizará en la realización del sistema de recomendaciones de *fanfics*. Página oficial: <https://pandas.pydata.org/>
- **Matplotlib:** Librería de *Python* que permite generar gráficos. Se usará para visualizar la información sobre los datos recogidos de los usuarios. Página oficial: <https://codeseven.github.io/toastr/>
- **Beautiful Soup:** Librería de *Python* que permite *scrapear* páginas web. Se usará para obtener impotar *fanfics* y comprobar la existencia de nuevos capítulos. Página oficial: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- **Gantt Project:** Programa usado para generar la planificación de las tareas. También se usará para llevar un control de lo que se ha hecho, lo que se está haciendo y lo que queda por hacer. Página oficial: <https://www.ganttproject.biz/>
- **Google Drive:** Aplicación web de almacenamiento online que permite realizar sincronizaciones de datos con archivos locales. Se usará para almacenar todo lo que tenga que ver con la documentación del trabajo. Página oficial: [https://www.google.com/intl/es\\_ALL/drive/](https://www.google.com/intl/es_ALL/drive/)
- **Adobe XD:** Herramienta del paquete de *Adobe* especializada en la creación de prototipos de aplicaciones. Los prototipos se realizarán con este programa. Página oficial: <https://www.adobe.com/es/products/xd.html>
- **JetBrains Pycharm:** Entorno de desarrollo que se usará para todo lo relacionado con la parte de la implementación del proyecto. Página oficial: <https://www.sublimetext.com/>

- **XAMPP:** Varios servicios combinados que permitirán crear una base de datos local, además de poder acceder ella mediante una consola de comandos integrada. Página oficial: <https://www.apachefriends.org/es/index.html>
- **HeidiSQL:** Programa que permite ver, editar, insertar y borrar datos de forma visual en una base de datos. Se usará junto con XAMPP para editar los datos de las bases de datos locales. Página oficial: <https://www.heidisql.com/>
- **Visual Paradigm:** Herramienta de modelado de diagramas usada en varias zonas del proyecto. Se han realizado diagramas de flujo y de clases entre otros. Página oficial: <https://www.visual-paradigm.com/>
- **Overleaf:** Aplicación web que usa la tecnología *latex* para la edición de la memoria del trabajo. Página oficial: <https://www.overleaf.com>
- **Clockify:** Herramienta online que permite registrar la cantidad de horas dedicadas a una tarea. Se usará para poder comparar las horas estimadas de cada tarea con las horas reales finalmente empleadas. Página oficial: <https://clockify.me>
- **Jupyter Notebook:** Libreta que permite mezclar texto con código de *Python* ejecutable. Se ha utilizado para realizar comprobaciones internas en el proceso de creación del sistema de recomendaciones de *fanfics*. Página oficial: <https://jupyter.org/>
- **SendGrid:** Servidor de correo electrónico utilizado en la recuperación de la contraseña de los usuarios de la aplicación. Página oficial: <https://sendgrid.com/>
- **Coverage:** Librería de *Python* que permite conocer el porcentaje de código cubierto por los tests realizados.

## 5.4. Alcance

El alcance del proyecto se ha definido gracias al uso de un diagrama EDT. Se han definido las tareas necesarias para llevar a cabo el proyecto, y a continuación, se han establecido siete grupos principales que se pueden observar en la Figura 4.

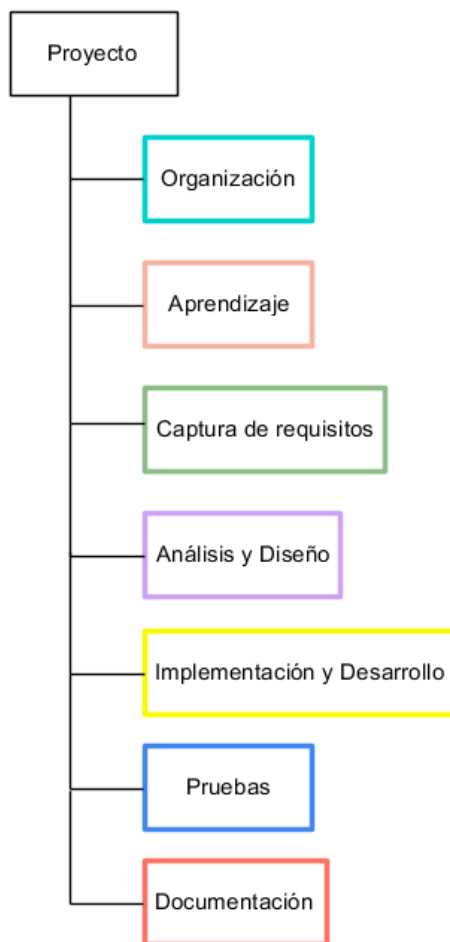


Figura 4: Grupos principales de las tareas

La descripción de las tareas se encuentra en las siguientes hojas.

### 5.4.1. Organización

El paquete de organización contendrá aquellas tareas que sirvan para hacerse una mejor de idea de la aplicación, así como todas las que tengan relación con la gestión del proyecto.

<b><i>Planteamiento de la aplicación</i></b>
<b>Paquete de trabajo:</b> Organización
<b>Duración estimada:</b> 1,5h
<b>Descripción:</b> Esta tarea consiste en obtener una idea general de lo que se desea hacer. Se responderán a preguntas cómo que problema se quiere solucionar, para qué se quiere hacer y para quién entre otras.
<b>Salidas/entregables:</b> Documento propuesta estudiante TFG relleno.
<b>Recursos necesarios:</b> Ordenador, documento de la propuesta del estudiante para el TFG, papel, impresora y bolígrafo.

Tabla 1: Tarea Planteamiento de la aplicación

<b><i>Nombrar la aplicación</i></b>
<b>Paquete de trabajo:</b> Organización
<b>Duración estimada:</b> 1h
<b>Descripción:</b> Se pretende obtener el nombre de la aplicación a desarrollar.
<b>Salidas/entregables:</b> N/A
<b>Recursos necesarios:</b> Papel y lápiz.

Tabla 2: Tarea Nombrar la aplicación

<b><i>Planificación de Tareas</i></b>
<b>Paquete de trabajo:</b> Organización
<b>Duración estimada:</b> 8h
<b>Descripción:</b> Identificar las tareas que componen el proyecto y redactar una descripción de cada una de ellas.
<b>Salidas/entregables:</b> EDT y hoja con las descripciones de las tareas.
<b>Recursos necesarios:</b> Visual Paradigm, papel y lápiz.

Tabla 3: Tarea Planificación de Tareas

<b><i>Planificación Temporal</i></b>
<b>Paquete de trabajo:</b> Organización
<b>Duración estimada:</b> 5h
<b>Descripción:</b> Estimar el tiempo a emplear en cada una de las tareas que componen el proyecto y planificar los tiempos de inicio y de fin.
<b>Salidas/entregables:</b> Diagrama Gantt y hoja con las estimaciones de las tareas.
<b>Recursos necesarios:</b> Ordenador, Gantt Project, papel y lápiz.

Tabla 4: Tarea Planificación temporal

<b><i>Elegir lenguaje de programación</i></b>
<b>Paquete de trabajo:</b> Organización
<b>Duración estimada:</b> 1h
<b>Descripción:</b> Se elegirá el lenguaje o/y framework a utilizar.
<b>Salidas/entregables:</b> N/A
<b>Recursos necesarios:</b> Artículos de internet y ordenador

Tabla 5: Tarea Elegir lenguaje de programación



<b><i>Elegir Software</i></b>
<b>Paquete de trabajo:</b> Organización
<b>Duración estimada:</b> 2,5h
<b>Descripción:</b> Se elegirán las herramientas software a utilizar durante el proyecto.
<b>Salidas/entregables:</b> Hoja con las herramientas a usar.
<b>Recursos necesarios:</b> Papel, lápiz y ordenador.

Tabla 6: Tarea Elegir software

<b><i>Preparación entorno de desarrollo</i></b>
<b>Paquete de trabajo:</b> Organización
<b>Duración estimada:</b> 1,5h
<b>Descripción:</b> Se instalarán las herramientas software seleccionadas.
<b>Salidas/entregables:</b> N/A
<b>Recursos necesarios:</b> Ordenador.

Tabla 7: Tarea Preparación entorno de desarrollo

<b><i>Reuniones con el director del proyecto</i></b>
<b>Paquete de trabajo:</b> Organización
<b>Duración estimada:</b> 30h
<b>Descripción:</b> Se revisará el trabajo realizado hasta el momento y se preguntará lo que se considere oportuno para mejorar el trabajo.
<b>Salidas/entregables:</b> Hoja con los aspectos a cambiar o mejorar apuntados.
<b>Recursos necesarios:</b> Lápiz y papel.

Tabla 8: Tarea Reuniones con el director del proyecto

### 5.4.2. Aprendizaje

El paquete de aprendizaje contendrá aquellas tareas que contribuyan a adquirir conocimientos sobre el lenguaje y las herramientas elegidas.

<b><i>Aprendizaje de modularidad en Python</i></b>
<b>Paquete de trabajo:</b> Aprendizaje
<b>Duración estimada:</b> 1h
<b>Descripción:</b> Estudio y repaso del lenguaje de programación <i>Python</i> . Se centrará en estudiar la orientación de objetos.
<b>Salidas/entregables:</b> N/A
<b>Recursos necesarios:</b> Ordenador y tutoriales de internet.

Tabla 9: Tarea Aprendizaje de modularidad en Python

<b><i>Aprendizaje de Django</i></b>
<b>Paquete de trabajo:</b> Aprendizaje
<b>Duración estimada:</b> 56h
<b>Descripción:</b> Estudio del framework de python llamado Django, además de las librerías a integrar y utilizar.
<b>Salidas/entregables:</b> Hoja con apuntes de lo aprendido.
<b>Recursos necesarios:</b> Lápiz, papel, libro “Django By Example” escrito por Antonio Melé y documentación oficial de Django.

Tabla 10: Tarea Aprendizaje de Django

<b><i>Aprendizaje de CSS Flexbox</i></b>
<b>Paquete de trabajo:</b> Aprendizaje
<b>Duración estimada:</b> 10h
<b>Descripción:</b> Aprendizaje de la herramienta CSS Flexbox para lograr un diseño <i>responsive</i> mediante la manipulación de contenedores.
<b>Salidas/entregables:</b> Hoja con apuntes de lo aprendido.
<b>Recursos necesarios:</b> Lápiz, papel y libro “Flexbox in CSS” escrito por Estelle Weyl.

Tabla 11: Tarea Aprendizaje de CSS Flexbox

<b><i>Aprendizaje de CSS Grid</i></b>
<b>Paquete de trabajo:</b> Aprendizaje
<b>Duración estimada:</b> 8h
<b>Descripción:</b> Aprendizaje de la herramienta CSS Grid para lograr un diseño <i>responsive</i> mediante la manipulación de contenedores, y poder ordenar estos en filas y columnas.
<b>Salidas/entregables:</b> Hoja con apuntes de lo aprendido.
<b>Recursos necesarios:</b> Lápiz, papel y libro “Grid Layout in CSS” escrito por Eric A. Meyer.

Tabla 12: Tarea Aprendizaje de CSS Grid

<b><i>Aprendizaje de UI</i></b>
<b>Paquete de trabajo:</b> Aprendizaje
<b>Duración estimada:</b> 9h
<b>Descripción:</b> Estudiar las técnicas básicas de diseño de interfaces para satisfacer las necesidades del usuario en cuanto a la interacción con el diseño elaborado. El objetivo será lograr que los usuarios se sientan cómodos para que permanezcan el mayor tiempo posible en el sitio web.
<b>Salidas/entregables:</b> Hoja con apuntes de lo aprendido.
<b>Recursos necesarios:</b> Lápiz, papel, material.io y Learn UI Design Blog.

Tabla 13: Tarea Aprendizaje de UI

<b><i>Aprendizaje de UX</i></b>
<b>Paquete de trabajo:</b> Aprendizaje
<b>Duración estimada:</b> 12h
<b>Descripción:</b> Estudiar las reglas básicas de la experiencia del usuario para conseguir diseñar una aplicación intuitiva a nivel de uso.
<b>Salidas/entregables:</b> Hoja con apuntes de lo aprendido.
<b>Recursos necesarios:</b> Lápiz, papel y libro “Don’t Make me Think” escrito por Steve Krug.

Tabla 14: Tarea Aprendizaje de UX

<b><i>Aprendizaje de FicWad.com</i></b>
<b>Paquete de trabajo:</b> Aprendizaje
<b>Duración estimada:</b> 1h
<b>Descripción:</b> Observar la estructura y el funcionamiento del sitio web.
<b>Salidas/entregables:</b> Hoja con apuntes de lo aprendido.
<b>Recursos necesarios:</b> Lápiz y papel.

Tabla 15: Tarea Aprendizaje de Ficwad.com

<b><i>Aprendizaje de AvengersFanfiction.com</i></b>
<b>Paquete de trabajo:</b> Aprendizaje
<b>Duración estimada:</b> 1h
<b>Descripción:</b> Observar la estructura y el funcionamiento del sitio web.
<b>Salidas/entregables:</b> Hoja con apuntes de lo aprendido.
<b>Recursos necesarios:</b> Lápiz y papel.

Tabla 16: Tarea Aprendizaje de AvengersFanfiction.com

<b><i>Aprendizaje de ArchiveOfOurOwn.org</i></b>
<b>Paquete de trabajo:</b> Aprendizaje
<b>Duración estimada:</b> 1h
<b>Descripción:</b> Observar la estructura y el funcionamiento del sitio web.
<b>Salidas/entregables:</b> Hoja con apuntes de lo aprendido.
<b>Recursos necesarios:</b> Lápiz y papel.

Tabla 17: Tarea Aprendizaje de ArchiveOfOurOwn.org

### 5.4.3. Captura de requisitos

Paquete que contiene las tareas que ayudarán a realizar un reconocimiento de las funcionalidades que hacen falta.

<b><i>Listar funcionalidades</i></b>
<b>Paquete de trabajo:</b> Captura de requisitos
<b>Duración estimada:</b> 4h
<b>Descripción:</b> Pensar en todas las funcionalidades que incluirá la aplicación.
<b>Salidas/entregables:</b> Papel con lista de funcionalidades.
<b>Recursos necesarios:</b> Lápiz y papel.

Tabla 18: Tarea Listar funcionalidades

<b><i>Plantear esquema web</i></b>
<b>Paquete de trabajo:</b> Captura de requisitos
<b>Duración estimada:</b> 9h
<b>Descripción:</b> Pensar el mapa web de la aplicación para identificar cuantas y qué páginas son necesarias.
<b>Salidas/entregables:</b> Papel con el mapa web.
<b>Recursos necesarios:</b> Lápiz y papel.

Tabla 19: Tarea Plantear esquema web

<b><i>Diseño prototipos low-res</i></b>
<b>Paquete de trabajo:</b> Captura de requisitos
<b>Duración estimada:</b> 15h
<b>Descripción:</b> Diseñar los prototipos de la aplicación con un nivel bajo de detalles.
<b>Salidas/entregables:</b> Papel con los prototipos dibujados.
<b>Recursos necesarios:</b> Papel, lápiz y bolígrafos.

Tabla 20: Tarea Diseño prototipos low-res

<i>Diseño prototipos high-res</i>
<b>Paquete de trabajo:</b> Captura de requisitos
<b>Duración estimada:</b> 28h
<b>Descripción:</b> Diseñar los prototipos de la aplicación con un nivel alto de detalles teniendo en cuenta colores y tipografía.
<b>Salidas/entregables:</b> Ficheros de Adobe XD e imágenes.
<b>Recursos necesarios:</b> Ordenador y Adobe XD.

Tabla 21: Tarea Diseño prototipos high-res

<i>Jerarquía de actores</i>
<b>Paquete de trabajo:</b> Captura de requisitos
<b>Duración estimada:</b> 0,5h
<b>Descripción:</b> Identificar los actores que tendrán algún papel en la aplicación.
<b>Salidas/entregables:</b> Lista con los actores participantes.
<b>Recursos necesarios:</b> Papel y lápiz.

Tabla 22: Tarea Jerarquía de actores

<i>Casos de uso</i>
<b>Paquete de trabajo:</b> Captura de requisitos
<b>Duración estimada:</b> 5h
<b>Descripción:</b> Identificar y recoger en un diagrama los casos de uso correspondientes a los actores participantes en la aplicación.
<b>Salidas/entregables:</b> Diagrama de casos de uso.
<b>Recursos necesarios:</b> Visual Paradigm y ordenador.

Tabla 23: Tarea Casos de uso

<i>Modelo de dominio</i>
<b>Paquete de trabajo:</b> Captura de requisitos
<b>Duración estimada:</b> 10h
<b>Descripción:</b> Identificar las entidades necesarias y sus relaciones entre sí.
<b>Salidas/entregables:</b> Diagrama del modelo de dominio.
<b>Recursos necesarios:</b> Ordenador y Visual Paradigm

Tabla 24: Tarea Modelo de dominio

#### 5.4.4. Análisis y Diseño

Paquete que contiene las tareas dedicadas a visualizar el código de la aplicación de forma gráfica.

<i>Diagramas de clases</i>
<b>Paquete de trabajo:</b> Análisis y Diseño
<b>Duración estimada:</b> 15h
<b>Descripción:</b> Realización del diagrama de clases de las funcionalidades principales.
<b>Salidas/entregables:</b> Diagrama de clases.
<b>Recursos necesarios:</b> Visual Paradigm y ordenador.

Tabla 25: Tarea Diagramas de uso



<i>Diagramas de secuencia</i>
<b>Paquete de trabajo:</b> Análisis y Diseño
<b>Duración estimada:</b> 15h
<b>Descripción:</b> Se realizarán los diagramas de secuencia de las funcionalidades principales.
<b>Salidas/entregables:</b> Diagramas de secuencia.
<b>Recursos necesarios:</b> Ordenador y Visual Paradigm.

Tabla 26: Tarea Diagramas de secuencia

<i>Diagramas de la base de datos</i>
<b>Paquete de trabajo:</b> Análisis y Diseño
<b>Duración estimada:</b> 6h
<b>Descripción:</b> Diseñar la base de datos a utilizar y realizar el diagrama de ésta.
<b>Salidas/entregables:</b> Diagrama de la base de datos.
<b>Recursos necesarios:</b> Ordenador y Visual Paradigm.

Tabla 27: Tarea Diagramas de la base de datos

#### 5.4.5. Implementación y Desarrollo

Este paquete contiene cualquier tarea que consista en escribir código para la programación de la aplicación y sus funcionalidades.

<b><i>Registro e identificación de usuarios</i></b>
<b>Paquete de trabajo:</b> Implementación y Desarrollo
<b>Duración estimada:</b> 12h
<b>Descripción:</b> Implementación del código necesario para el registro e identificación de los usuarios.
<b>Salidas/entregables:</b> Ficheros .py, .css y .js que contienen el código necesario para el buen funcionamiento de esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 28: Tarea Registro e identificación de usuarios

<b><i>Página de aterrizaje</i></b>
<b>Paquete de trabajo:</b> Implementación y Desarrollo
<b>Duración estimada:</b> 20h
<b>Descripción:</b> Implementación del código necesario para la consecución de la página de aterrizaje, más conocida por el término <i>landing page</i> .
<b>Salidas/entregables:</b> Ficheros .py, .css y .js que contienen el código necesario para el buen funcionamiento de esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 29: Tarea Página de aterrizaje

<b><i>Añadir fanfic externo</i></b>
<b>Paquete de trabajo:</b> Implementación y Desarrollo
<b>Duración estimada:</b> 25h
<b>Descripción:</b> Implementación del código necesario para que los usuarios puedan añadir correctamente <i>fanfics</i> externos.
<b>Salidas/entregables:</b> Ficheros .py, .css y .js que contienen el código necesario para el buen funcionamiento de esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 30: Tarea Añadir fanfic externo

<b><i>Cambiar configuración de la cuenta</i></b>
<b>Paquete de trabajo:</b> Implementación y Desarrollo
<b>Duración estimada:</b> 9h
<b>Descripción:</b> Desarrollar la funcionalidad que permite al usuario poder cambiar diversos aspectos de su cuenta.
<b>Salidas/entregables:</b> Ficheros .py, .css y .js que contienen el código necesario para el buen funcionamiento de esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 31: Tarea Cambiar configuración de la cuenta

<b><i>Ver perfil de usuario</i></b>
<b>Paquete de trabajo:</b> Implementación y Desarrollo
<b>Duración estimada:</b> 3h
<b>Descripción:</b> Desarrollar la funcionalidad que permite al usuario poder ver su perfil de usuario, y el de otros miembros.
<b>Salidas/entregables:</b> Ficheros .py, .css y .js que contienen el código necesario para el buen funcionamiento de esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 32: Tarea Ver perfil de usuario

<b><i>Ver página de fanfic</i></b>
<b>Paquete de trabajo:</b> Implementación y Desarrollo
<b>Duración estimada:</b> 4h
<b>Descripción:</b> Desarrollar la funcionalidad que permite al usuario poder ver la página de los <i>fanfics</i> .
<b>Salidas/entregables:</b> Ficheros .py, .css y .js que contienen el código necesario para el buen funcionamiento de esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 33: Tarea Ver página de fanfic

<b><i>Comentar en fanfics</i></b>
<b>Paquete de trabajo:</b> Implementación y Desarrollo
<b>Duración estimada:</b> 9h
<b>Descripción:</b> Desarrollar la funcionalidad que permite al usuario poder comentar en los <i>fanfics</i> .
<b>Salidas/entregables:</b> Ficheros .py, .css y .js que contienen el código necesario para el buen funcionamiento de esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 34: Tarea Comentar en fanfics

<b><i>Añadir notas a capítulos de los fanfics</i></b>
<b>Paquete de trabajo:</b> Implementación y Desarrollo
<b>Duración estimada:</b> 9h
<b>Descripción:</b> Implementación del código necesario para que los usuarios tengan la posibilidad de guardar notas privadas en cada capítulo de los <i>fanfics</i> .
<b>Salidas/entregables:</b> Ficheros .py, .css y .js que contienen el código necesario para el buen funcionamiento de esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 35: Tarea Añadir notas a capítulos de los fanfics

<b><i>Explorar fanfics</i></b>
<b>Paquete de trabajo:</b> Implementación y Desarrollo
<b>Duración estimada:</b> 10h
<b>Descripción:</b> Desarrollar la funcionalidad que permite al usuario poder explorar los <i>fanfics</i> según la categoría que seleccionen (películas, series, etc.).
<b>Salidas/entregables:</b> Ficheros .py, .css y .js que contienen el código necesario para el buen funcionamiento de esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 36: Tarea Explorar fanfics

<b><i>Filtrar Fanfics</i></b>
<b>Paquete de trabajo:</b> Implementación y Desarrollo
<b>Duración estimada:</b> 15h
<b>Descripción:</b> Implementación del código necesario para que los usuarios puedan buscar en la base de datos los <i>fanfics</i> según diversos criterios.
<b>Salidas/entregables:</b> Ficheros .py, .css y .js que contienen el código necesario para el buen funcionamiento de esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 37: Tarea Filtrar Fanfics

<b><i>Clasificar Fanfics</i></b>
<b>Paquete de trabajo:</b> Implementación y Desarrollo
<b>Duración estimada:</b> 9h
<b>Descripción:</b> Implementación del código necesario para que los <i>fanfics</i> se puedan añadir y eliminar de las listas creadas por los usuarios.
<b>Salidas/entregables:</b> Ficheros .py, .css y .js que contienen el código necesario para el buen funcionamiento de esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 38: Tarea Clasificar Fanfics

<b><i>Ver mis fanfics</i></b>
<b>Paquete de trabajo:</b> Implementación y Desarrollo
<b>Duración estimada:</b> 10h
<b>Descripción:</b> Implementación del código necesario para que los usuarios puedan ver sus <i>fanfics</i> ordenados por listas.
<b>Salidas/entregables:</b> Ficheros .py, .css y .js que contienen el código necesario para el buen funcionamiento de esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 39: Tarea Ver mis fanfics

<b><i>Gestionar listas del usuario</i></b>
<b>Paquete de trabajo:</b> Implementación y Desarrollo
<b>Duración estimada:</b> 5h
<b>Descripción:</b> Implementación del código necesario para que los usuarios puedan añadir, modificar y eliminar sus listas de <i>fanfics</i> .
<b>Salidas/entregables:</b> Ficheros .py, .css y .js que contienen el código necesario para el buen funcionamiento de esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 40: Tarea Gestionar listas del usuario

<b><i>Dashboard</i></b>
<b>Paquete de trabajo:</b> Implementación y Desarrollo
<b>Duración estimada:</b> 18h
<b>Descripción:</b> Implementación del código necesario para realizar un resumen de los diferentes datos del usuario, junto con las historias que haya añadido.
<b>Salidas/entregables:</b> Ficheros .py, .css y .js que contienen el código necesario para el buen funcionamiento de esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 41: Tarea Dashboard

<b><i>Registrar capítulos leídos</i></b>
<b>Paquete de trabajo:</b> Implementación y Desarrollo
<b>Duración estimada:</b> 5h
<b>Descripción:</b> Implementación del código necesario para que los usuarios puedan marcar que capítulos son los que han leído, ya sea desde la página de un <i>fanfic</i> o desde el <i>dashboard</i> .
<b>Salidas/entregables:</b> Ficheros .py, .css y .js que contienen el código necesario para el buen funcionamiento de esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 42: Tarea Registrar capítulos leídos

<b><i>Buscar fanfics y usuarios</i></b>
<b>Paquete de trabajo:</b> Implementación y Desarrollo
<b>Duración estimada:</b> 6h
<b>Descripción:</b> Implementación del código necesario para que los usuarios puedan buscar tanto <i>fanfics</i> como usuarios en la base de datos.
<b>Salidas/entregables:</b> Ficheros .py, .css y .js que contienen el código necesario para el buen funcionamiento de esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 43: Tarea Buscar fanfics y usuarios

<b><i>Avisar de nuevos capítulos</i></b>
<b>Paquete de trabajo:</b> Implementación y Desarrollo
<b>Duración estimada:</b> 25h
<b>Descripción:</b> Implementación del código necesario para que los usuarios sean avisados siempre que alguno de los <i>fanfics</i> que tengan añadidos cuente con algún capítulo nuevo.
<b>Salidas/entregables:</b> Ficheros .py, .css y .js que contienen el código necesario para el buen funcionamiento de esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 44: Tarea Avisar de nuevos capítulos

<b><i>Crear copias de seguridad</i></b>
<b>Paquete de trabajo:</b> Implementación y Desarrollo
<b>Duración estimada:</b> 5h
<b>Descripción:</b> Implementación del código necesario para que el usuario pueda crear copias de seguridad de todos sus datos.
<b>Salidas/entregables:</b> Ficheros .py, .css y .js que contienen el código necesario para el buen funcionamiento de esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 45: Tarea Crear copias de seguridad

<b><i>Funciones administrativas</i></b>
<b>Paquete de trabajo:</b> Implementación y Desarrollo
<b>Duración estimada:</b> 9h
<b>Descripción:</b> Implementación del código necesario para que el administrador y el grupo de bibliotecarios cuente con funcionalidades administrativas, como bien pueden ser editar y borrar <i>fanfics</i> .
<b>Salidas/entregables:</b> Ficheros .py, .css y .js que contienen el código necesario para el buen funcionamiento de esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 46: Tarea Funciones administrativas

<b><i>Ciencia de datos</i></b>
<b>Paquete de trabajo:</b> Implementación y Desarrollo
<b>Duración estimada:</b> 25h
<b>Descripción:</b> Implementación del código necesario para que los usuarios puedan visualizar <i>fanfics</i> similares a los añadidos y tener la posibilidad de ver diferentes gráficos de la información recolectada.
<b>Salidas/entregables:</b> Ficheros .py, .css y .js que contienen el código necesario para el buen funcionamiento de esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 47: Tarea Ciencia de datos

#### 5.4.6. Pruebas

El paquete de pruebas recoge las tareas que se encargan de comprobar el buen funcionamiento de las funcionalidades desarrolladas previamente para el proyecto.



<b><i>Pruebas del registro e identificación de usuarios</i></b>
<b>Paquete de trabajo:</b> Pruebas
<b>Duración estimada:</b> 2h
<b>Descripción:</b> Comprobación del correcto funcionamiento del registro e identificación de los usuarios.
<b>Salidas/entregables:</b> Hoja con los posibles errores que pueden surgir al realizar esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 48: Tarea Pruebas del registro y log in de usuarios

<b><i>Pruebas de la página de aterrizaje</i></b>
<b>Paquete de trabajo:</b> Pruebas
<b>Duración estimada:</b> 1h
<b>Descripción:</b> Comprobación del correcto funcionamiento de la página de aterrizaje.
<b>Salidas/entregables:</b> Hoja con los posibles errores que pueden surgir al realizar esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 49: Tarea Pruebas de la página de aterrizaje

<b><i>Pruebas de añadir fanfic externo</i></b>
<b>Paquete de trabajo:</b> Pruebas
<b>Duración estimada:</b> 4h
<b>Descripción:</b> Comprobación del correcto funcionamiento al añadir un <i>fanfic</i> externo.
<b>Salidas/entregables:</b> Hoja con los posibles errores que pueden surgir al realizar esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 50: Tarea Pruebas de Añadir fanfic externo

<b><i>Pruebas de cambiar configuración de la cuenta</i></b>
<b>Paquete de trabajo:</b> Pruebas
<b>Duración estimada:</b> 4h
<b>Descripción:</b> Comprobación del correcto funcionamiento de cambiar la configuración de la cuenta de un usuario.
<b>Salidas/entregables:</b> Hoja con los posibles errores que pueden surgir al realizar esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 51: Tarea Cambiar configuración de la cuenta

<b><i>Pruebas de ver perfil de usuario</i></b>
<b>Paquete de trabajo:</b> Pruebas
<b>Duración estimada:</b> 2h
<b>Descripción:</b> Comprobación del correcto funcionamiento de ver el perfil de un usuario.
<b>Salidas/entregables:</b> Hoja con los posibles errores que pueden surgir al realizar esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 52: Tarea Ver perfil de usuario

<b><i>Pruebas de ver página de fanfic</i></b>
<b>Paquete de trabajo:</b> Pruebas
<b>Duración estimada:</b> 3h
<b>Descripción:</b> Comprobación del correcto funcionamiento de ver la página de un <i>fanfic</i> .
<b>Salidas/entregables:</b> Hoja con los posibles errores que pueden surgir al realizar esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 53: Tarea Ver página de fanfic

<b><i>Pruebas de comentar en fanfics</i></b>
<b>Paquete de trabajo:</b> Pruebas
<b>Duración estimada:</b> 1h
<b>Descripción:</b> Comprobación del correcto funcionamiento al comentar en <i>fanfics</i> .
<b>Salidas/entregables:</b> Hoja con los posibles errores que pueden surgir al realizar esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 54: Tarea Pruebas de Comentar en fanfics

<b><i>Pruebas de añadir notas a capítulos de los fanfics</i></b>
<b>Paquete de trabajo:</b> Pruebas
<b>Duración estimada:</b> 1h
<b>Descripción:</b> Comprobación del correcto funcionamiento al añadir notas a capítulos de los <i>fanfics</i> .
<b>Salidas/entregables:</b> Hoja con los posibles errores que pueden surgir al realizar esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 55: Tarea Pruebas de añadir notas a capítulos de los fanfics

<b><i>Pruebas de explorar fanfics</i></b>
<b>Paquete de trabajo:</b> Pruebas
<b>Duración estimada:</b> 1h
<b>Descripción:</b> Comprobación del correcto funcionamiento al explorar los <i>fanfics</i> del sistema.
<b>Salidas/entregables:</b> Hoja con los posibles errores que pueden surgir al realizar esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 56: Tarea Explorar fanfics

<b><i>Pruebas de filtrar fanfics</i></b>
<b>Paquete de trabajo:</b> Pruebas
<b>Duración estimada:</b> 3h
<b>Descripción:</b> Comprobación del correcto funcionamiento de buscar <i>fanfics</i> según distintos parámetros.
<b>Salidas/entregables:</b> Hoja con los posibles errores que pueden surgir al realizar esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 57: Tarea Pruebas de Filtrar fanfics

<b><i>Pruebas de clasificar fanfics</i></b>
<b>Paquete de trabajo:</b> Pruebas
<b>Duración estimada:</b> 2h
<b>Descripción:</b> Comprobación del correcto funcionamiento de clasificar <i>fanfics</i> en listas de usuarios.
<b>Salidas/entregables:</b> Hoja con los posibles errores que pueden surgir al realizar esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 58: Tarea Pruebas de clasificar fanfics

<b><i>Pruebas de ver mis fanfics</i></b>
<b>Paquete de trabajo:</b> Pruebas
<b>Duración estimada:</b> 3h
<b>Descripción:</b> Comprobación del correcto funcionamiento de ver los <i>fanfics</i> propios del usuario.
<b>Salidas/entregables:</b> Hoja con los posibles errores que pueden surgir al realizar esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 59: Tarea Pruebas de ver mis fanfics

<i><b>Pruebas de gestionar listas del usuario</b></i>
<b>Paquete de trabajo:</b> Pruebas
<b>Duración estimada:</b> 1h
<b>Descripción:</b> Comprobación del correcto funcionamiento de gestionar las listas del usuario.
<b>Salidas/entregables:</b> Hoja con los posibles errores que pueden surgir al realizar esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 60: Tarea Pruebas de gestionar listas del usuario

<i><b>Pruebas del dashboard</b></i>
<b>Paquete de trabajo:</b> Pruebas
<b>Duración estimada:</b> 3h
<b>Descripción:</b> Comprobación del correcto funcionamiento del dashboard.
<b>Salidas/entregables:</b> Hoja con los posibles errores que pueden surgir al realizar esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 61: Tarea Pruebas del dashboard

<i><b>Pruebas de registrar capítulos leídos</b></i>
<b>Paquete de trabajo:</b> Pruebas
<b>Duración estimada:</b> 1h
<b>Descripción:</b> Comprobación del correcto funcionamiento al marcar capítulos como leídos.
<b>Salidas/entregables:</b> Hoja con los posibles errores que pueden surgir al realizar esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 62: Tarea Pruebas de Registrar capítulos leídos

<b><i>Pruebas de buscar fanfics y usuarios</i></b>
<b>Paquete de trabajo:</b> Pruebas
<b>Duración estimada:</b> 2h
<b>Descripción:</b> Comprobación del correcto funcionamiento de buscar <i>fanfics</i> y usuarios.
<b>Salidas/entregables:</b> Hoja con los posibles errores que pueden surgir al realizar esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 63: Tarea Pruebas de Buscar fanfics y usuarios

<b><i>Pruebas de avisar de nuevos capítulos</i></b>
<b>Paquete de trabajo:</b> Pruebas
<b>Duración estimada:</b> 5h
<b>Descripción:</b> Comprobación del correcto funcionamiento del aviso del sistema a los usuarios siempre que haya un <i>fanfic</i> que se haya actualizado.
<b>Salidas/entregables:</b> Hoja con los posibles errores que pueden surgir al realizar esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 64: Tarea Pruebas de Avisar de nuevos capítulos

<b><i>Pruebas de crear copias de seguridad</i></b>
<b>Paquete de trabajo:</b> Pruebas
<b>Duración estimada:</b> 4h
<b>Descripción:</b> Comprobación del correcto funcionamiento de la creación de copias de seguridad.
<b>Salidas/entregables:</b> Hoja con los posibles errores que pueden surgir al realizar esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 65: Tarea Pruebas de Crear copia de Seguridad

<b><i>Pruebas de funciones administrativas</i></b>
<b>Paquete de trabajo:</b> Pruebas
<b>Duración estimada:</b> 2h
<b>Descripción:</b> Comprobación del correcto funcionamiento de las herramientas administrativas que posee el administrador.
<b>Salidas/entregables:</b> Hoja con los posibles errores que pueden surgir al realizar esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 66: Tarea Pruebas de Funciones administrativas

<b><i>Pruebas de ciencia de datos</i></b>
<b>Paquete de trabajo:</b> Pruebas
<b>Duración estimada:</b> 5h
<b>Descripción:</b> Comprobación del correcto funcionamiento de la generación de <i>fanfics</i> similares dado uno, así como la posibilidad de ver gráficos de la información recolectada.
<b>Salidas/entregables:</b> Hoja con los posibles errores que pueden surgir al realizar esta funcionalidad.
<b>Recursos necesarios:</b> Ordenador, JetBrains Pycharm y Django.

Tabla 67: Tarea Pruebas de ciencia de datos

#### 5.4.7. Documentación

Paquete que contiene aquellas tareas que conlleven la escritura de la información del proyecto.

<b><i>Redacción de la Memoria</i></b>
<b>Paquete de trabajo:</b> Documentación
<b>Duración estimada:</b> 55h
<b>Descripción:</b> Redacción de la Memoria del Proyecto.
<b>Salidas/entregables:</b> Memoria del proyecto.
<b>Recursos necesarios:</b> Ordenador y Overleaf.

Tabla 68: Tarea Redacción de la Memoria del Proyecto

<b><i>Redacción de Manuales</i></b>
<b>Paquete de trabajo:</b> Documentación
<b>Duración estimada:</b> 8h
<b>Descripción:</b> Redacción de manuales de uso y de instalación de la aplicación.
<b>Salidas/entregables:</b> Manuales
<b>Recursos necesarios:</b> Ordenador y Overleaf.

Tabla 69: Tarea Redacción de Manuales

<b><i>Preparación de la defensa</i></b>
<b>Paquete de trabajo:</b> Documentación
<b>Duración estimada:</b> 26h
<b>Descripción:</b> Elaboración y preparación de la presentación de la defensa.
<b>Salidas/entregables:</b> Presentación del proyecto.
<b>Recursos necesarios:</b> Ordenador, PowerPoint, proyecto y documentación.

Tabla 70: Tarea Preparación de la defensa



#### 5.4.8. Resumen de la planificación realizada

Tras detectar las tareas que componen el proyecto y aportar información sobre cada una de ellas, en la Tabla 71 se muestra el nombre, las dependencias y la duración de cada una de ellas.

En la última fila de la tabla se puede observar como el número de horas previstas para la completitud del proyecto asciende a 604 horas.

Tabla 71: Lista de Tareas resumidas

<b>Id</b>	<b>Nombre de la Tarea</b>	<b>Predecesoras</b>	<b>Duración estimada</b>
<b>Organización</b>			<b>50,5</b>
1	Planteamiento de la aplicación	-	1,5
2	Nombrar la aplicación	18	1
3	Planificación de Tareas	7	8
4	Planificación Temporal	3	5
5	Elegir lenguaje de programación	2	1
6	Elegir Software	5	2,5
7	Preparación entorno de desarrollo	6	1,5
8	Reuniones con el director del proyecto	1	30
<b>Aprendizaje</b>			<b>99</b>
9	Aprendizaje de modularidad en Python	4	1
10	Aprendizaje de Django	9	56
11	Aprendizaje de CSS Flexbox	10	10
12	Aprendizaje de CSS Grid	11	8
13	Aprendizaje de UI	12	9
14	Aprendizaje de UX	13	12
15	Aprendizaje de FicWad.com	14	1
16	Aprendizaje de AvengersFanfiction.com	15	1
17	Aprendizaje de ArchiveOfOurOwn.org	16	1
<b>Captura de requisitos</b>			<b>71,5</b>

Continuación de la Tabla 71

<b>Id</b>	<b>Nombre de la Tarea</b>	<b>Predecesoras</b>	<b>Duración estimada</b>
18	Listar Funcionalidades	1	4
19	Plantear esquema web	17	9
20	Diseño prototipos low-res	19	15
21	Diseño prototipos high-res	20	28
22	Jerarquía de actores	21	0,5
23	Casos de uso	22	5
24	Modelo de dominio	23	10
<b>Análisis y Diseño</b>			<b>36</b>
25	Diagramas de clases	24	15
26	Diagramas de secuencia	25	15
27	Diagramas de la base de datos	26	6
<b>Implementación y Desarrollo</b>			<b>208</b>
28	Registro e identificación de usuarios	27	12
29	Página de aterrizaje	28	20
30	Añadir fanfic externo	29	25
31	Cambiar configuración de la cuenta	30	9
32	Ver perfil del usuario	31	3
33	Ver página de fanfic	32	4
34	Comentar en fanfics	33	9
35	Añadir notas a capítulos de los fanfics	34	9
36	Explorar fanfics	35	10
37	Filtrar Fanfics	36	15
38	Clasificar Fanfics	37	9
39	Ver mis fanfics	38	10
40	Gestionar listas del usuario	39	5
41	Dashboard	40	18
42	Registrar capítulos leídos	41	5
43	Buscar fanfics y usuarios	42	6
44	Avisar de nuevos capítulos	43	25
45	Crear copias de seguridad	44	5
46	Funciones administrativas	45	9

Continuación de la Tabla 71

<b>Id</b>	<b>Nombre de la Tarea</b>	<b>Predecesoras</b>	<b>Duración estimada</b>
47	Ciencia de datos	46	25
<b>Pruebas</b>			<b>50</b>
48	Pruebas del registro e identificación de usuarios	47	2
49	Pruebas de la página de aterrizaje	48	1
50	Pruebas de añadir fanfic externo	49	4
51	Pruebas de cambiar configuración de la cuenta	50	4
52	Pruebas de ver perfil de usuario	51	2
53	Pruebas de ver página de fanfic	52	3
54	Pruebas de comentar en fanfics	53	1
55	Pruebas de añadir notas a capítulos de los fanfics	54	1
56	Pruebas de explorar fanfics	55	1
57	Pruebas de filtrar fanfics	56	3
58	Pruebas de clasificar fanfics	57	2
59	Pruebas de ver mis fanfics	58	3
60	Pruebas de gestionar listas del usuario	59	1
61	Pruebas del dashboard	60	3
62	Pruebas de registrar capítulos leídos	61	1
63	Pruebas de buscar fanfics y usuarios	62	2
64	Pruebas de avisar de nuevos capítulos	63	5
65	Pruebas de crear copias de seguridad	64	4
66	Pruebas de funciones administrativas	65	2
67	Pruebas de ciencia de datos	66	5
<b>Documentación</b>			<b>89</b>
68	Redacción de la Memoria	1	55
69	Redacción de Manuales	67	8
70	Preparación de la defensa	68, 69	26
<b>Horas Totales</b>			<b>604</b>

## 5.5. Planificación Temporal

Una vez que se tienen las tareas necesarias para la completitud del proyecto, junto con su estimación de tiempo, es necesaria una organización para poder ejecutar éstas de forma ordenada.

Se hará uso de los diagramas Gantt para estructurar las tareas mediante fechas de inicio y de fin. Una vez realizado el anterior diagrama, se tendrá acceso visual al flujo y orden de la ejecución de las tareas.

Antes de realizar el diagrama Gantt visto en la Figura 5 se han tenido en cuenta los bloques de tiempo libres de los que se dispone de forma semanal.

Este trabajo se realizará de forma concurrente a los estudios del grado y las prácticas escogidas. Dichas prácticas ocupan cuatro horas diarias de lunes a viernes. Se han distinguido cinco etapas a lo largo del curso:

- La *primera etapa* hace referencia al primer cuatrimestre (septiembre-diciembre). En esta etapa se dispondrán semanalmente alrededor de 20 horas libres.
- La *segunda etapa* constituirá el periodo de vacaciones de Navidad (diciembre-enero). Al no tener ni clase ni práctica, el periodo libre aumentará considerablemente a 30 horas libres por semana.
- La *tercera etapa* corresponde al segundo cuatrimestre (enero-mayo). El horario será similar al de la primera etapa, por lo que el número de horas libres semanales estimadas es el mismo.
- La *cuarta etapa* hace referencia a las vacaciones de Semana Santa (abril). Como ha ocurrido anteriormente, este periodo equivale al de la segunda etapa, estimándose así el mismo número de horas.
- La *quinta y última etapa* empezará a mediados de mayo y terminará a finales de junio. El tiempo libre disponible rondará alrededor de las 63 horas semanales.

En cualquier etapa se ha de tener en cuenta de que el número de horas no será el mismo todos los días de la semana.

Para estimar de forma rápida la cantidad de horas mínimas de las que se dispondrá se va a suponer que:

- Se tienen nueve meses (octubre - junio).
- Los meses tienen cuatro semanas.
- Se disponen de 20 horas semanales libres, es decir, el mínimo del que se dispondrá.

En la Fórmula 1 se ha calculado el tiempo mínimo del que se dispondrá, sin tener en cuenta que el número de horas libres varía durante el año escolar.

$$9 \text{ meses} * 4 \text{ semanas} * 20 \text{ horas semanales} = 720 \text{ horas} \quad (1)$$

Como se puede apreciar, el número de horas libres estimadas (720 horas) es mayor al número de horas de trabajo estimadas (604 horas). Esto implica que el proyecto podrá estar terminado a finales de junio.

Utilizando los datos de la Tabla 71, construimos un diagrama de Gantt (Figura 5), para visualizar el tiempo a dedicar a cada tarea.

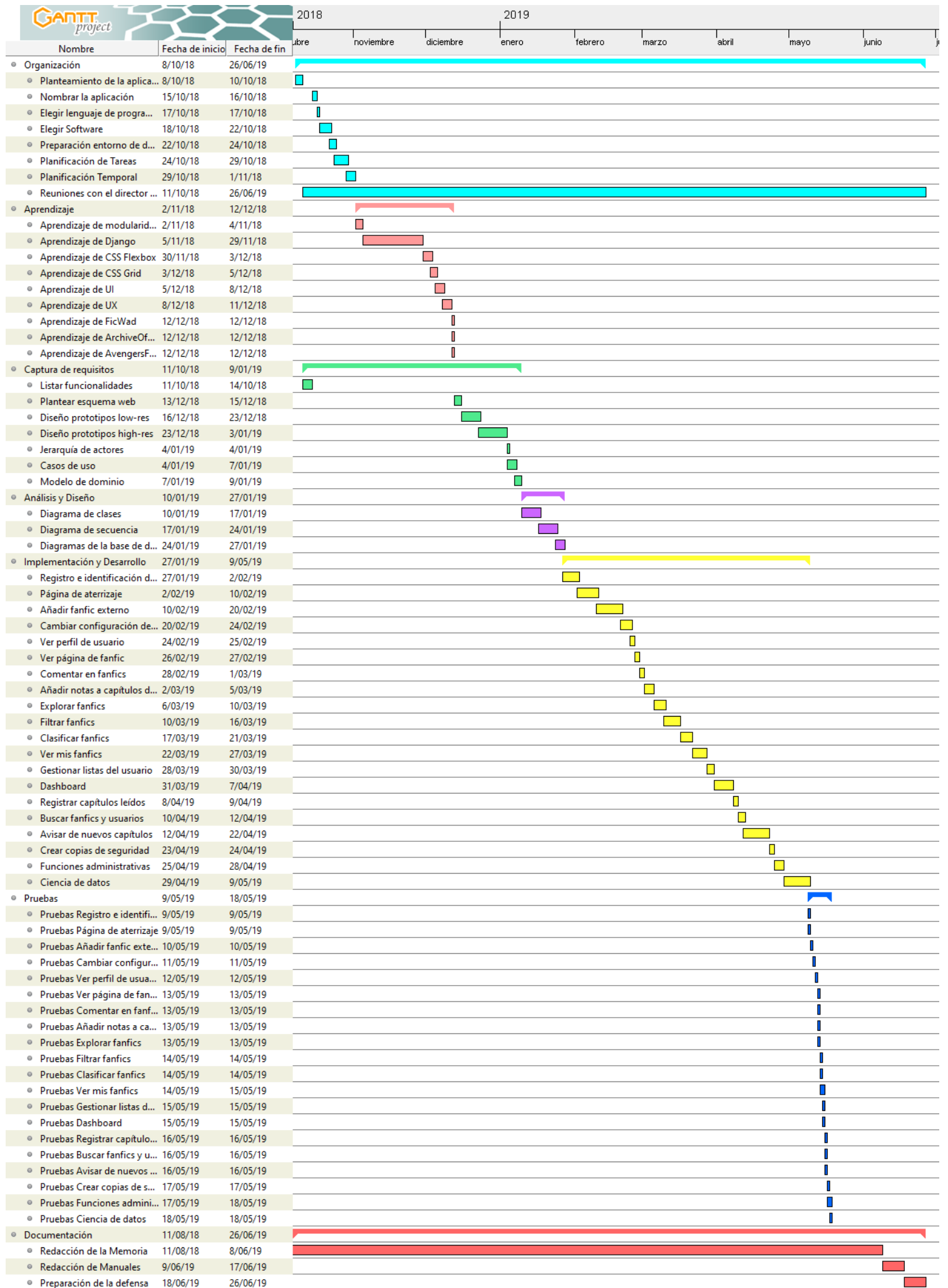


Figura 5: Diagrama Gantt de las tareas

## 5.6. Evaluación económica

En esta sección se calcularán los gastos que conlleva la realización de este proyecto, además de proponer una estrategia para conseguir que la aplicación web a desarrollar sea rentable.

### 5.6.1. Recursos humanos

Para calcular el coste de los recursos humanos, se ha tenido en cuenta el salario de un graduado superior en empresas de servicios de informática. Este salario se ha obtenido del BOE [4], y es de unos 25.189,02€ anuales. Por lo tanto, la cantidad mensual (30 días) será de 2.099,085€.

Se ha supuesto que de media se trabajarán unas 30 horas semanales, que corresponde a 4,28 horas por día, y a 128,4 horas mensuales.

Mediante el uso de la siguiente fórmula conseguimos obtener el sueldo por hora.

$$\text{Salario/hora} = \frac{2099,085}{128,4} = 16,34 \quad (2)$$

Se cobrarán 16,34€ por hora trabajada.

$$\text{Salario total ganado} = 16,34 * 604 = 9874 \quad (3)$$

El gasto económico en lo referente a la mano de obra sería de 9.874€.

## 5.6.2. Software

En este proyecto solo se utilizan dos programas de pago: *Visual Paradigm* y *JetBrains Pycharm*.

### 5.6.2.1. Visual Paradigm

La página oficial ofrece una suscripción mensual de 19€. Dicha suscripción solo será contratada durante los meses de trabajo, que son nueve. El coste se calcula de la siguiente manera:

$$\text{Coste total} = 19 * 9 = 171 \quad (4)$$

El coste total es de 171€.

### 5.6.2.2. JetBrains Pycharm

La página oficial de este producto también ofrece una suscripción mensual, en esta ocasión de 8,90€. Dicha suscripción, al igual que en el caso anterior, solo será contratada durante los meses de trabajo, que son nueve. El coste se calcula de la siguiente manera:

$$\text{Coste total} = 8,90 * 9 = 80,1 \quad (5)$$

El coste total es de 80,1€.



### 5.6.3. Hardware

#### 5.6.3.1. Ordenador portátil HP bs-129ns

La vida útil que se estima para este portátil es de unos 5 años, y su coste fue de aproximadamente unos 750€. Se calcula la amortización mensual:

$$Amortización\ mensual = \frac{750}{\frac{5\ años}{12\ meses}} = 12,5 \quad (6)$$

El resultado es 12,5€ por mes, y como el ordenador va a ser utilizado durante todo el proyecto (9 meses):

$$Amortización\ total = Amort.\ mensual * N^o\ Meses = 12,5 * 9 = 112,5 \quad (7)$$

La amortización total del ordenador es de unos 112,5€.

#### 5.6.3.2. Monitor DELL S2319H

La vida útil que se estima para este monitor es de unos 6 años, y su coste fue de aproximadamente unos 190€. Se calcula la amortización mensual:

$$Amortización\ mensual = \frac{190}{\frac{6\ años}{12\ meses}} = 2,63 \quad (8)$$

El resultado es 2,63€ por mes, y como el monitor va a ser utilizado durante todo el proyecto (9 meses):

$$Amortización\ total = Amort.\ mensual * N^o\ Meses = 2,63 * 9 = 23,67 \quad (9)$$

La amortización total del monitor es de unos 23,67€.

#### 5.6.4. Libros

En el aprendizaje de las herramientas y lenguajes de programación, se han utilizado una serie de libros que se han tenido que comprar. Los libros con sus costes correspondientes se indican en los siguientes puntos:

- **“Django By Example”, por Antonio Melé:** Se ha comprado en formato *Kindle* y su coste ha sido de 27,94€.
- **“Flexbox in CSS”, por Estelle Weyl:** Se ha comprado en formato *Kindle* y su coste ha sido de 9,19€.
- **“Grid Layout in CSS”, por Eric A. Meyer:** Se ha comprado en formato *Kindle* y su coste ha sido de 7,47€.
- **“Don’t Make Me Think”, por Steve Krug:** Se ha comprado en formato físico y su coste ha sido de 26,93€.

El coste total de los libros ha sido de 71,53€.

#### 5.6.5. Otros

Los gastos comunes del proyecto (luz, papel, internet, etc.) constarán el 5 % del coste total del proyecto.

#### 5.6.6. Gastos totales

Los gastos totales son 10.849,44€ y el desglose se puede apreciar en la Tabla 72.

Tipo	Cantidad
Recursos humanos	9.874€
Software	251,1€
Hardware	136,17€
Libros	71,53€
Total	10.332,8€
Otros	516,64€
<b>Total</b>	<b>10.849,44€</b>

Tabla 72: Gastos totales

### 5.6.7. Rentabilidad

Según la *Ley de Propiedad Intelectual*, “para que una persona pueda realizar un relato o una obra gráfica basada en un personaje o una historia cuyos derechos pertenecen a un tercero, ella deberá solicitar la preceptiva autorización al titular de derechos para que permita la transformación de su obra.” [5]

Esto implica que no se puede asegurar, si los autores de los *fanfics* han obtenido consentimiento alguno para escribir dichas historias. Por esta razón, no se considera que sea ético obtener una recompensa económica por permitir gestionar y/o difundir historias que posiblemente no tengan permisos legales.

En un caso hipotético en el que se comercializara para sufragar los costes generados, se propondría convertir el producto desarrollado en un producto *freemium*. Ofreciendo de esta manera funcionalidades básicas de forma gratuita, y otras como la notificación de nuevos capítulos, con un cierto coste. Se ofrecería una suscripción mensual de 2€ y una anual, de 22€.

Con 500 usuarios que hayan contratado el plan mensual durante un año:

$$Beneficios = 500 * 2 * 12 = 12000 \quad (10)$$

Si han contratado el plan anual:

$$\text{Beneficios} = 500 * 22 = 11000 \quad (11)$$

Con cualquiera de estos dos cálculos anteriores se obtendrían beneficios de unos 1.150,56€ y 150,56€, respectivamente.

Aunque estos números anteriores no se pueden asegurar, si que ofrecen una idea de lo rentable que podría ser la aplicación web si el paquete de pago fuera lo suficientemente atractivo para que un volumen considerable de usuarios lo comprara.

## 5.7. Gestión de Riesgos

En cualquier proyecto siempre es posible que surjan ciertos imprevistos. Estos imprevistos son los llamados *riesgos*, y en este apartado se identificarán algunos, junto con su plan de prevención y la probabilidad de que estos suceda.

Además también se formularán planes de contingencia, para que en caso de que no se pueda evitar que ocurran algunos de estos riesgos, se pueda retomar la continuidad del proyecto lo antes posible.

Para calcular el impacto se ha estipulado cuatro horas como la cantidad de tiempo diaria dedicada al trabajo.

La asignación de la *probabilidad* ha sido la siguiente:

- Muy baja (10 %): El evento puede ocurrir solo en circunstancias excepcionales.
- Baja (30 %): El evento puede ocurrir en algún momento.

- Media (60 %): El evento podría ocurrir en algún momento.
- Alta (80 %): El evento probablemente ocurrirá en varias circunstancias.
- Muy alta (100 %): Se espera que el evento ocurra en la mayoría de las circunstancias.

Y el *impacto* se ha clasificado en niveles según la cantidad de horas repetidas por el riesgo.

- Muy bajo: retraso  $< 1$  hora
- Bajo:  $1 \text{ hora} \leq \text{retraso} < 3 \text{ horas}$
- Normal:  $3 \text{ horas} \leq \text{retraso} < 5 \text{ horas}$
- Alto:  $5 \text{ horas} \leq \text{retraso} < 7 \text{ horas}$
- Muy Alto:  $7 \text{ horas} \leq \text{retraso}$

### 5.7.1. Enfermedad o lesión

Los imprevistos que puedan surgir relacionados con la salud de uno mismo. Se incluye la imposibilidad de continuar con el proyecto por tener la vista cansada, dolores de espalda y de muñecas.

#### Prevención

- **Derivadas del Trabajo:**
  - Mantener siempre una distancia adecuada de la pantalla. No se debe acercarse ni alejarse demasiado.
  - Sentarse con una buena postura, es decir, con la espalda recta. No hay que sentarse de forma encorvada.
  - Hacer los descansos recomendados de diez minutos cada hora para no forzar la vista.

- En todo momento hay que ser consciente del estado físico de nuestro cuerpo.
- Poner las manos en una postura adecuada mientras se escribe para evitar lesiones de muñeca.
- **No derivadas del Trabajo:**
  - Dormir las horas necesarias para que nuestro cuerpo y mente estén en buen estado.
  - Tener cuidado con las muñecas y las manos, al ser estas las partes físicas más importantes y necesarias para terminar este proyecto.

### **Plan de contingencia**

- Si los ojos están cansados se parará lo que se esté haciendo con el ordenador y se realizarán actividades que no requieran de una pantalla digital para descansar los ojos.
- Si se tiene dolor de espalda, se entiende que no se ha sido capaz de mantener una buena postura a lo largo del trabajo, entonces se arreglará esta inmediatamente.
- Si se percibe un dolor constante en las muñecas que impide continuar con el trabajo, se dará un margen de 30 minutos.
- Si se contrae alguna enfermedad, se aplicarán los métodos de cura especificados por el médico.

### **Probabilidad**

- Contraer una enfermedad. Muy baja: 10 %
- Dolor de espalda o vista cansada. Baja: 30 %
- Dolor de muñecas. Muy baja: 10 %

## Impacto

- Contraer una enfermedad.  $0,10 * 6 \text{ días} = 2,4 \text{ horas}$ . Impacto bajo.
- Dolor de espalda o vista cansada.  $0,30 * 1 \text{ día} = 1,2 \text{ horas}$ . Impacto bajo.
- Dolor de muñecas.  $0,10 * 1 \text{ día} = 0,4 \text{ horas}$ . Impacto muy bajo.

### 5.7.2. Error en la planificación temporal

Al no tener mucha experiencia estimando tiempos es probable que la planificación temporal no se ajuste a la realidad.

## Prevención

- **Derivadas del Trabajo:**
  - Cuando se estimen las horas de las tareas tener en cuenta un amplio margen de error siempre aumentando las horas necesarias.
  - Intentar llevar la ejecución de las tareas al día y no dejarlo para más tarde.
- **No derivadas del Trabajo:**
  - Llevar las asignaturas del *grado* cursado al día para tener un flujo constante de trabajo durante todo el proyecto, y no intervalos.

## Plan de contingencia

- Si el número de horas estimadas ha sido menor al real, se ajustará el diagrama de las tareas cuanto antes para poder continuar de forma ordenada.

- Si el número de horas estimadas ha sido mayor al real, decidir si merece la pena utilizar ese margen de tiempo para avanzar con la documentación de forma más dedicada.

### **Probabilidad**

- Subestimación de las horas a necesitar. Alta: 80 %
- Sobreestimación de las horas a necesitar. Alta: 80 %

### **Impacto**

- Falta de tiempo para completar las tareas.  $0,80 * 4 \text{ días} = 12,8 \text{ horas}$ . Impacto muy alto.
- Sobra tiempo para completar las tareas.  $0,80 * 0 \text{ día} = 0 \text{ horas}$ . No hay impacto porque es favorecedor.

#### **5.7.3. Falta de conocimientos**

Se utilizarán herramientas y lenguajes que nunca se han usado y esto puede generar situaciones en las que no se sepa realizar una acción necesaria para la realización del proyecto.

### **Prevención**

- Aprovechar la tareas de aprendizaje para conseguir el máximo conocimiento de lo que se quiere instruirse.
- Poner marcadores, ya sea en material digital o libros, de los apartados que posteriormente se podrían llegar a utilizar.
- Prestar atención a lo que se está aprendiendo y no *hacer por hacer* sin realmente escuchar.



### **Plan de contingencia**

- Utilizar los marcadores anteriormente señalados para un acceso rápido a la información previamente procesada.
- Usar *google* como medio de respuesta a preguntas que no se encuentren en el material de aprendizaje usado.

### **Probabilidad**

- Encontrar un problema que no se sepa resolver con el conocimiento actual y recursos de aprendizaje. Alta: 80 %

### **Impacto**

- Buscar una solución al problema.  $0,80 * 2 \text{ días} = 6,4 \text{ horas}$ . Impacto alto.

#### **5.7.4. Pérdida de información**

La información y algunos aspectos más del proyecto se crearán y guardarán en algunas herramientas online. Estos sitios web parecen bastante fiables respecto a la disponibilidad de los datos, pero nunca se puede estar demasiado seguro en lo que concierne a la pérdida de un trabajo cuyo impacto de pérdida sería extremadamente alto.

### **Prevención**

- Realizar una copia de seguridad cada semana de toda la información que se encuentre en cualquier herramienta online.

### **Plan de contingencia**

- Si la aplicación web ha dejado de estar disponible se buscará una alternativa en la que importar los datos de la copia de seguridad.

### **Probabilidad**

- Fallo de una herramienta online utilizada. Muy baja: 10 %

### **Impacto**

- Pérdida de datos y búsqueda de una nueva herramienta.  $0,10 * 1 \text{ días} = 0,4 \text{ horas}$ . Impacto muy bajo.

#### **5.7.5. Trabajo menos complejo de lo esperado**

Si la realización del proyecto es más simple de lo que se esperaba en un momento inicial, se añadirán nuevas funcionalidades para lograr una mayor complejidad.

### **Prevención**

- Realizar un diseño de la aplicación que permita, en la medida posible, añadir nuevas funcionalidades.
- Realizar una planificación de las tareas flexible donde sea moderadamente fácil incluir otras más.

### **Plan de contingencia**

- Formular nuevas funcionalidades en forma de tareas y añadir éstas a la planificación ya realizada.

### **Probabilidad**

- Complejidad baja del proyecto. Baja: 30 %

### **Impacto**

- Incorporación y realización de nuevas funcionalidades.  $0,30 * 30 \text{ días} = 36 \text{ horas}$ . Impacto muy alto.

#### **5.7.6. Problemas con el equipo informático**

El trabajo se elaborará y producirá en el ordenador personal que se posee, por lo que cualquier fallo de hardware o software puede ser determinante en la continuidad del proyecto.

### **Prevención**

- Instalar las actualizaciones sugeridas por el sistema operativo y no dejarlas para más tarde.
- Tener cuidado en el transporte del equipo, y procurar no golpearlo.
- Realizar una imagen del sistema cada dos semanas, para en caso de fallo, tener disponibles el sistema operativo, los programas y los ficheros.

### **Plan de contingencia**

- Si el problema ha sido de *software*, solo habrá que restaurar la última imagen del sistema para recuperar todos los datos. A continuación, se procederá a sincronizar los ficheros que se tengan en la *nube* con sus correspondientes copias en el ordenador.
- Por el contrario, si es un error de *hardware*, se procederá a utilizar otro ordenador restaurando nuestro entorno mediante la imagen del sistema previamente creada. Como en el caso anterior, se continuará sincronizando los ficheros almacenados en herramientas online, con sus correspondientes copias de seguridad en el ordenador nuevo.

### **Probabilidad**

- Error de software. Muy baja: 10 %
- Error de hardware. Muy baja: 10 %

### **Impacto**

- Carga de la imagen del sistema.  $0,10 * 2 \text{ días} = 0,8 \text{ horas}$ . Impacto muy bajo.

#### **5.7.7. Cambio en la estructura de las páginas web**

Los *fanfics* estarán alojados en páginas web, y se utilizarán técnicas de *scraping* para obtener los datos correspondientes. Cualquier cambio en su estructura puede ocasionar que ciertas funcionalidades, como la comprobación de la existencia de nuevos capítulos y la importación de *fanfics* externos no funcionen.

### **Prevención**

- No se puede prevenir.

### **Plan de contingencia**

- Si el problema se produce antes de la fecha de entrega del proyecto, se cambiará rápidamente la forma de obtener la información de los *fanfics*.
- Por el contrario, si se produce después, no habrá solución, y las funcionalidades nombradas anteriormente no funcionarán.

### **Probabilidad**

- Cambio de estructura. Muy baja: 10 %

Es muy improbable que este riesgo se produzca debido a que en más de cuatro años, las páginas web protagonistas no han sufrido cambio alguno.

### **Impacto**

- Modificar la forma de *scrapear*  $0,10 * 2 \text{ días} = 0,8 \text{ horas}$ . Impacto muy bajo.

## 6. Análisis de antecedentes

En la actualidad solo se ha encontrado una aplicación similar a la que se pretende desarrollar, *FicTrack* (Figura 6). Esta aplicación cuenta con varias funcionalidades base de nuestro proyecto, como bien son: registrar capítulos como leídos, gestionar listas y obtener recomendaciones. Y uno de sus problemas, es que la última actualización es del 2017, por lo que se piensa que la aplicación ha sido abandonada.

Las funcionalidades que se han añadido a estas anteriores nombradas, han sido las siguientes:

- Sistema de notificaciones para que el usuario esté al tanto en todo momento de lo que sucede en la aplicación.
- Filtrado de *fanfics* específico para explorar los *fanfics* del sistema, permitiendo elegir el idioma, personajes a incluir, etc.
- Los usuarios también podrán realizar un filtrado cuando estén viendo sus listas y *fanfics*.
- Los miembros pueden seguir a otros usuarios del sistema.
- Los usuarios pueden elegir si otros miembros pueden ver sus *fanfics* añadidos.
- Los usuarios podrán escribir reseñas en los *fanfics*.
- Se podrán añadir notas privadas a cada capítulo de un *fanfic*.
- Existencia de un grupo de bibliotecarios con acceso a un panel de control, para que puedan contribuir a la gestión de los *fanfics* que los usuarios añaden.
- Los usuarios tendrán disponible un enlace directo a cada capítulo de un *fanfic*.

- Permitir exportar los datos de un usuario para realizar copias de seguridad.
- Se podrán buscar tanto usuarios como *fanfics*.
- Generar gráficos respecto de los datos del usuario, para saber que *fandom* es el más popular en todo momento.
- Comprobación periódica de nuevos capítulos de los *fanfics*.

Por otra parte, también se planea trabajar el diseño y la usabilidad, ya que la página mostrada en la Figura 6 no aparenta ser una página seria de confianza. Una de las razones es la tipografía, ya que aparenta ser una de las que suelen usarse por defecto, contribuyendo a este efecto de diseño poco cuidado.

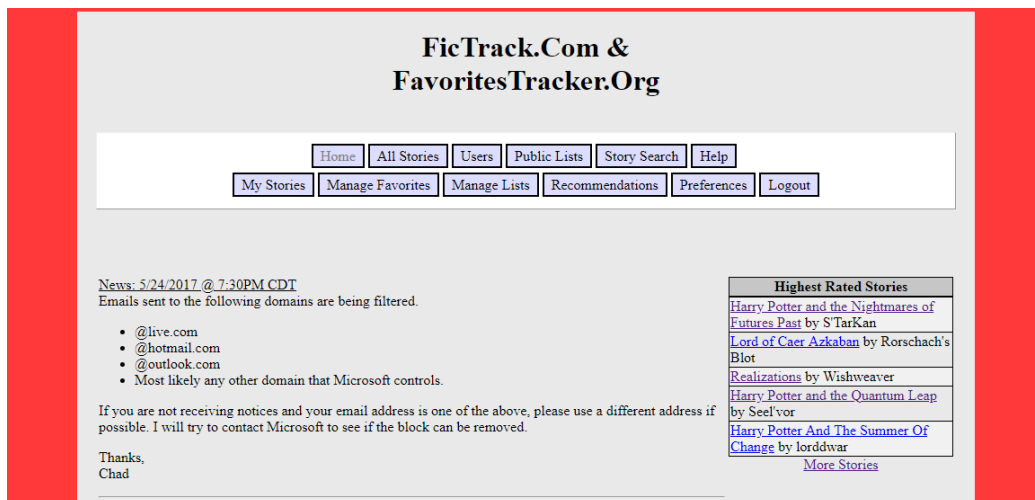
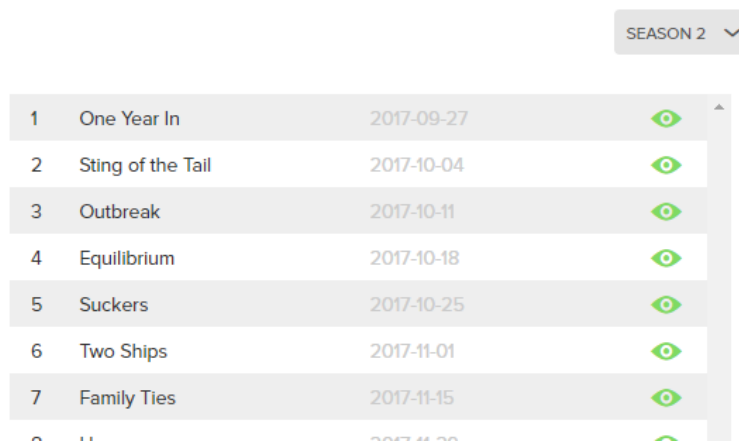


Figura 6: Página principal de *FicTrack.com*

Se han tenido en cuenta dos aplicaciones web para creación del proyecto: *TvTime* y *GoodReads*.

Se ha utilizado *TvTime* como inspiración para el diseño de la página individual de un *fanfic*. *TvTime* es una aplicación web que se encarga de gestionar las series que los usuarios ven. En la Figura 7 se puede apreciar que por cada temporada, se tienen una serie de episodios que el usuario puede ir marcando a medida que va viendo. Se planea realizar algo similar para los capítulos de un *fanfic*.



SEASON 2			
1	One Year In	2017-09-27	👁
2	Sting of the Tail	2017-10-04	👁
3	Outbreak	2017-10-11	👁
4	Equilibrium	2017-10-18	👁
5	Suckers	2017-10-25	👁
6	Two Ships	2017-11-01	👁
7	Family Ties	2017-11-15	👁

Figura 7: Página principal de *TvTime.com*

Por otra parte, *GoodReads* se encarga de llevar un registro de lo que se ha leído y se planea leer mediante el uso de estanterías. Este sistema estará presente mediante listas que el usuario podrá crear, modificar y eliminar. Cada *fanfic* podrá ser añadido (o eliminado) a una lista.



## 7. Captura de requisitos

La herramienta a desarrollar debe de cumplir ciertos requisitos respecto al usuario para poder llegar a satisfacer sus necesidades.

### 7.1. Jerarquía de actores

Se identifican los diferentes tipos de usuarios que interaccionarán con el sistema. Cada tipo de usuario tendrá acciones diferentes a poder realizar, aunque es probable que una misma actividad pueda ser ejecutada por diferentes actores.

Se parte de la idea de que un usuario debe haber iniciado sesión para poder hacer uso de las funcionalidades del sistema. Así mismo, existirá un grupo especial de usuarios que poseerán privilegios especiales. En un principio se pueden intuir tres tipos de usuarios: *usuario identificado*, *bibliotecario* y *administrador*.

En la Figura 8 se pueden apreciar los tipos de usuarios encontrados en este proyecto. Una descripción de los actores se puede encontrar en la sección 7.2.

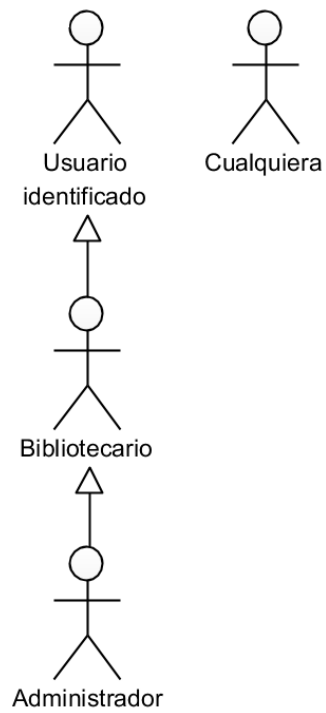


Figura 8: Jerarquía de actores

## 7.2. Casos de Uso

Una vez identificados los actores, se describen las diferentes tareas que éstos pueden realizar desde el punto de vista de la interacción con el sistema en la Figura 9.

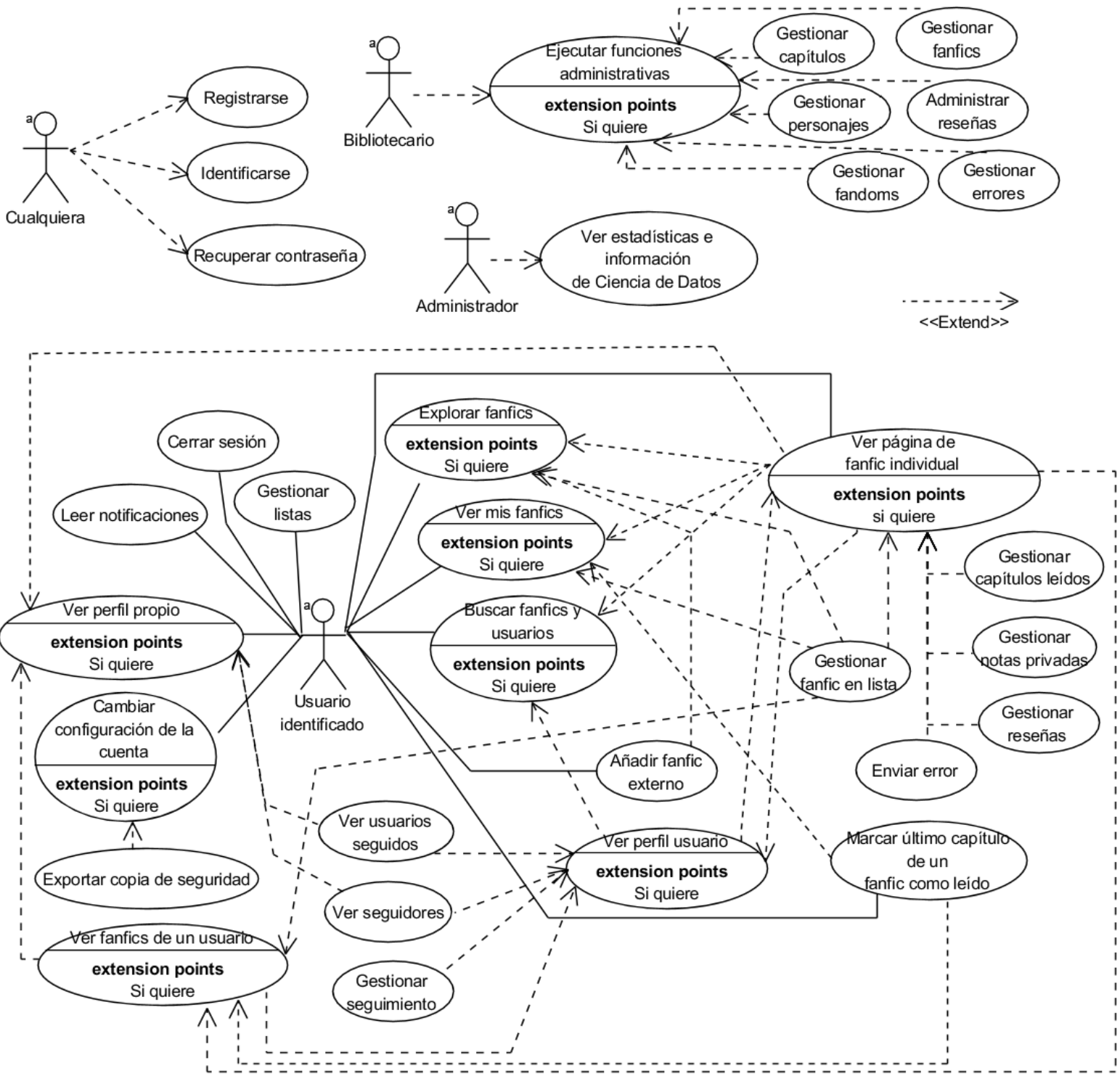


Figura 9: Diagrama casos de uso

### 7.2.1. Cualquiera

Término genérico para hacer referencia a personas anónimas que se encuentran visitando el sistema.

Este tipo de usuario solo podrá realizar las tres acciones que se muestran en la Figura 9.

- **Registrarse:** Acción que permitirá a un usuario anónimo inscribirse y quedar registrado en el sistema.
- **Identificarse:** Método por el cual un usuario anónimo adquiere una identidad propia y distinguible en la aplicación. Cuando un usuario anónimo se identifica, su lista de funcionalidades disponibles varía (ver Sección 7.2.2).
- **Recuperar contraseña:** Mediante esta opción, cualquier usuario ya registrado en el sistema que haya olvidado su contraseña podrá recuperarla.

### 7.2.2. Usuario identificado

Hace referencia a cualquier persona que se haya identificado en la aplicación. En otras palabras, si se ha iniciado sesión se pertenecerá a este grupo de actores.

Este tipo de usuario podrá ejecutar la mayoría de las funcionalidades del sistema.

- **Cerrar Sesión:** Se dejará de estar identificado en el sistema y se pasará a ser un usuario anónimo (Sección 7.2.1).

- **Gestionar listas:** Se podrán crear, modificar y eliminar las listas que el usuario utilizará para organizar sus *fanfics*.
- **Leer notificaciones:** Acción de leer los avisos recibidos por parte de la aplicación.
- **Ver perfil propio:** El usuario tendrá la posibilidad de ver su perfil propio.
- **Exportar copia de seguridad:** Posibilidad de crear una copia de seguridad de la información del usuario almacenada hasta el momento. Esta funcionalidad otorga una sensación de control al usuario, ya que puede respaldar sus datos cuando quiera.
- **Cambiar la configuración de la cuenta:** Los ajustes de la cuenta podrán ser cambiados por los usuarios.
- **Ver perfil usuario:** Además de ver el perfil propio, se tendrá acceso al perfil del resto de usuarios del sistema.
- **Explorar fanfics:** El sistema dispondrá de un catálogo por categorías que se podrá explorar, buscar, filtrar y ordenar para localizar *fanfics* según gustos personales.
- **Ver mis fanfics:** Cada usuario dispondrá de una “biblioteca” compuesta de listas en las que estarán clasificados los *fanfics*. La agrupación principal se realizará mediante *fandoms*.
- **Buscar fanfics y usuarios:** Acción de buscar términos en la base de datos del sistema. El resultado de la búsqueda será un listado de *fanfics* o usuarios afines al término usado.
- **Añadir fanfic externo:** Posibilidad de importar un *fanfic* a la aplicación. Cuando se haya introducido la dirección web de la historia interesada se procederá a importar el *fanfic* al sistema, siempre y cuando este no haya sido importado previamente. Se notificará al usuario cuando la historia haya sido añadida o se haya producido un error.

- **Ver página de fanfic individual:** Visualizar la página de un *fanfic*. Se dispondrá de los datos técnicos, la lista de capítulos y los comentarios de otros miembros registrados. Los usuarios podrán marcar los capítulos como leídos, así como añadir notas privadas a cada uno.
- **Marcar último capítulo de un fanfic como leído:** El usuario puede indicar que ha leído el último capítulo de un *fanfic*.
- **Ver fanfics de un usuario:** Ver los *fanfics* de un usuario siempre que se tenga el nivel de privacidad requerido. Este lo especifica el usuario desde su configuración.
- **Ver usuarios seguidos:** Ver los usuarios a los que un usuario sigue.
- **Ver seguidores:** Ver los usuarios que siguen a otro.
- **Gestionar seguimiento:** Seguir o dejar de seguir a un usuario.
- **Ver perfil usuario:** Ver el perfil de un usuario que no es el propio.
- **Gestionar capítulos leídos:** Marcar/Desmarcar como leídos capítulos de un *fanfic*.
- **Gestionar notas privadas:** Añadir, modificar y eliminar notas privadas de capítulos.
- **Gestionar reseñas:** Añadir, modificar y eliminar reseñas de los *fanfics*.
- **Gestionar fanfic en lista:** Añadir y/o eliminar un *fanfic* de una lista.
- **Enviar error:** Notificar de un error en la información del *fanfic* mostrada por pantalla.

### 7.2.3. Bibliotecario

A esta categoría pertenecerán aquellos usuarios identificados que posean privilegios de bibliotecario. Este tipo de usuarios podrán acceder al panel de

administración.

- **Ejecutar funciones administrativas:** Esta funcionalidad englobará aquellas acciones que no pueda ejecutar el usuario identificado, y que tengan que ver con la administración de la aplicación web. Por ejemplo: la eliminación de *fandoms* y/o *fanfics*.
- **Gestionar capítulos:** Aquellas acciones que permitan la gestión de los capítulos de los *fanfics*.
- **Gestionar personajes:** Aquellas acciones que permitan la gestión de los personajes pertenecientes a un *fandom*.
- **Gestionar fandoms:** Aquellas acciones que permitan la gestión de los *fandoms*, ya sea cambiar el nombre, eliminarlos, etc.
- **Gestionar fanfics:** Aquellas acciones que permitan la gestión de los *fanfics* del sistema.
- **Administrar reseñas:** Aquellas acciones que permitan la gestión de las opiniones de los usuarios respecto de los *fanfics*.
- **Gestionar errores:** Aquellas acciones que permitan la gestión de errores enviados por el usuario desde la página de un *fanfic*.

#### 7.2.4. Administrador

A esta categoría pertenecerán aquellos usuarios identificados que posean privilegios especiales, superiores a los de un bibliotecario.

- **Ver estadísticas e información de Ciencia de Datos:** La aplicación contará con información de las interacciones del usuario con el sistema, ya sea el número de veces que una historia ha sido añadida, o el número de *fanfics* añadidos a un mismo *fandom*. Esta información

en un principio no dice nada, pero tras aplicar técnicas de Ciencia de Datos se puede extraer conocimiento. Este conocimiento es el que se podrá visualizar en esta funcionalidad.

Los casos de uso explicados anteriormente se pueden encontrar detallados en la Sección 12.

### **7.3. Modelo de Dominio**

El diagrama surgido en esta sección permite visualizar de manera conceptual el sistema, haciendo uso de entidades, relaciones y atributos. Cada *caja* mostrada en la Figura 10 corresponde a una entidad, y las líneas serán las relaciones. El idioma utilizado ha sido el inglés ya que la aplicación en sí estará en inglés.



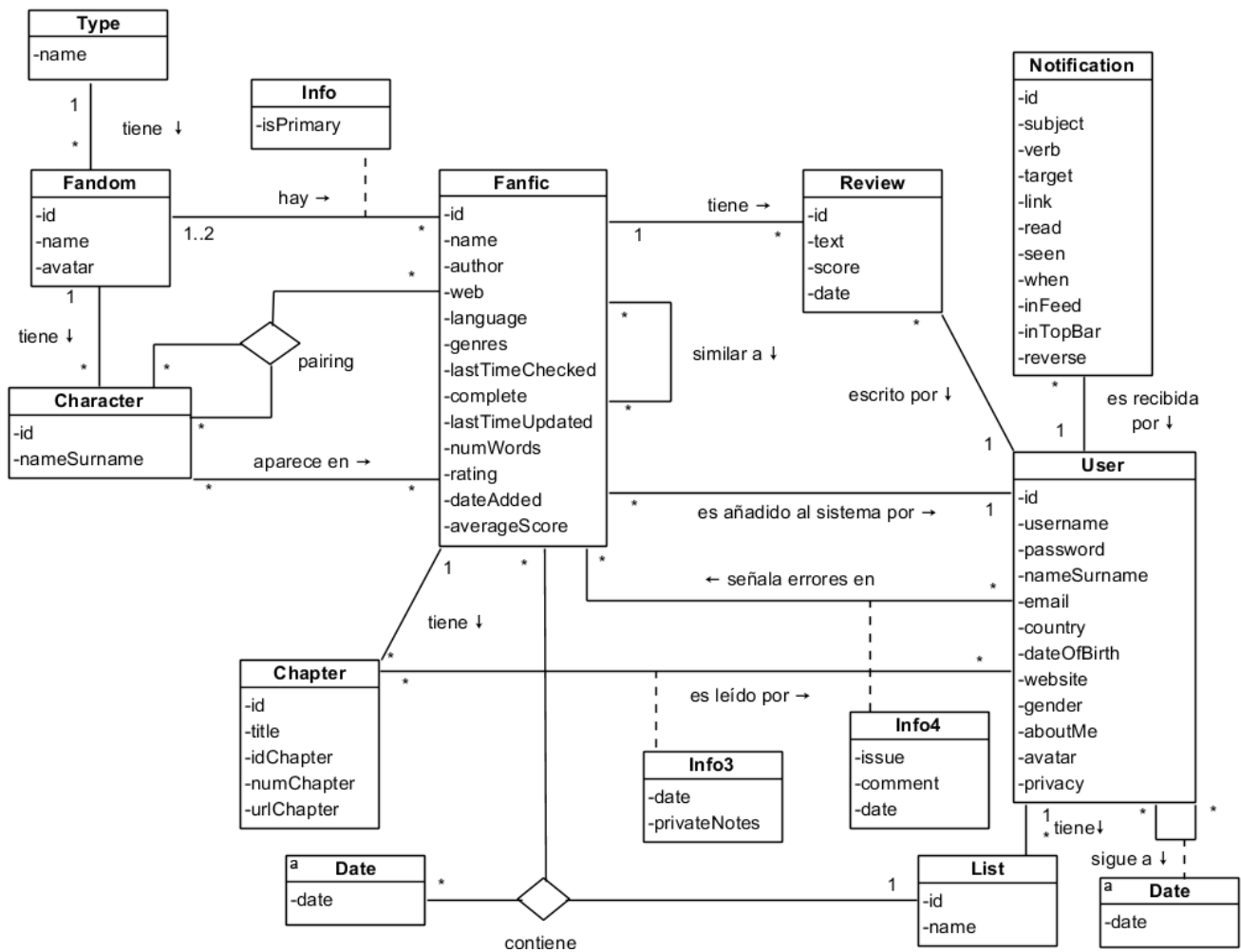


Figura 10: Modelo de dominio

A continuación, se describe cada una de las entidades que aparecen en la anterior figura (Figura 10):

- **Type:** Esta entidad representa el medio (*TV, Películas,...*) al cual pertenece la obra original en la que está basado el *fanfic*. Se guarda su nombre ya que no existen dos medios con el mismo nombre. Un *fandom* solo podrá estar asociado a un medio.

- **Fandom:** Entidad que representa la obra original en la que está basado el *fanfic*. *Harry Potter* y *El señor de los anillos* son ejemplos de *fandom*. Por cada entidad de este tipo se guardará un identificador y un nombre.
- **Character:** Entidad que representa a los personajes que aparecen en los *fandoms*. Cada *fandom* tendrá una serie de personajes asociados a éste. Además, la relación de un personaje con otro personaje se constituirá como la creación de un *pairing*. Por cada personaje se guardará un identificador y su nombre completo.
- **Fanfic:** Representa cada obra ficticia añadida en el sistema, es decir cada *fanfic*. Por cada entidad de este tipo se guardarán varios datos, de los que cabe destacar: el nombre del *fanfic*, el nombre del autor, la web donde se almacena, el estado (*completado, en progreso,..*) y la última vez que se comprobó si existían nuevos capítulos.

Un *fanfic* estará relacionado con aquellos que sean similares a él, y para esto se hará uso de la ciencia de datos.

Cada *fanfic* podrá pertenecer a uno o dos *fandoms*, ya que el sistema ofrecerá la posibilidad de almacenar *crossovers*. Dicho de otra manera, cada *fanfic* puede contener elementos y personajes de varios *fandoms*. También tendrá asociados *pairings*. Si un *fanfic* tiene dos *fandoms*, se debe indicar cual será el principal y cual el secundario.

- **Chapter:** La entidad capítulo hará referencia a cada capítulo de un *fanfic*. Se incluirá un identificador, el título, el número de capítulo y un enlace a la web donde poder leerlo.
- **Review:** Opinión escrita por un usuario asociada a un *fanfic*. Por cada opinión se guardará un identificador, un texto, una puntuación y la fecha en la que se publicó. Asimismo, por cada *fanfic* se tendrá una puntuación media de todas las reseñas.
- **User:** Entidad que representa a cada usuario registrado en la aplicación. Algunos de los datos a guardar serán: un identificador, nombre de usuario, contraseña, nombre completo, email, país y fecha de naci-

miento. Los usuarios podrán seguirse entre sí, además de poder señalar la existencia de errores en los *fanfics* del sistema.

Un usuario podrá marcar capítulos como leídos, además de poder añadir notas privadas a estos.

Por otro lado, tendrán la posibilidad de añadir *fanfics* que no se encuentren todavía en el sistema.

- **List:** Entidad que representa las listas con las que cuenta el usuario para organizar sus *fanfics* añadidos. Cuando un usuario añade una historia a una de sus listas también se añade la fecha para que quede registrado cuando se añadió. Esta relación será una ternaria por ser posible que un usuario añada una misma historia dos veces a una misma lista (*se añade, se borra y se añade*).
- **Notification:** Entidad que representa las notificaciones que el sistema enviará a los usuarios para notificarles de ciertas acciones que ocurran (*un usuario les sigue, etc.*).

## 8. Análisis y Diseño

Tras la captura de requisitos, se procede a definir la estructura que tendrá la aplicación, haciendo uso de diagramas.

### 8.1. Diagrama de la Base de Datos

A partir del modelo de dominio anteriormente creado (Sección 7.3), se obtiene el diagrama de la Figura 11 correspondiente a la base de datos del sistema. Cabe destacar que el idioma utilizado es el inglés, ya que se propone crear el sistema en este idioma.

Cada “caja” hará referencia a una tabla de la base de datos, y su explicación se encuentra en la Sección 7.3. Las claves primarias serán el identificador de un conjunto de datos de cada tabla. En cambio, las claves foráneas actuarán de enlace entre las tablas creando relaciones.

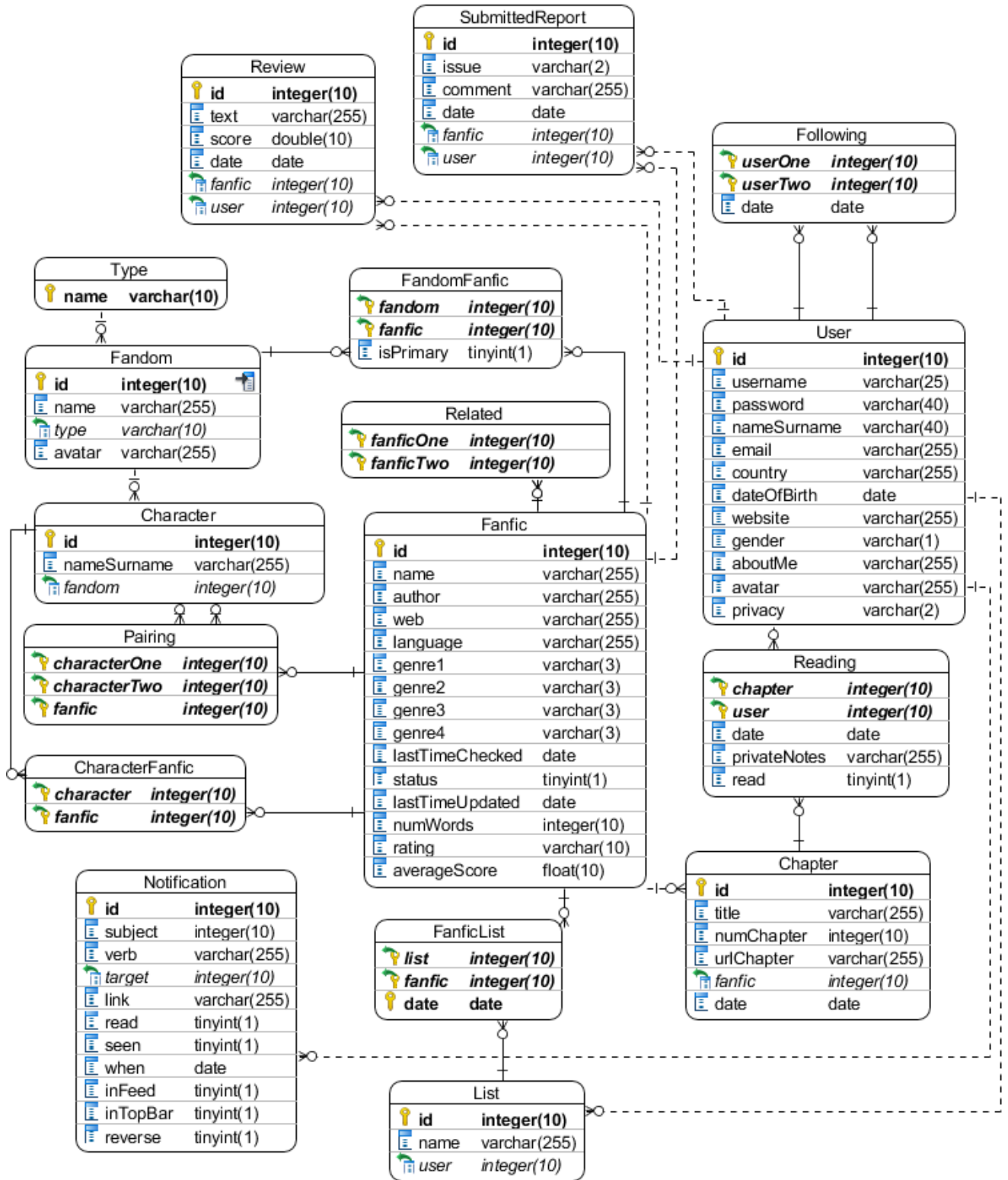


Figura 11: Diagrama relacional de la Base de Datos

## 8.2. Diagrama de clases

Como se ha comentado anteriormente, se está trabajando con el *framework* de *Django*. Este *framework* viene con unas clases por defecto que hay que manipular para conseguir el resultado esperado y construir la aplicación deseada. Aún así, gracias al patrón MTV (Modelo-Plantilla-Vista), las entidades mostradas en el modelo de dominio (Sección 7.3) y en el diagrama de la base de datos (Sección 8.1) constituirán la parte de *modelo*, siendo así cada una de ellas una clase.

En el diagrama mostrado en la Figura 12 se mostrarán aquellas *clases* o *modelos* a los que se hayan añadido métodos propios. Los atributos serán los mismos del diagrama de la base de datos (Sección 8.1) y/o modelo de dominio (Sección 7.3).

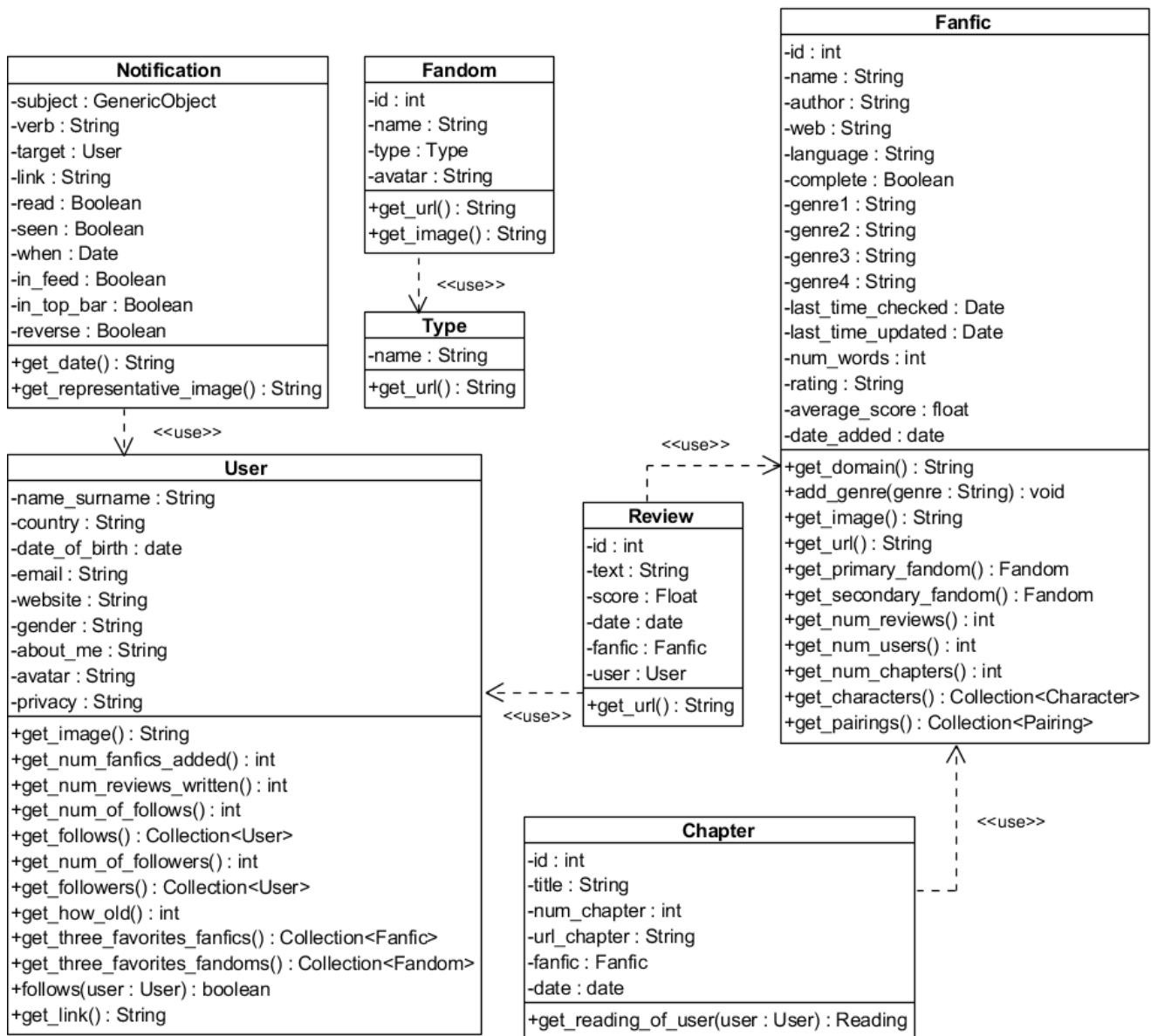


Figura 12: Diagrama de clases

Por otro lado, en el siguiente diagrama (Figura 13) se mostrarán aquellas clases que se han tenido que crear de forma manual para poder importar *fanfics* externos y poder comprobar la existencia de nuevos capítulos.

Se ha utilizado el patrón *Strategy* en el diseño de las clases.

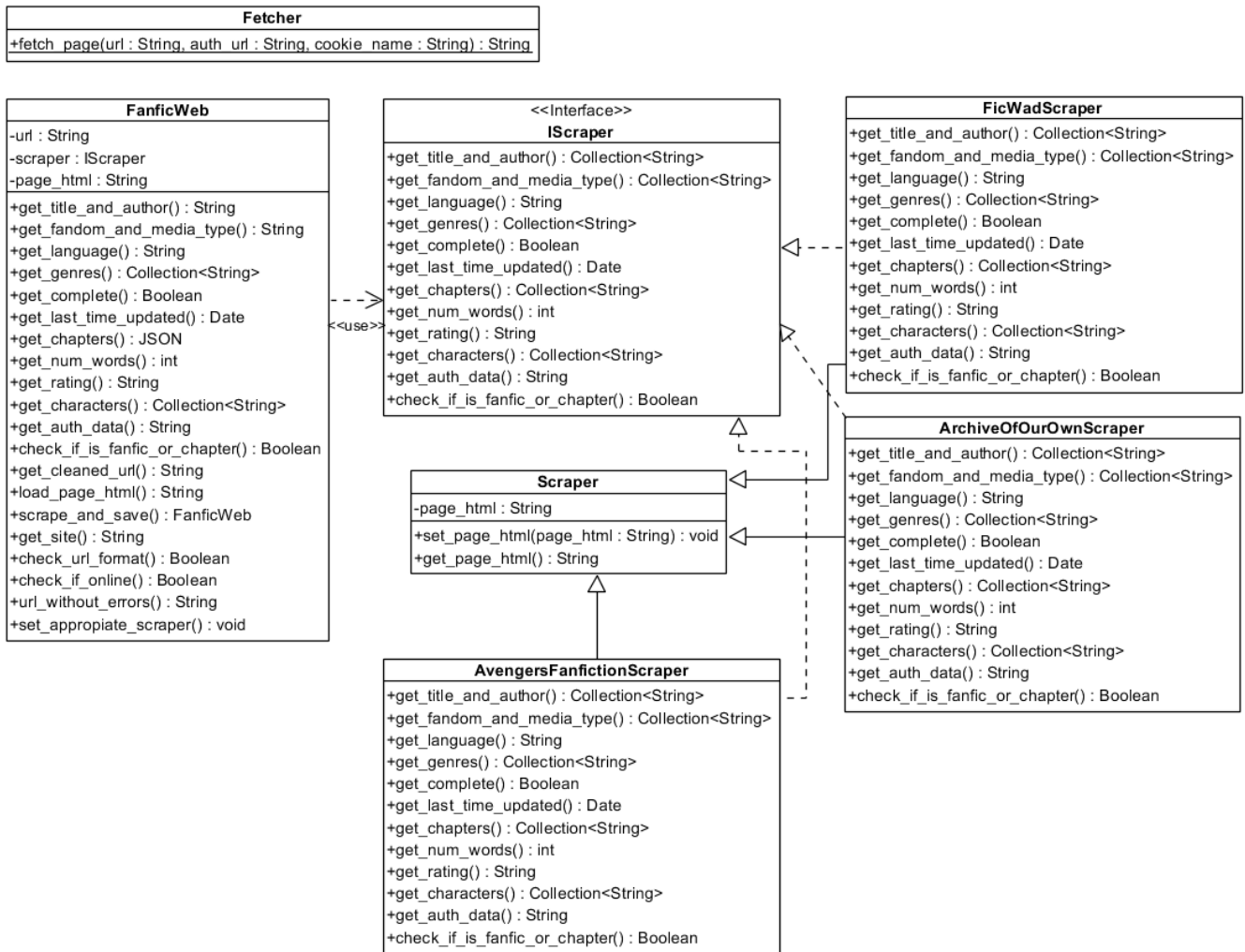


Figura 13: Diagrama de clases “añadir un *fanfic*”

Para realizar el sistema de recomendaciones de *fanfics* similares, también se han tenido que añadir varias clases de forma manual (Figura 14).



<b>Recommender</b>
-dataframe : Dataframe -indices : Series -cosine_sim
+get_data() : Dataframe +preprocess_data() : void +get_recommendations(fanfic_id : int) : Collection<int> +save_recommendations() : void +start()

<b>Dataframer</b>
<u>+get_fanfics_dataframe() : Dataframe</u>
<u>+get_fandoms_dataframe() : Dataframe</u>
<u>+get_characters_dataframe() : Dataframe</u>
<u>+query_to_dataframe(query_set) : Dataframe</u>
<u>+join_column(dataframe : Dataframe, column_names : Collection&lt;String&gt;, column_to_keep : String) : Dataframe</u>
<u>+remove_duplicated_rows_content(df : Dataframe, data_column_name : String, id_column : String) : Dataframe</u>
<u>+join_two_dataframes(one_df : Dataframe, one_key : String, two_df : Dataframe, two_key : String, union_type : String) : Dataframe</u>
<u>+create_map(dataframe : Dataframe, column_to_index : String) : Series</u>

Figura 14: Diagrama de clases “sistema de recomendación”

Por último, la clase mostrada en la Figura 15 representa de forma abstracta a la librería de *Celery*. Los métodos contenidos en dicha clase en realidad estarán repartidos en varios ficheros y clases, por lo tanto, por simplicidad se utilizará “CeleryTasksManager” para representar el conjunto de todas las clases de *Celery* a modificar.

<b>CeleryTasksManager</b>
+update_chapters_of_fanfics() : void
+scrape_and_add_fanfic(url : String, current_user : int)
+run_recommender_system() : void

Figura 15: Clase de las tareas

### 8.3. Diagramas de secuencia

Se han realizado varios diagramas de secuencia que se pueden encontrar en el Anexo (Sección 13).

El diagrama de la Figura 104 intentará mostrar visualmente el proceso de adición de un *fanfic* a la cola de tareas a procesar. Estas tareas se realizan de forma asíncrona.

Este proceso continuará en el diagrama de la Figura 107. Cuando la librería *Celery* considere oportuno empezar la ejecución de la funcionalidad indicada en el diagrama anterior, se inicializará el proceso de importación mediante la técnica *scraping* (Figura 108).

La comprobación de nuevos capítulos se puede ver en la Figura 109.

Por último, el flujo para la creación y actualización del sistema de recomendación de *fanfics* se describe en la Figura 110. Uno de los métodos de este último diagrama se encuentra en la Figura 111.

## 9. Desarrollo

En este apartado se explicará lo que se ha hecho a lo largo del trabajo para llegar a desarrollar este.

### 9.1. Desarrollo de prototipos

El primer paso fue crear múltiples prototipos como el de la Figura 16 para obtener una idea de la estructura gráfica de la aplicación. Estos prototipos son **prototipos de baja fidelidad**, y ofrecen una manera rápida de visualizar la aplicación web.

Tras aprender varios aspectos del diseño de experiencia de usuarios (UX), las interfaces se diseñaron teniendo en cuenta que:

- No leemos páginas. Las escaneamos.
- Diseñar para escanear, no leer.
- No tomamos decisiones óptimas, sino la primera que pueda servir.
- Elegir claridad antes que consistencia.

El resto de los prototipos de este tipo se pueden encontrar en el Anexo (Sección 14).

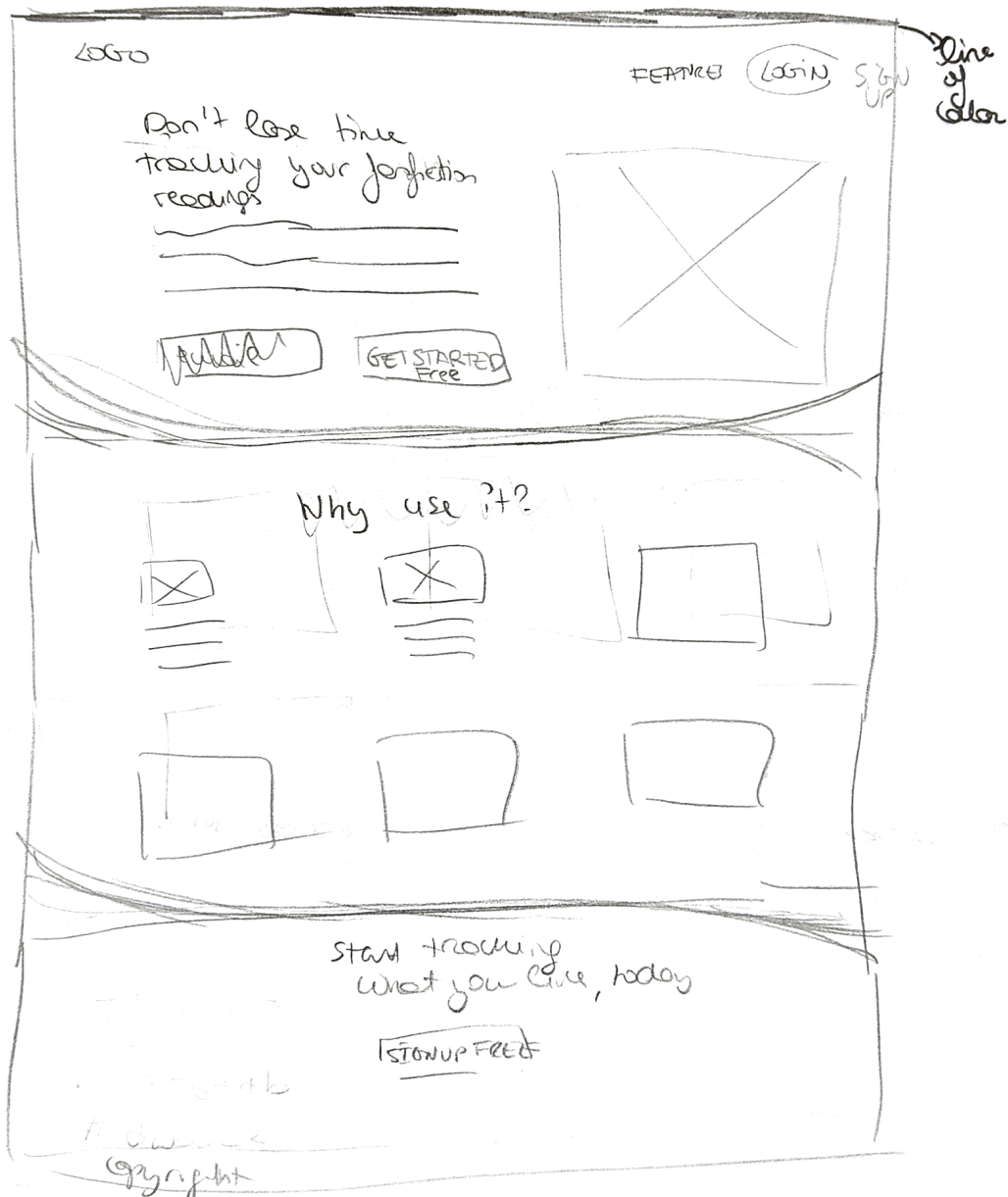


Figura 16: Prototipo baja fidelidad de la página de aterrizaje

Después, se desarrollaron los **prototipos de alta fidelidad**. Este tipo de prototipos pretenden ser muy similares al diseño de la versión final del producto. Se suelen realizar utilizando programas de prototipado, como *Ado-*

be XD. Un prototipo de este tipo se puede ver en la Figura 17. El resto se encuentran en el Anexo (Sección 15).

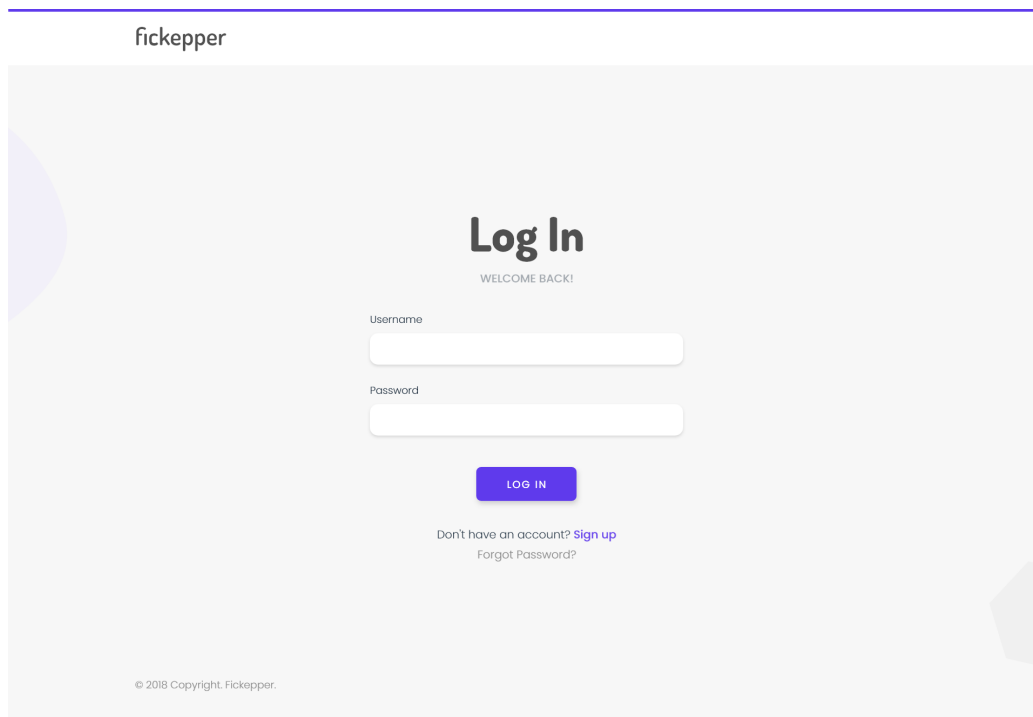


Figura 17: Prototipo alta fidelidad de la identificación

Con la creación de los prototipos anteriores se ha conseguido que las decisiones de diseño estén tomadas antes de empezar a implementar, evitando así una carga adicional de trabajo. Al implementar, no habrá que pensar donde colocar cada elemento, sino que solamente se tendrá que copiar el diseño de los prototipos ya creados.

Se han utilizado los espacios en blanco como forma de guiar el ojo y permitir respirar a los elementos. Por otro lado, la gama de colores que predomina en los prototipos se encuentra en la Figura 18, y solo se han utilizado dos fuentes a lo largo de las pantallas creadas (Figura 19).



Figura 18: Gama de colores

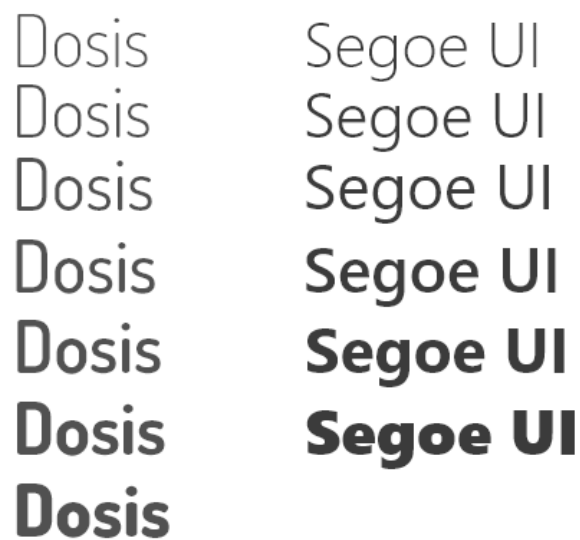


Figura 19: Fuentes utilizadas

En la Figura 17 se puede observar una línea de color en la parte superior del prototipo. Esta línea aportará color al diseño de una forma sencilla y elegante. Sitios web populares como *Stack Overflow* la utilizan.

Los prototipos se han codificado con la característica de *responsive* (Figura 20 y Figura 21) en mente. El *framework* de CSS utilizado, llamado *Bulma* (Figura 22), es útil para crear estructuras y añadir elementos HTML de for-

ma sencilla y rápida, además de facilitar el diseño *responsive*, al contar sus componentes con esta característica.

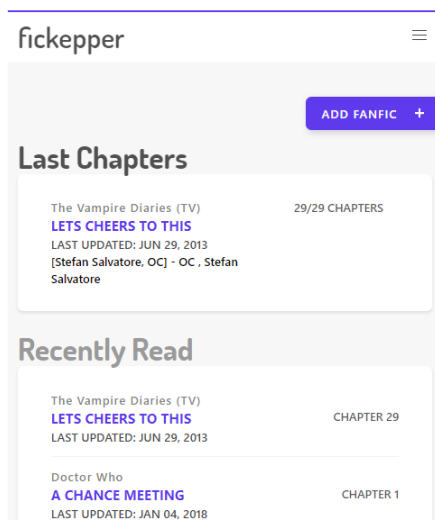


Figura 20: Diseño *responsive* (I)

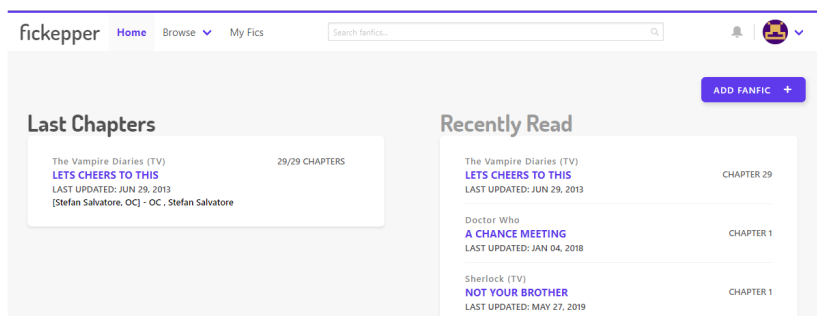


Figura 21: Diseño *responsive* (II)

También se ha combinado código CSS creado desde cero con el ya existente para lograr diseños parecidos a los mostrados en los prototipos de alta

fidelidad. Para esto se han utilizado los modos de diseño *Flexbox* y *CSS Grid* para estructurar ciertos elementos de la aplicación.

*Bulma* no cuenta con código *javascript*, por lo que se han tenido que escribir desde cero todas las acciones dinámicas implementadas escritas en este lenguaje.

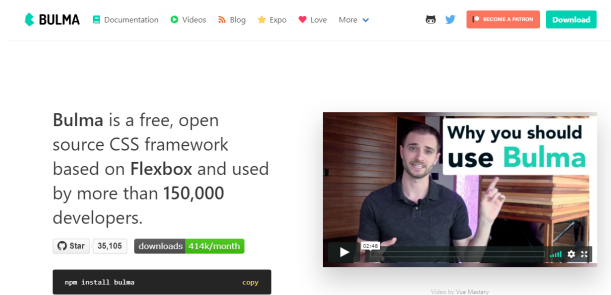


Figura 22: Página principal de *Bulma.io*

Por último, las ilustraciones mostradas en los prototipos se han obtenido de *Undraw.co* (Figura 23), un sitio web que proporciona ilustraciones vectoriales de forma gratuita.

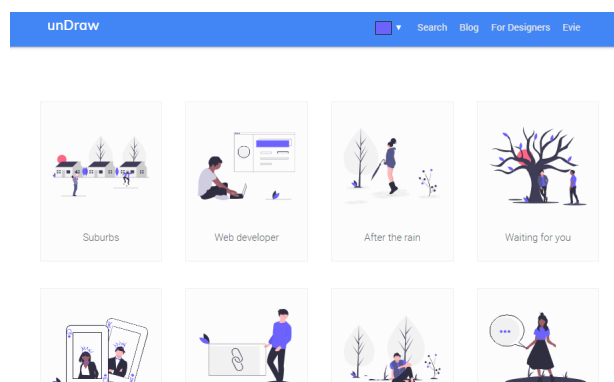


Figura 23: Página principal de *Undraw.co*



## 9.2. Framework Django

Como se ha comentado en varias ocasiones, se ha utilizado el *framework* de Django para desarrollar el proyecto. Este *framework* permite la creación de un proyecto, con sus correspondientes aplicaciones. Cada aplicación constituirá una aplicación web que se encargará de realizar alguna funcionalidad, como por ejemplo un sistema de notificaciones. Dichas aplicaciones se importan mediante el fichero de configuración del proyecto.

El proyecto de *Django* se ha llamado *fickeep*, y las aplicaciones creadas junto con su explicación (Figura 24):

- **analyzer:** Aplicación creada para gestionar las acciones relacionadas con la ciencia de datos.
- **chapters:** Gestiona los capítulos de los *fanfics*.
- **characters:** Se ocupa de los personajes de los *fandoms*.
- **common:** Contiene funciones comunes que no pertenecen a las aplicaciones existentes.
- **fandoms:** Se ocupa de las acciones relacionadas con los *fandoms*.
- **fanfics:** Contiene la lógica de los *fanfics*.
- **lists:** Gestiona las listas de los usuarios.
- **media\_types:** Gestiona los tipos a los que los *fandoms* pueden pertenecer.
- **notifier:** Es el sistema de notificaciones creado.
- **pages:** Gestiona aquellas páginas a las que el usuario puede acceder antes de haber iniciado sesión.
- **reviews:** Gestiona las reseñas de los *fanfics*.

- **users:** Gestiona las acciones relacionadas con los usuarios de la aplicación.

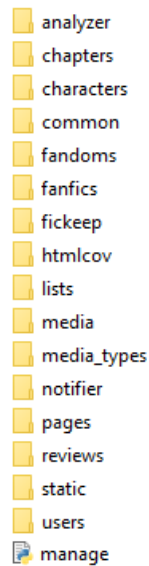


Figura 24: Estructura del proyecto

La carpeta *htmlcov* mostrada en la Figura 24 contiene el porcentaje de código cubierto por los tests ejecutados, *media* contendrá las imágenes de la aplicación, y *static* los recursos *javascript* y *CSS*.

### 9.2.1. Peticiones

Se ha intentado seguir una metodología en la creación de los enlaces y urls, utilizado *GET*, *POST*, *DELETE* y *PUT*.

*GET* se utilizará para pedir la carga de páginas web, para filtrar resultados, y para indicar un término de búsqueda. *POST* se utilizará para crear

nuevos recursos, *PUT* para actualizar estos, y por último, *DELETE* para borrar recursos.

Este sistema genera un problema, ya que *Django* no soporta peticiones *PUT* y *DELETE*. La solución aplicada consiste en añadir un campo invisible a los formularios *POST* que indique el tipo de petición (Figura 25) a realizar.

Cuando el formulario se envíe, la clase correspondiente que lo gestione (Figura 26) deberá heredar de otra anteriormente creada (Figura 27), que compruebe el método especificado en el campo oculto, y así redirija la petición al método correspondiente de la clase.

```
<form action="{% url 'chapters_fanfics:chapters' fanfic_id=fanfic.id%}"
method="post">
  {% csrf_token %}
  <input type="hidden" name="_method" value="delete">
  <button onclick="return confirm('Are you sure you ' +
'want to delete your data for this fanfic?')"
  type="submit">Clear
  all</button>
</form>
```

Figura 25: Formulario para petición *DELETE*

```
class ChaptersView(LoginRequiredMixin, BaseView):
    def delete(self, request, fanfic_id=None):=

    def post(self, request, fanfic_id=None):=
```

Figura 26: Clase que implementa *DELETE*

```

class BaseView(View):
    """ Allow methods like DELETE and PUT """

    def dispatch(self, request, *args, **kwargs):
        method = self.request.POST.get('_method', '').lower()
        handler = None
        if method == 'put':
            handler = getattr(self, "put", self.http_method_not_allowed)
        if method == 'delete':
            handler = getattr(self, "delete", self.http_method_not_allowed)
        if handler is not None:
            return handler(request, *args, **kwargs)
        return super().dispatch(request, *args, **kwargs)

```

Figura 27: Clase creada para la redirección de *DELETE* y *PUT*

### 9.2.2. Tareas asíncronas

La librería usada en la creación y ejecución de tareas asíncronas es *Celery*. A continuación se va a detallar la forma de implementar la tarea de importar *fanfics* al sistema.

Primero, se añade *Celery* al proyecto siguiendo las instrucciones oportunas, y según la configuración de la librería, esta descubrirá automáticamente todas las tareas que se encuentren dentro de ficheros de nombre *tasks.py*. Por lo tanto, habrá que elegir la aplicación donde se implementará esta tarea asíncrona mencionada, y se ha elegido *common*, al necesitar modelos de diferentes aplicaciones del proyecto.

Tras crear el fichero *tasks* en *common*, se procede a implementar la tarea. Dicha tarea se llamará *scrape\_and\_add\_fanfic* y recibirá dos parámetros: la url del *fanfic* a añadir, y el *id* del usuario que realiza la petición. Para más detalle, la lógica de este método se encuentra en los diagramas de clase y secuencia de los apartados anteriores.

Como esta tarea realizará peticiones para conseguir el código de una web, se permitirán cinco intentos antes de considerar la ejecución un fallo. Cada vez que produzca un error, este se capturará mediante un *try/except*, y se usará el método *retry* (Figura 28) para volver a ejecutar la tarea pasados 60 segundos.

```
except Exception as e:
    logger.error("Error in celery task scraping website {}".format(e))
    logger.error(traceback.print_exc())
    scrape_and_add_fanfic.retry(exc=e,
                               countdown=60)
```

Figura 28: Bloque excepción de la tarea asíncrona

La tarea se define de la siguiente forma:

```
@task(name="scrape_and_add_fanfic", max_retries=5)
def scrape_and_add_fanfic(url, current_user):
```

El número de intentos ya realizados se corresponde con el siguiente código:

```
number_of_retries_done = int(scrape_and_add_fanfic.request.retries)
```

La forma de iniciar la tarea desde código síncrono consiste en añadir *delay()* al nombre del método (Figura 29). Esto añadirá la tarea a la cola para ser ejecutada.

```
scrape_and_add_fanfic.delay(url_fanfic, current_user)
```

Figura 29: Inicializar tarea asíncrona

Gracias a la extensión *django\_celery\_results*, los resultados de las tareas asíncronas se guardaran en la base de datos y serán visibles mediante el panel de administración.

Las tareas a ejecutar de forma periódica contarán con una cabecera diferente:

```
@periodic_task(run_every=crontab(minute=0, hour='*/72'))
def run_recommender_system():
```

El código anterior se ejecutará cada 72 horas.

Antes de poder ejecutar estas tareas anteriores, es necesario tener instalado un servidor *redis*. Para instalar este servidor, se necesita tener el servicio *WSL (Windows Subsystem for Linux)* activado. Tras activarlo habrá que ejecutar lo siguiente en la consola de comandos:

```
sudo service redis-server start
```

Ahora se necesitan iniciar los *workers* que procesarán e iniciarán las tareas. Se abrirá una consola de comandos de Windows con el entorno virtual correspondiente, y se introducirá el siguiente código:

```
celery -A fickeep worker -l info -P gevent
```

Las tareas periódicas necesitan de *beats* que indiquen cuando estas deben ser ejecutadas. De esta manera, se ejecutará el siguiente código en otra consola de comandos de Windows:

```
celery -A fickeep beat -l info
```

En resumen, se tendrán que tener tres consolas de comandos abiertas, y cuando un usuario intente añadir un *fanfic*, el *worker* registra la tarea (Figura 30) y procede a ejecutarla cuando cree necesario. Se recibe una notificación (Figura 31) cuando su ejecución termina para indicar el estado del *fanfic* que se ha querido añadir.

```
[2019-06-10 12:51:26,386: INFO/MainProcess] Connected to redis://localhost:6379/0
[2019-06-10 12:51:26,395: INFO/MainProcess] mingle: searching for neighbors
[2019-06-10 12:51:27,427: INFO/MainProcess] mingle: all alone
[2019-06-10 12:51:27,469: INFO/MainProcess] pidbox: Connected to redis://localhost:6379/0.
[2019-06-10 12:51:27,475: INFO/MainProcess] celery@PAULA-PC ready.
[2019-06-10 12:51:31,474: INFO/MainProcess] Received task: scrape_and_add_fanfic[07a7cf1d-4d45-44bc-9c9d-6382e58d745e] ETA:[2019-06-10 13:51:55.457893+03:00]
```

Figura 30: El *worker* recibe una tarea

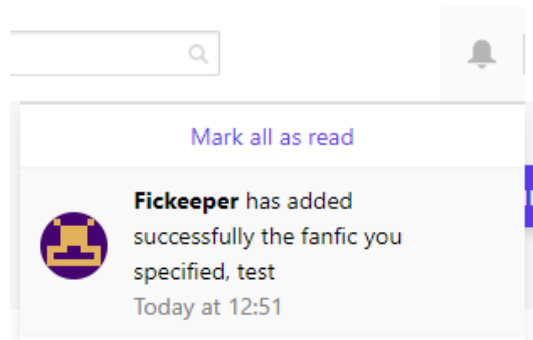


Figura 31: Notificación de *fanfic* añadido

Se comprobará cada 6 horas la existencia de nuevos capítulos de los *fanfics* del sistema, y cada 72 horas se volverá a formar el sistema de recomendaciones.

La versión de *Celery* utilizada (4.2.1) no es oficialmente compatible con Microsoft Windows, y esto no se supo hasta después de tener implementadas

las tareas necesarias para ejecutar las funcionalidades asíncronas anteriores.

Esta incompatibilidad se descubrió cuando una de las veces que se intentó ejecutar el “worker” apareció el siguiente error:

```
ValueError: need more than 0 values to unpack.
```

En vez de volver a programar las tareas en una versión anterior de *Celery*, se descubrió que utilizar el parámetro “-P gevent” en la ejecución del “worker” de *Celery* resolvía el problema.

### 9.2.3. Notificaciones

Las notificaciones se han desarrollado mediante la librería *Django Channels* y señales de *Django*.

La librería *Django Channels* cuenta con dos elementos: canales y *consumers*.

Los *consumers* son clases similares a las *vistas* de *Django*, que permitirán recibir y enviar datos entre el servidor y el usuario. Recordemos que la conexión es bidireccional. Por otro lado, los canales serán una línea de comunicación entre un *consumer* y un usuario.

Cuando un usuario abra la aplicación, un código *javascript* pedirá abrir una conexión de tipo *WebSocket* con el servidor (Figura 32), es decir, cada vez que un usuario abra la aplicación web en una pestaña diferente, se creará un canal entre el usuario y un *consumer*. Esta conexión la administrará *Django Channels* mediante el fichero *routing.py* de la aplicación correspondiente, en este caso *notifier*.



```
window.notifierSocket = new WebSocket(  
    'ws://' + window.location.host +  
    '/ws/notification/');
```

Figura 32: Petición conexión *WebSocket*

Si un usuario tiene abiertas dos páginas de la aplicación, se quiere que la notificación se reciba en las dos, por este motivo será necesario que los *consumers* se comuniquen entre ellos. *Django Channels* facilita esta comunicación con la creación de grupos o capas. Cada usuario tendrá un conjunto de *consumers* a los que estará conectado, y sus canales pertenecerán a un mismo grupo. Cuando se envíe datos a un grupo, todos los usuarios asociados a los canales del grupo recibirán el mensaje.

La estructura de un *consumer* se puede apreciar en la Figura 33.

```

class NotificationConsumer(WebSocketConsumer):
    def connect(self):
        self.accept()
        self.user = self.scope["user"]
        self.group_name = "notifier_" + str(self.user.id)
        async_to_sync(self.channel_layer.group_add)(
            self.group_name, self.channel_name)

    def disconnect(self, close_code):
        async_to_sync(self.channel_layer.group_discard)(
            self.group_name, self.channel_name)

    def message_new(self, event):=

    def receive(self, text_data):=

    def send_data(self, text_data):
        group_name = "notifier_" + str(self.user.id)

        async_to_sync(self.channel_layer.group_send)(
            group_name,
            {
                "type": "message.new",
                "text": text_data,
            },
        )

```

Figura 33: Estructura de un *consumer*

- **connect(self):** El usuario abre un canal y se conecta a un *consumer* (Figura 32). Se obtiene el *id* del usuario, y se añade el canal de comunicación a un grupo que solamente contiene usuarios con el mismo *id*, es decir, un grupo compuesto únicamente por él. El nombre del grupo será “notifier\_«user\_id»”.
- **disconnect(self):** El usuario quiere desconectarse del canal. Se elimina el canal del grupo.
- **message\_new(self, event):** Enviar un mensaje a un solo canal.
- **receive(self, text\_data):** Se reciben datos por parte del usuario. Se ejecutará cuando el usuario: seleccione el icono de las notificaciones para

dejar de mostrar el número, seleccione “Mark all as read”, o marque una notificación como “leída”. También se envía una respuesta para asegurar la correcta ejecución de la acción que ha intentado realizar.

- **send\_data(self, text\_data):** Método a ejecutar cuando se quieran enviar datos a un grupo. El valor especificado en *type* indica que por cada canal del grupo, se utilizará el método *message\_new* para enviar el mensaje.

El usuario recibe los datos mediante código *javascript*.

Cuando se crea un objeto de tipo *Notification*, una señal de *Django* (Figura 34) registrará el evento y enviará el mensaje al usuario correspondiente, utilizando los métodos vistos en la Figura 33.

```
channel_layer = get_channel_layer()

def send_new_notification_to_user(sender, instance, created, **kwargs):=

post_save.connect(send_new_notification_to_user,
                  sender=Notification)
```

Figura 34: Señal de las notificaciones

#### 9.2.4. Panel de administración

Solo los usuarios que formen parte del *staff* podrán acceder al panel de administración. El grupo de *bibliotecarios* estará formado por aquellos usuarios que puedan acceder al panel de administración pero que no sean administradores. Dicho grupo tendrá asignados un conjunto de permisos de *Django*, para permitir la adición, modificación y eliminación de ciertos objetos.

El panel de administración se personaliza mediante el fichero *admin.py* y formularios. El primer paso es registrar el modelo a gestionar de forma interactiva, también se pueden personalizar los campos a mostrar, y se utilizan formularios para cambiar la creación y modificación de objetos.

Las opciones del panel de administración se configuran sobrescribiendo atributos y métodos heredados de las clases padre.

Un ejemplo de atributos y métodos sobrescritos se encuentra en la página de la Figura 35, que a su vez se corresponde con el código de la Figura 36.

Select Characters in Fanfics to view

ADD CHARACTERS IN FANFICS +

Q |  Search

Action:  Go 0 of 19 selected

<input type="checkbox"/>	ID	FANFIC	CHARACTER	PAIRING ID
<input type="checkbox"/>	62	<a href="#">Sherlolly Drabbles (Sherlock (TV))</a>	Molly Hooper	3
<input type="checkbox"/>	61	<a href="#">Sherlolly Drabbles (Sherlock (TV))</a>	Sherlock Holmes	3
<input type="checkbox"/>	60	<a href="#">Lets Cheers to This (The Vampire Diaries (TV))</a>	Stefan Salvatore	2
<input type="checkbox"/>	59	<a href="#">Lets Cheers to This (The Vampire Diaries (TV))</a>	OC	2
<input type="checkbox"/>	57	<a href="#">An Unplanned Life (Avengers)</a>	Loki	
<input type="checkbox"/>	56	<a href="#">That's What You Get When You Let Your Heart Win (Avengers)</a>	Loki	
<input type="checkbox"/>	55	<a href="#">Only Fools Love Monsters (Avengers)</a>	Loki	
<input type="checkbox"/>	54	<a href="#">Great Expectations (Loki Story) (Avengers)</a>	Loki	

**FILTER**

By media type

- All
- anime
- books
- cartoons
- comics
- games
- movies
- other
- plays
- tv

By fandom

- All
- 2012 (2009)
- A Series of Unfortunate Events

Figura 35: Una página del panel de administración

```

@admin.register(CharacterFanic)
class CharacterFanicAdmin(admin.ModelAdmin):
    list_display = ('id', 'fanfic', 'character', 'pairing_id')
    list_display_links = ('fanfic',)

    list_filter = (
        TypeListFilterCharacterFanic,
        FandomListFilterCharacterFanic)

    search_fields = ('character__name_surname', 'fanfic__name')

    fields = ('fanfic', 'character')

    actions = ['create_pairing', 'delete_pairing']

    def has_change_permission(self, request, obj=None):=

    def pairing_id(self, obj):=

    def formfield_for_dbfield(self, db_field, request, **kwargs):=

    def create_pairing(self, request, queryset):=

    create_pairing.short_description = "Create a pairing between the " \
        "characters"

```

Figura 36: Código de una página del panel de administración

Los aspectos más importantes:

- **list\_display:** Los campos a mostrar en la tabla.
- **list\_filter:** Se han creado filtros propios (Figura 37) para poder mostrar los diferentes tipos y *fandoms* en el panel lateral.
- **search\_fields:** Los campos que se comprobarán si el usuario introduce un término a buscar.
- **actions:** Acciones personalizadas que se pueden aplicar a uno o varios

objetos al mismo tiempo.

- **formfield\_for\_dbfield:** Sobrescribir el método que se encarga de elegir los *widgets* correspondientes para el formulario de creación y edición de objetos (Figura 38). En esta ocasión, el *widget* que se ha utilizado es uno creado que permite añadir clases al elemento *HTML* final. Cada *widget* creado contará con sus propios ficheros *css* y *js*.

```
class FandomListFilterCharacterFanfic(admin.SimpleListFilter):
    # Human-readable title which will be displayed in the
    # right admin sidebar just above the filter options.
    title = 'fandom'

    # Parameter for the filter that will be used in the URL query.
    parameter_name = 'fandom'

    def lookups(self, request, model_admin):=

    def queryset(self, request, queryset):=
```

Figura 37: Filtro creado para mostrar los *fandoms*

```
def formfield_for_dbfield(self, db_field, request, **kwargs):
    if db_field.name == "character":
        # change widget
        kwargs['widget'] = SelectCustomWidget(
            attrs={'class': 'custom-select character-select'})
    elif db_field.name == "fanfic":
        # change widget
        kwargs['widget'] = SelectCustomWidget(
            attrs={'class': 'custom-select fanfic-select'})
    return super().formfield_for_dbfield(db_field, request, **kwargs)
```

Figura 38: *Widgets* personalizados

Realizar las funciones administrativas ha resultado complicado, ya que no se ha encontrado documentación para realizar algunas de las acciones que se tenían en mente. Por lo tanto, se han tenido que mirar bastantes clases de *Django* para saber lo que sobrescribir. También se han tenido que modificar métodos propios de los formularios, para cambiar la forma de limpiar, validar y guardar los datos.

Otra de las dificultades ha sido la complejidad de las acciones a implementar. Un ejemplo es el cambio de *fandom* de un *fanfic*. Se recuerda que los *fanfics* podían tener dos *fandoms*; uno primario y otro secundario. Para este escenario, por cada *pairing* asociado a ese *fanfic*, es necesario comprobar que los dos personajes siguen perteneciendo a uno de los *fandoms* asociados al *fanfic*.

#### 9.2.5. Tests

En el apartado 10 se pueden encontrar los resultados de los tests implementados.

Algunos de los métodos que se han tenido que probar contenían la condición de tener que ser peticiones *ajax*, o iniciaban una tarea asíncrona, que para la ejecución de las pruebas no interesaba.

Esta situación se ha evitado utilizando *mocks*. Un ejemplo de su uso se aprecia en la Figura 39. El método mostrado en la siguiente figura pretende probar que un *fanfic* se añade correctamente a la cola de tareas cuando la dirección web introducida es correcta.

```

@mock.patch('fanfics.custom_models.FanficWeb.get_cleaned_url')
@mock.patch('common.tasks.scrape_and_add_fanfic.delay')
@mock.patch('fanfics.custom_models.FanficWeb.check_if_online')
@mock.patch('fanfics.custom_models.FanficWeb.url_without_errors')
def test_post_external_add_good_url(self, mock_url_without_errors,
                                   mock_check_if_online, mock_scrape,
                                   mock_cleaned_url):

    """ Try to add fanfic """
    mock_url_without_errors.return_value = "http://web-ok.com"
    mock_check_if_online.return_value = True
    mock_scrape.return_value = ""
    mock_cleaned_url.return_value = "http://web-ok.com"

    post_data = {'url_fanfic': 'http://web-ok.com'}

    response = self.client.post(reverse('fanfics:external_add'),
                                data=post_data)
    self.assertRedirects(response, reverse('fanfics:external_done'))

```

Figura 39: *Mocks* en un método a probar

En la cabecera del método están aquellos *paths* de las funciones que se pretenden *mockear* o sobrescribir su resultado. Los resultados de estas funciones se definen en las primeras líneas del método.

De esta manera no se depende de fuentes externas para comprobar la buena funcionalidad del sistema.

### 9.2.6. Otros

- Se ha utilizado la guía de estilos PEP8 en el código escrito en *Python*. El límite de caracteres por línea es de 79.
- Se han utilizado validadores propios para que un usuario no pueda tener cero años (Figura 40).



```

def date_of_birth_validator(date_param):
    """ Check if a user's age is zero or less """
    today = date.today()
    born = date_param
    years_now = today.year - born.year - \
        ((today.month, today.day) < (born.month, born.day))
    if years_now < 15:
        raise ValidationError(
            _('You cannot have less than 15 years'),
            params={'date_param': date_param},
        )
    elif years_now > 100:
        raise ValidationError(
            _('You cannot have more than 100 years'),
            params={'date_param': date_param}
        )

```

Figura 40: Validador creado para comprobar la edad de un usuario

- Al instalar el conector MySQL-*Django* mediante *pip* se produjo un error, y la única solución fue descargarse el fichero correspondiente en formato “.wheel” e instalarlo utilizando ese fichero.
- Fue necesario instalar la librería *Pillow* para que el campo de la imagen de usuarios y *fandoms* funcionase.
- El uso de transacciones de *Django* en operaciones de la base de datos. Si un error ocurriese, todos los cambios harían *rollback* o se desharían.

```

try:
    with transaction.atomic():
        # código
except Exception as e:
    # error

```

- También se evitaron errores de importación circulares, utilizando el siguiente código en vez de importar de forma tradicional la clase necesaria.

```
from django.apps import apps
modelo_importado = apps.get_model('app_name.model_name')
```

- Se ha creado un decorador propio (Figura 41) para comprobar que un usuario no está identificado. Este decorador se usará en las páginas que solo verán los usuarios no identificados, como la página de aterrizaje.

```
def anonymous_required(redirect_url='common:home'):
    """
    Decorator for views that checks that the user is not logged in, redirecting
    to the home page if it's logged in.
    """
    actual_decorator = user_passes_test(
        lambda u: u.is_anonymous,
        login_url=redirect_url,
        redirect_field_name=None
    )
    return actual_decorator
```

Figura 41: Decorador creado

- Algunas de las páginas de *fanfics* necesitan de autenticación para poder ver algunas de las historias. La solución ha sido crear un usuario específico para esta aplicación web, y utilizar *cookies* para obtener la página del *fanfic* introducido, después de haber iniciado sesión con la petición correspondiente (Figura 42).

```

if auth_url is not None:
    data = {'username': FANFIC_USERNAME,
           'password': FANFIC_PASSWORD}
    r = requests.post(auth_url, data=data)
    session_cookie = r.cookies.get(cookie_name)

    cookies = {cookie_name: session_cookie}

page = requests.get(url, cookies=cookies,
                   timeout=(connect_timeout, read_timeout)).text
page_html = BeautifulSoup(page, 'html.parser')

```

Figura 42: Autenticación y obtención del código HTML de una página web

## 9.3. Ciencia de datos

### 9.3.1. Sistema de recomendaciones

De forma general, los sistemas de recomendaciones se pueden clasificar en tres tipos:

- **Sistemas simples:** la lista de recomendaciones es igual para todos los usuarios y se forma según un criterio (popularidad, visitas,..). Un ejemplo: las 10 películas más vistas.
- **Sistemas de recomendación basados en el contenido:** este tipo de sistemas consisten en sugerir productos similares a uno dado. Para ello se utilizan los metadatos (Ejempl de película: director, actores, géneros, etc.), y la idea es que si a una persona le gusta un producto, también le gustarán aquellos que sean similares a él.
- **Sistemas de filtrado colaborativo:** intentan predecir la calificación que un usuario asignaría a un producto conociendo su historial.

Se ha elegido implementar el *sistema de recomendación basado en el contenido*, ya que el primero no utiliza recomendaciones personalizadas, y el último cuenta con el problema de “inicio frío” o “cold start”. Este término hace referencia a la falta de datos para lograr buenas recomendaciones, debido a que la aplicación es nueva y ahora mismo no hay información disponible de los usuarios.

El primer paso ha sido obtener los datos a utilizar. Estos datos se encuentran esparcidos por la base de datos, así que el objetivo principal de este primer paso será juntarlos. Por cada *fanfic*, se quiere obtener: el autor, los géneros, el idioma, los personajes y el *fandom*.

Se han ido generando *dataframes*, gracias a la librería *Panda*, y se han juntado las columnas para que todos los datos de un mismo *fanfic* estén en una celda (Figura 43). Los nombres de los *fandoms* y personajes se han juntado, para que ‘Emily Blunt’ no sea igual a ‘Emily Deschanel’.

	data	FanficIndex
0	rom ejdaniels other	2
1	fan rom gary_b doctorwho avengers	4
2	ang rom silversurfer60 doctorwho	5
3	ej-justej naruto doctorwho	6
4	dra drt doctorwho	7
5	kiyohime93 doctorwho	8
6	dra kaybet doctorwho	9
7	dra rom thedoctorstardisxo doctorwho	10
8	dra rom zugrian other	11
9	fan hum usa_tiger other	12
10	ang hum rom marvolo666 other	13
11	queen_serenity other	14
12	hum hussar other	15
13	fan texaswookie other	16
14	texaswookie other	17
15	fan texaswookie other	24
16	fan texaswookie other	25
17	firithariel avengers loki	26
18	firithariel avengers loki	27
19	fireangel76 avengers loki	28
20	fireangel76 avengers loki	29
21	animefreak141 avengers loki	30

Figura 43: Datos juntos de los *fanfics*

Una vez que se tienen los datos, se ha utilizado la librería “CountVectorizer” en vez de “TF-IDF” para preprocesarlos. La diferencia reside en que “CountVectorizer” no restará importancia a términos que aparezcan bastante, y si un género o *fandom* aparece bastante, no se le quiere quitar importancia.

También se obtiene la matriz de similitud del coseno para obtener un número que indique el grado de similitud entre dos *fanfics* (Figura 44).

```

def preprocess_data(self):
    """ Preprocess the data """
    count = CountVectorizer()
    count_matrix = count.fit_transform(self.dataframe['data'])
    self.cosine_sim = cosine_similarity(count_matrix, count_matrix)

```

Figura 44: Método para preprocesar los datos

Por último, se ha creado un método para tener que aportar solamente el *id* del *fanfic* para conseguir las recomendaciones oportunas. En un futuro, cuando haya añadidos más *fanfics*, habría que revisar este sistema para comprobar su eficacia. En la Figura 45 se puede apreciar como para un *fanfic* del *fandom* “Doctor Who”, las tres primeras recomendaciones son *fanfics* de igual *fandom*. Por el momento se puede concluir que el sistema funciona.

The screenshot shows a web interface for a fanfiction recommendation system. At the top, it says "Tv > Doctor Who". Below this, there is a circular profile picture of a man and a woman. To the right of the picture, the title "The Wife Of River Song" is displayed, followed by "by Gary\_B". Below the title, there are five stars and a rating of "0.00". Further down, it says "K - FANTASY/ROMANCE/", "LAST PUBLISHED: NOV 08, 2018 - LAST CHECKED: MAY 28, 2019", and "1 CHAPTERS - 922 WORDS". Below this, it says "HOSTED BY" followed by a URL "HTTPS://FICWAD.COM/STORY/278...". To the right of this, there is a section titled "YOU MAY ALSO LIKE" with three recommendations: "A CHANCE MEETING (DOCTOR WHO)", "AN UNEXPECTED JOURNEY (DOCTOR WHO)", and "LOVE THE ONE YOU'RE (NOT) WITH (AVENGERS)". At the bottom left, there are two buttons: "ADD IT" and "REPORT ERROR".

Figura 45: Recomendaciones de un *fanfic*

### 9.3.2. Visualización de datos de los usuarios

Se han generado dos gráficos para visualizar los datos de los usuarios.

El primer gráfico aporta el número de personas que han añadido *fanfics* de un *fandom* a sus listas en los últimos 30 días. Los resultados se desglosan por género (Figura 46).

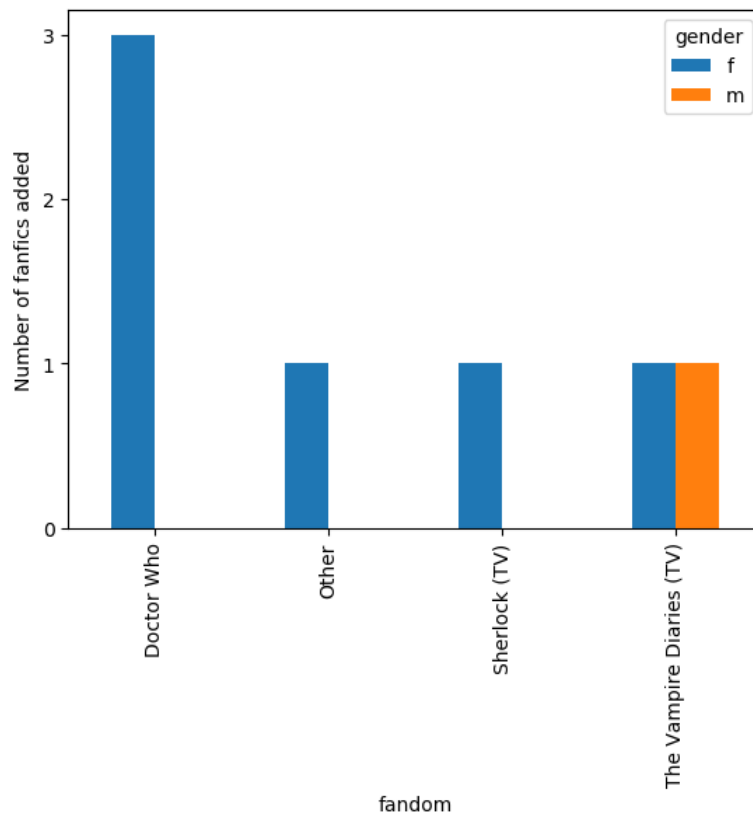


Figura 46: Gráfico

El segundo gráfico muestra información similar, pero según los géneros de los *fanfics* (Figura 47).

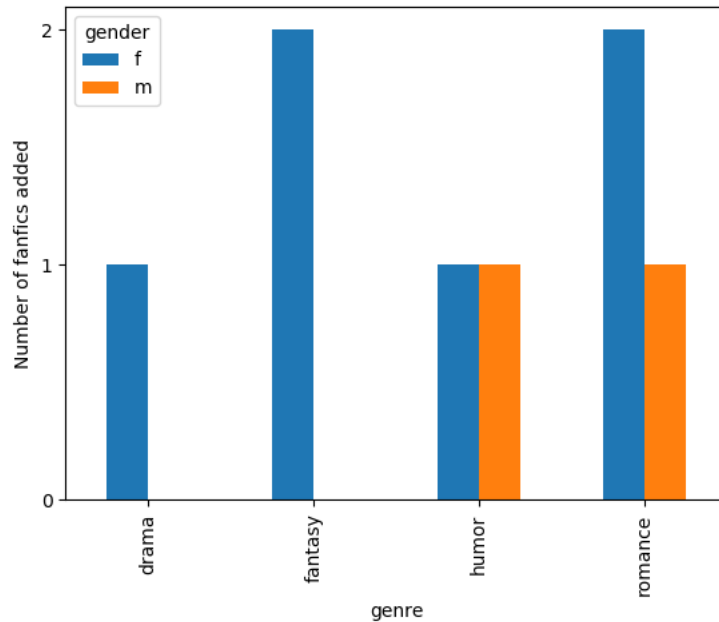


Figura 47: Gráfico

### 9.3.3. Compatibilidad de navegadores

Se puede asegurar un buen funcionamiento de la aplicación web en los siguientes navegadores:

- Google Chrome (Versión 74.0.3729.169)
- Mozilla Firebox (Versión 67.0)
- Opera (Versión 60.0.3255.170)
- Microsoft Edge (Versión 44.17763.1.0)

De todas maneras, se recomienda usar *Google Chrome*, ya que este ha sido el navegador usado para desarrollar la aplicación.



## 10. Verificación y evaluación

Se han programado múltiples pruebas automatizadas utilizando herramientas de *Django*. Estas permitirán simular peticiones *GET* y *POST*, para comprobar: si el estado de una página es 200, si se carga la plantilla correcta, si la ejecución de un método ha sido el adecuado, etc.

También se ha utilizado la librería *Coverage* para conocer que el porcentaje de código cubierto es de 72%.

Tabla 73: Lista de Tests

Código	Descripción	Rdo. esperado	Rdo. real	Comentarios
<b>Vistas</b>				
1	Borrar notas privadas de un usuario en un <i>fanfic</i>	Las notas se han borrado	Las notas no se han borrado	Se ha puesto mal la url
1a			Las notas no se han borrado	Se ha especificado mal la petición al servidor
1b			Las notas se han borrado	Correcto
2	Crear nota privada	La nota se ha creado	La nota se ha creado	Correcto
3	Editar nota privada: la nota no está vacía	La nota se ha actualizado	La nota no se ha actualizado	Se ha utilizado “load” en vez de “loads” al cargar la respuesta del servidor
3a			La nota se ha actualizado	Correcto
4	Editar nota privada: la nota está vacía	La nota se ha actualizado	La nota se ha actualizado	Correcto

Continuación de la Tabla 73

Código	Descripción	Rdo. esperado	Rdo. real	Comentarios
5	Borrar todos los datos de un <i>fanfic</i> de un usuario	Los datos se han borrado	Los datos se han borrado	Correcto
6	Marcar todos los capítulos de un <i>fanfic</i> de un usuario como leídos	Los capítulos están leídos	Los capítulos están leídos	Correcto
7	Marcar un capítulo como leído	El capítulo está leído	El capítulo está leído	Correcto
8	Marcar un capítulo como no leído	El capítulo no está leído	El capítulo no está leído	Correcto
9	Marcar último capítulo no leído como leído	El capítulo está leído	El capítulo está leído	Correcto
10	El <i>dashboard</i> se carga correctamente	La página se carga correctamente	La página se carga correctamente	Correcto
11	La página de búsqueda se carga correctamente	La página se carga correctamente	La página se carga correctamente	Correcto
12	Se busca un <i>fanfic</i>	Se obtienen los <i>fanfics</i> con ese término	Se obtienen los <i>fanfics</i> con ese término	Se imprime un error por consola por no ordenar los <i>fanfics</i> del resultado
12a			Se obtienen los <i>fanfics</i> con ese término	Correcto
13	Se busca un <i>fanfic</i> : el término está vacío	El resultado está vacío	El resultado está vacío	Correcto

Continuación de la Tabla 73

Código	Descripción	Rdo. esperado	Rdo. real	Comentarios
14	Se busca un <i>fanfic</i> : no se indica el parámetro necesario en la petición	El resultado está vacío	El resultado está vacío	Correcto
15	Ver los <i>fanfics</i> de un <i>fandom</i>	Se obtienen los <i>fanfics</i> de un <i>fandom</i>	Se obtienen los <i>fanfics</i> de un <i>fandom</i>	Correcto
16	Filtrar y ver los <i>fanfics</i> de un <i>fandom</i>	Se obtienen los <i>fanfics</i> de un <i>fandom</i> filtrados	Se obtienen los <i>fanfics</i> de un <i>fandom</i> filtrados	Correcto
17	La página para añadir un <i>fanfic</i> externo se carga correctamente	La página se carga correctamente	La página se carga correctamente	Correcto
18	Añadir un <i>fanfic</i> externo: la web especificada es errónea	La página se carga correctamente con un mensaje de error	La página se carga correctamente con un mensaje de error	Correcto
19	Añadir un <i>fanfic</i> externo: el <i>fanfic</i> ya está añadido	La página se carga correctamente con un mensaje de error	La página se carga correctamente con un mensaje de error	Correcto
20	Añadir un <i>fanfic</i> externo: el <i>fanfic</i> no está añadido y la web es correcta	La página se carga correctamente y se añade una tarea a la cola	La página se carga correctamente y se añade una tarea a la cola	Correcto
21	La página de un <i>fanfic</i> se carga correctamente	La página se carga correctamente	La página se carga correctamente	Correcto
22	Se avisa de un error en un <i>fanfic</i>	El error se ha guardado	El error se ha guardado	Correcto

Continuación de la Tabla 73

Código	Descripción	Rdo. esperado	Rdo. real	Comentarios
23	Obtener los personajes que pueden estar asociados a un <i>fanfic</i>	Los personajes son correctos	Los personajes son correctos	Correcto
24	Añadir un <i>fanfic</i> a una lista	El <i>fanfic</i> se ha añadido correctamente	El <i>fanfic</i> se ha añadido correctamente	Correcto
25	Añadir un <i>fanfic</i> a una lista: ya está en la lista	El <i>fanfic</i> está en la lista	El <i>fanfic</i> está en la lista	Correcto
26	Borrar un <i>fanfic</i> de una lista: ya no está en la lista	El <i>fanfic</i> no está en esa lista	Error	Se ha importado otra librería de tests, en vez de utilizar los tests de <i>Django</i>
26a			El <i>fanfic</i> no está en esa lista	Correcto
27	Borrar un <i>fanfic</i> de una lista: está en la lista	El <i>fanfic</i> se ha borrado de la lista	El <i>fanfic</i> se ha borrado de la lista	Correcto
28	Crear una lista	La lista se ha creado	La lista se ha creado	Correcto
29	Actualizar el nombre de una lista	La lista se ha actualizado	La lista se ha actualizado	Correcto
30	Borrar una lista	La lista se ha borrado	La lista se ha borrado	Correcto
31	Obtener los <i>fandoms</i> de un tipo	Los <i>fandoms</i> se obtienen correctamente	Los <i>fandoms</i> se obtienen correctamente	Correcto
32	La página de aterrizaje se carga correctamente	La página se carga correctamente	La página se carga correctamente	Correcto

Continuación de la Tabla 73

Código	Descripción	Rdo. esperado	Rdo. real	Comentarios
33	La página de edición de una reseña se carga correctamente	La página se carga correctamente	La página se carga correctamente	Correcto
34	Actualizar reseña: datos correctos	La reseña se ha actualizado correctamente	La reseña se ha actualizado correctamente	Correcto
35	Actualizar reseña: datos incorrectos	La reseña no se ha actualizado	La reseña no se ha actualizado	Correcto
36	Actualizar reseña: texto vacío	La reseña no se ha actualizado	La reseña no se ha actualizado	Correcto
37	La página de una reseña se carga correctamente	La página se carga correctamente	La página se carga correctamente	Correcto
38	Crear una reseña	La reseña se ha guardado correctamente	La reseña se ha guardado correctamente	Correcto
39	Borrar una reseña	La reseña se ha borrado correctamente	La reseña se ha borrado correctamente	Correcto
40	La página de registrarse se carga correctamente	La página se carga correctamente	La página se carga correctamente	Correcto
41	Comprobar si el nombre de usuario ya existe: el usuario existe	El usuario existe	El usuario existe	Correcto
42	Comprobar si el nombre de usuario ya existe: el usuario no existe	El usuario no existe	El usuario no existe	Correcto
43	La página de identificarse se carga correctamente	La página se carga correctamente	La página se carga correctamente	Correcto

Continuación de la Tabla 73

Código	Descripción	Rdo. esperado	Rdo. real	Comentarios
44	Cerrar la sesión de un usuario	La sesión se cierra correctamente	La sesión se cierra correctamente	Correcto
45	La página de configuración de la cuenta de un usuario se carga correctamente	La página se carga correctamente	La página se carga correctamente	Correcto
46	Actualizar la configuración de la cuenta de un usuario: datos válidos	La configuración se ha actualizado	La configuración se ha actualizado	Correcto
47	La página del perfil se carga correctamente	La página se carga correctamente	La página se carga correctamente	Correcto
48	Comprobar si un usuario sigue a otro	El usuario está siguiendo al otro usuario	El usuario no está siguiendo al otro usuario	La respuesta no estaba en formato texto, había un error en la comparación.
48a			El usuario está siguiendo al otro usuario	Correcto
49	La página que muestra los <i>fanfics</i> de un usuario se carga correctamente	La página se carga correctamente	La página se carga correctamente	Correcto
50	La página que muestra los <i>fanfics</i> de un usuario filtrados se carga correctamente	La página se carga correctamente	La página se carga correctamente	Correcto

Formularios

Continuación de la Tabla 73

Código	Descripción	Rdo. esperado	Rdo. real	Comentarios
51	Crear capítulo: datos correctos	Formulario válido	Formulario válido	Correcto
52	Crear capítulo: número incorrecto	Formulario no válido	Formulario no válido	Correcto
53	Crear personaje: datos correctos	Formulario válido	Formulario válido	Correcto
54	Parámetros correctos filtrando <i>fanfics</i>	Formulario válido	Formulario válido	Correcto
55	Crear mensaje de error de un <i>fanfic</i> : mensaje válido	Formulario válido	Formulario válido	Correcto
56	Crear mensaje de error de un <i>fanfic</i> : mensaje inválido	Formulario inválido	Formulario inválido	Correcto
57	Crear usuario: datos válidos	Formulario válido	Formulario válido	Correcto
58	Crear usuario: fecha de nacimiento inválida	Formulario inválido	Formulario inválido	Correcto
59	Editar usuario: datos válidos	Formulario válido	Formulario válido	Correcto
60	Editar usuario: país inválido	Formulario inválido	Formulario inválido	Correcto
<b>Modelos</b>				
61	El nombre de un <i>fanfic</i> se obtiene correctamente de la base de datos	El nombre es correcto	El nombre es correcto	Correcto
62	La web de un <i>fanfic</i> se obtiene correctamente de la base de datos	La web es correcta	La web es correcta	Correcto

Continuación de la Tabla 73

Código	Descripción	Rdo. esperado	Rdo. real	Comentarios
63	Comprobar si un género entra dentro de las opciones establecidas: género no correcto	El género no es correcto	El género no es correcto	Correcto
64	Comprobar si un género entra dentro de las opciones establecidas: género correcto	El género es correcto	El género es correcto	Correcto
65	Obtener la abreviatura de un género	La abreviatura es correcta	La abreviatura es correcta	Correcto
66	Obtener el nombre de usuario	El nombre de usuario es correcto	El nombre de usuario es correcto	Correcto
67	Obtener el número de <i>fanfics</i> añadidos por un usuario	El número de <i>fanfics</i> es correcto	El número de <i>fanfics</i> es correcto	Correcto
68	Obtener el número de reseñas escritas por un usuario	El número de reseñas es correcto	El número de reseñas es correcto	Correcto
69	Obtener el número de seguidores de un usuario	El número de seguidores es correcto	El número de seguidores es correcto	Correcto
70	Obtener el número de usuarios seguidos por un usuario	El número de usuarios seguidos es correcto	El número de usuarios seguidos es correcto	Correcto
<b>Clases propias</b>				
71	Obtener nombre del sitio web de un <i>fanfic</i> : es Ficwad	El nombre es correcto	El nombre es correcto	Correcto



Continuación de la Tabla 73

Código	Descripción	Rdo. esperado	Rdo. real	Comentarios
72	Obtener nombre del sitio web de un <i>fanfic</i> : es AvengersFanfiction	El nombre es correcto	El nombre es correcto	Correcto
73	Obtener nombre del sitio web de un <i>fanfic</i> : es ArchiveofourOwn	El nombre es correcto	El nombre es correcto	Correcto
74	Obtener nombre del sitio web de un <i>fanfic</i> : es incorrecto	El nombre es incorrecto	El nombre es incorrecto	Correcto
75	Comprobar si el formato de un enlace de un <i>fanfic</i> es correcto: es correcto	El enlace es correcto	El enlace es correcto	Correcto
76	Comprobar si el formato de un enlace de un <i>fanfic</i> es correcto: es incorrecto	El enlace es incorrecto	El enlace es incorrecto	Correcto
77	Comprobar si una web está disponible: está disponible	La web está disponible	La web está disponible	Correcto
78	Obtener el título y autor de un <i>fanfic</i> mediante su web	El título y autor son correctos	El título y autor son correctos	Correcto
79	Obtener el <i>fandom</i> y tipo de un <i>fanfic</i> mediante su web	El <i>fandom</i> y tipo son correctos	El <i>fandom</i> y tipo son correctos	Correcto
80	Obtener el lenguaje de un <i>fanfic</i> mediante su web	El lenguaje es correcto	El lenguaje es correcto	Correcto

**Continuación de la Tabla 73**

<b>Código</b>	<b>Descripción</b>	<b>Rdo. esperado</b>	<b>Rdo. real</b>	<b>Comentarios</b>
81	Obtener los géneros de un <i>fanfic</i> mediante su web	Los géneros son correctos	Los géneros son correctos	Correcto
82	Obtener el estado de un <i>fanfic</i> mediante su web	El estado es correcto	El estado es correcto	Correcto
83	Obtener el <i>rating</i> de un <i>fanfic</i> mediante su web	El <i>rating</i> es correcto	El <i>rating</i> es correcto	Correcto

## 11. Conclusiones

En esta sección se realizará un análisis entre la planificación estimada y la real. También se sugerirán cambios y mejoras para líneas futuras del proyecto.

### 11.1. Análisis entre planificación estimada y real

#### 11.1.1. Objetivos

Los objetivos planteados al comienzo no han variado durante el proyecto. Además, estos se han empezado a cumplir en la ejecución de las tareas del bloque de aprendizaje, y a medida que se avanzaba con la aplicación.

#### 11.1.2. Herramientas

Se pensaba utilizar una extensión de *Django* para realizar el sistema de notificaciones de la aplicación. Cuando llegó la hora de utilizarlo, se encontraron ciertos problemas que no se podían resolver sin modificar el código fuente de dicha aplicación. Por este motivo, se llegó a la conclusión de tener que implementar un sistema de notificaciones propio.

#### 11.1.3. Planificación temporal

La planificación temporal sufrió grandes cambios a partir del mes de febrero de 2019. La estimación de tiempo de las tareas a realizar hasta esta fecha fue bastante precisa, y aunque el número de horas real no fue igual al estimado, los plazos de tiempo estuvieron acertados. El tiempo que sobró se

utilizó para avanzar con la documentación.

En cambio, a partir de febrero de 2019 se tuvieron que realizar ciertos proyectos de programación para varias asignaturas, lo que consumió demasiado tiempo y no se pudo avanzar en gran medida en el proyecto hasta que las clases terminaron el 12 de mayo de 2019. A partir de esta última fecha, se dedicaron aproximadamente 9 horas diarias al proyecto para poder terminarlo para finales de junio.

Los tiempos finales se pueden observar en la Tabla 74, y la planificación temporal real que ha cambiado se aprecia en la Figura 48.

Los tiempos reales fueron mucho menores que los estimados en la mayoría de tareas. En cambio, el bloque de implementación consumió más horas de las previstas, alcanzando aproximadamente las 300 horas. La tarea que sin duda llevó más tiempo fue la de añadir un *fanfic* externo, al tener que implementar un sistema de notificaciones.

Tabla 74: Lista de Tareas resumidas

<b>Id</b>	<b>Nombre de la Tarea</b>	<b>Duración real</b>	<b>Duración estimada</b>
<b>Organización</b>		<b>31,5</b>	<b>50,5</b>
1	Planteamiento de la aplicación	1	1,5
2	Nombrar la aplicación	1	1
3	Planificación de Tareas	6	8
4	Planificación Temporal	13	5
5	Elegir lenguaje de programación	0,5	1
6	Elegir Software	2	2,5
7	Preparación entorno de desarrollo	3	1,5
8	Reuniones con el director del proyecto	5	30
<b>Aprendizaje</b>		<b>70,66</b>	<b>99</b>
9	Aprendizaje de modularidad en Python	1	1
10	Aprendizaje de Django	45,8	56
11	Aprendizaje de CSS Flexbox	7	10

Continuación de la Tabla 74

Id	Nombre de la Tarea	Duración real	Duración estimada
12	Aprendizaje de CSS Grid	5	8
13	Aprendizaje de UI	3,56	9
14	Aprendizaje de UX	6	12
15	Aprendizaje de FicWad.com	1	1
16	Aprendizaje de AvengersFanfiction.com	0,3	1
17	Aprendizaje de ArchiveOfOurOwn.org	1	1
<b>Captura de requisitos</b>		<b>41,4</b>	<b>71,5</b>
18	Listar Funcionalidades	3	4
19	Plantear esquema web	2	9
20	Diseño prototipos low-res	5	15
21	Diseño prototipos high-res	19,3	28
22	Jerarquía de actores	0,1	0,5
23	Casos de uso	9	5
24	Modelo de dominio	3	10
<b>Análisis y Diseño</b>		<b>23,3</b>	<b>36</b>
25	Diagramas de clases	8	15
26	Diagramas de secuencia	12	15
27	Diagramas de la base de datos	3,3	6
<b>Implementación y Desarrollo</b>		<b>297,21</b>	<b>208</b>
28	Registro e identificación de usuarios	20,4	12
29	Página de aterrizaje	11	20
30	Añadir fanfic externo	41,05	25
31	Cambiar configuración de la cuenta	16,36	9
32	Ver perfil del usuario	16	3
33	Ver página de fanfic	26,3	4
34	Comentar en fanfics	15	9
35	Añadir notas a capítulos de los fanfics	6	9
36	Explorar fanfics	9	10
37	Filtrar Fanfics	4,2	15
38	Clasificar Fanfics	9,3	9
39	Ver mis fanfics	13	10

Continuación de la Tabla 74

<b>Id</b>	<b>Nombre de la Tarea</b>	<b>Duración real</b>	<b>Duración estimada</b>
40	Gestionar listas del usuario	9	5
41	Dashboard	12	18
42	Registrar capítulos leídos	4	5
43	Buscar fanfics y usuarios	4	6
44	Avisar de nuevos capítulos	16	25
45	Crear copias de seguridad	2,2	5
46	Funciones administrativas	25,3	9
47	Ciencia de datos	37,1	25
	<b>Pruebas</b>	<b>11,1</b>	<b>50</b>
48	Pruebas del registro e identificación de usuarios	1	2
49	Pruebas de la página de aterrizaje	0,2	1
50	Pruebas de añadir fanfic externo	1	4
51	Pruebas de cambiar configuración de la cuenta	0,3	4
52	Pruebas de ver perfil de usuario	0,2	2
53	Pruebas de ver página de fanfic	0,2	3
54	Pruebas de comentar en fanfics	0,4	1
55	Pruebas de añadir notas a capítulos de los fanfics	1	1
56	Pruebas de explorar fanfics	0,3	1
57	Pruebas de filtrar fanfics	0,4	3
58	Pruebas de clasificar fanfics	0,3	2
59	Pruebas de ver mis fanfics	0,2	3
60	Pruebas de gestionar listas del usuario	0,2	1
61	Pruebas del dashboard	0,2	3
62	Pruebas de registrar capítulos leídos	0,4	1
63	Pruebas de buscar fanfics y usuarios	0,1	2
64	Pruebas de avisar de nuevos capítulos	2	5
65	Pruebas de crear copias de seguridad	0,4	4
66	Pruebas de funciones administrativas	1,3	2

Continuación de la Tabla 74

<b>Id</b>	<b>Nombre de la Tarea</b>	<b>Duración real</b>	<b>Duración estimada</b>
67	Pruebas de ciencia de datos	1	5
<b>Documentación</b>		<b>116</b>	<b>89</b>
68	Redacción de la Memoria	85	55
69	Redacción de Manuales	16	8
70	Preparación de la defensa	15	26
<b>Horas Totales</b>		<b>591,17</b>	<b>604</b>

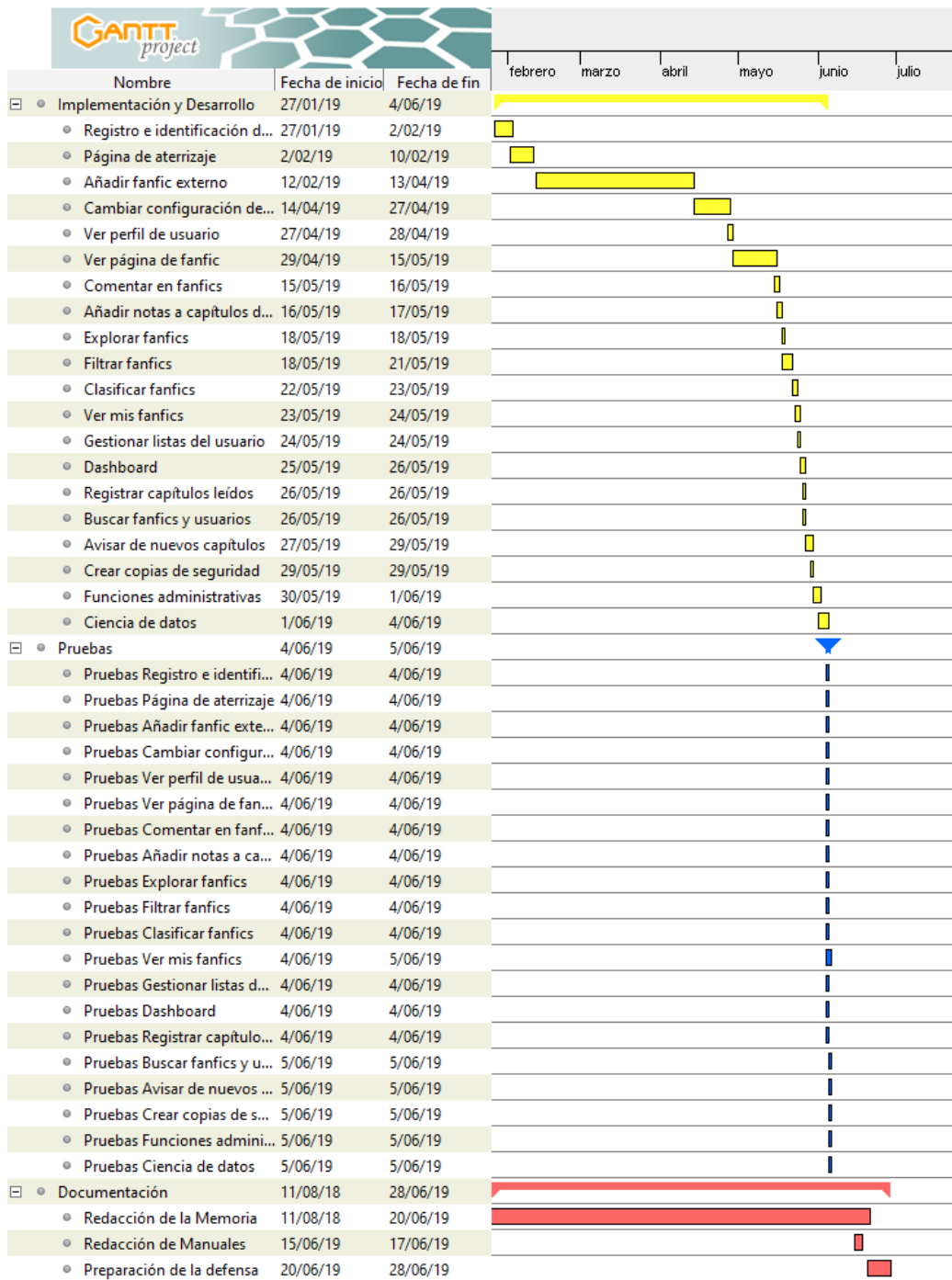


Figura 48: Planificación temporal real que ha cambiado



#### **11.1.4. Evaluación económica**

Al haberse producido un cambio en las horas trabajadas, el gasto final del proyecto es de 10.624,44€. Este gasto no dista mucho del obtenido en el apartado de la evaluación económica.

### **11.2. Líneas futuras**

Durante el proyecto fueron surgiendo nuevas funcionalidades, que aunque interesantes, no se tuvo tiempo para desarrollar.

#### **11.2.1. Identificación externa**

Esta funcionalidad consiste en utilizar productos externos, como bien son *Facebook* o *Twitter*, para permitir al usuario iniciar sesión de forma directa.

#### **11.2.2. Aplicación multidioma**

En mitad del desarrollo se descubrió que *Django* permite realizar esta funcionalidad de forma sencilla, tan solo añadiendo unos caracteres a las palabras y frases a traducir. Sería interesante ofrecer soporte multidioma para que usuarios sin conocimiento de inglés puedan usar el proyecto desarrollado.

#### **11.2.3. Chat entre usuarios**

Los usuarios tendrán la posibilidad de *chatear* entre ellos. Además, gracias al uso de Django Channels, tan solo habría que implementar el código necesario para lograr un *chat* con mensajes en tiempo real.

#### 11.2.4. Extensión de Google Chrome

Desarrollar una extensión de Google Chrome que permita añadir automáticamente una historia presionando su icono. Solo funcionará si la página en la que se está es una de las permitidas.

#### 11.2.5. Listas privadas de *fanfics*

Los usuarios podrán ocultar sus listas de *fanfics*, y aquellos *fanfics* pertenecientes a estas estarán ocultos para el resto de los usuarios en la página del perfil de dicho usuario.

#### 11.2.6. Tests de usabilidad

Sería beneficioso realizar tests de usabilidad usando los prototipos o la propia aplicación para mejorar su usabilidad.

### 11.3. Conclusión personal

A medida que he ido aprendiendo y desarrollando la aplicación, me ha ido gustando más y más la combinación de *Python* y *Django*. No pensaba que desarrollar una aplicación utilizando un *framework* iba a ser más rápido que hacerlo “a mano”, pero esta experiencia me ha abierto los ojos. No solo me ha encantado la utilización de un *framework*, sino que ahora quiero aprender otros como *Laravel*, un *framework* de php.

Por la parte de CSS, creo que las herramientas *Flexbox* y *CSS Grid* no las voy a olvidar y las usaré bastante a partir de ahora, ya que facilitan en gran medida la alineación de los elementos.

Finalmente, gracias al sistema de recomendación realizado, he podido comprobar la utilidad de la *ciencia de datos* de forma práctica, y me han parecido interesantes las diversas aplicaciones que este campo puede ofrecer en proyectos reales.

## Referencias

- [1] Laravel: PHP Framework,  
<https://laravel.com/>  
(último acceso el 02-02-2019)
  
- [2] Django: The web framework for perfectionists with deadlines.,  
<https://www.djangoproject.com/>  
(último acceso el 02-06-2019)
  
- [3] ASGI,  
<https://asgi.readthedocs.io/en/latest/>  
(último acceso el 02-06-2019)
  
- [4] BOE. XVII Convenio colectivo estatal de empresas de consultoría, y estudios de mercados y de la opinión pública,  
<https://www.boe.es/boe/dias/2018/03/06/pdfs/BOE-A-2018-3156.pdf>  
(último acceso el 02-06-2019)
  
- [5] SafeCreative. La protección jurídica de los Fanfiction,  
<http://es.safecreative.net/2009/09/11/la-proteccion-juridica-de-los-fanfiction/>  
(último acceso el 03-06-2019)
  
- [6] Clarisa Peterson (2014, First Edition). *Learning Responsive Web Design: A Beginner's Guide*, Chapter 1: What is Responsive Design?

## 12. Anexo I: Casos de uso extendidos

### 12.1. Cualquiera

<i><b>Registrarse</b></i>
<b>Descripción:</b> Acción que permitirá a un usuario anónimo inscribirse y quedar registrado en el sistema.
<b>Precondiciones:</b> Ninguna.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b> 1.- El usuario selecciona “Sign Up” en la página principal (Figura 49). 2.- Se rellenan los campos mostrados y se pulsa de nuevo en “Sign Up” (Figura 50).  <i>Si el nombre de usuario ya existe</i> 3a.- Se muestra un error por pantalla  <i>Si el nombre de usuario tiene menos de 15 años o más de 100</i> 3b.- Se muestra un error por pantalla
<b>Poscondiciones:</b> Si los datos son correctos se habrá guardado un usuario en la base de datos.

Tabla 75: Caso de uso extendido “Registrarse”

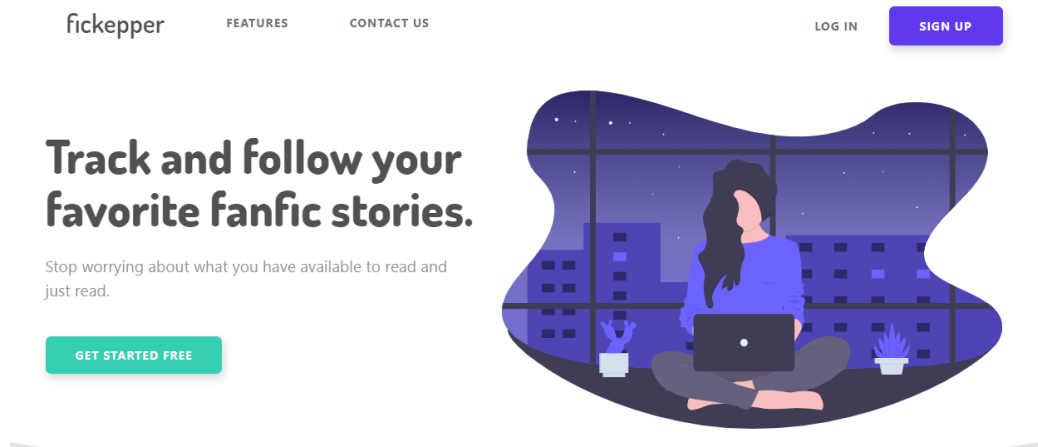


Figura 49: Botón “registrarse”

Password: \*

Your password can't be too similar to your other personal information.  
Your password must contain at least 8 characters.  
Your password can't be a commonly used password.  
Your password can't be entirely numeric.

Password confirmation: \*

Enter the same password as before, for verification.

[SIGN UP](#)

Figura 50: Varios campos del formulario de registro

<b><i>Identificarse</i></b>
<b>Descripción:</b> Método por el cual un usuario anónimo adquiere una identidad propia y distinguible en la aplicación.
<b>Precondiciones:</b> Ninguna.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b> 1.- El usuario selecciona “Log In” en la página principal (Figura 51). 2.- Se rellenan los campos usuario y contraseña, y se pulsa de nuevo en “Log In” (Figura 52).  <i>Si los datos no son correctos</i> 3a.- Aparece un mensaje de error por pantalla. <b>Poscondiciones:</b> Si los datos son correctos el usuario estará identificado en el sistema.

Tabla 76: Caso de uso extendido “Identificarse”

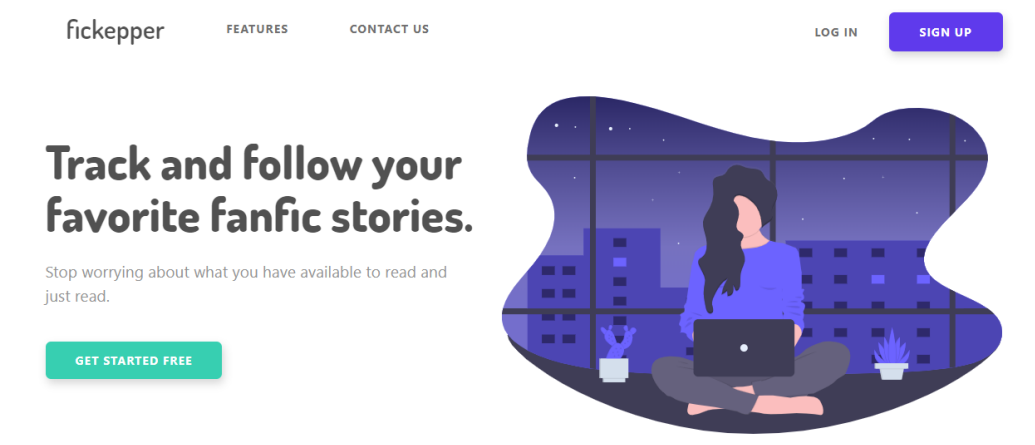


Figura 51: Botón “iniciar sesión”

**Log In**  
WELCOME BACK!

Username:

Password:

**LOG IN**

Figura 52: Formulario de inicio de sesión



<b><i>Recuperar contraseña</i></b>
<b>Descripción:</b> Mediante esta opción, cualquier usuario ya registrado en el sistema que haya olvidado su contraseña podrá recuperarla.
<b>Precondiciones:</b> Ninguna.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b> 1a.- El usuario selecciona “Log In” en la página principal (Figura 49). 1b.- El usuario selecciona “Sign Up” en la página principal (Figura 49). 2.- Se selecciona “Forgot Password?”. 3.- Se introduce el email y se pincha en “Send me instructions” (Figura 53).
<b>Poscondiciones:</b> Se ha enviado un email con instrucciones para recuperar la contraseña.

Tabla 77: Caso de uso extendido “Recuperar contraseña”

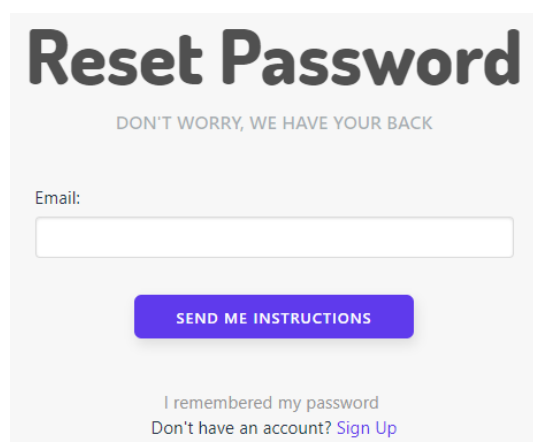


Figura 53: Pantalla de recuperar contraseña

## 12.2. Bibliotecario

<b><i>Ejecutar funciones administrativas</i></b>
<b>Descripción:</b> Esta funcionalidad englobará aquellas acciones que tengan que ver con la administración de la aplicación web.
<b>Precondiciones:</b> Ninguna.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b> 1.- El usuario selecciona “Admin panel”. (Figura 54). 2.- Se accede al panel de administración (Figura 55).  <i>Si el usuario quiere gestionar los capítulos</i> 3a.- Ver Tabla 79.  <i>Si el usuario quiere gestionar los personajes</i> 3b.- Ver Tabla 80.  <i>Si el usuario quiere gestionar los fandoms</i> 3c.- Ver Tabla 82.  <i>Si el usuario quiere gestionar los fanfics</i> 3d.- Ver Tabla 83.  <i>Si el usuario quiere gestionar las reseñas</i> 3e.- Ver Tabla 84.  <i>Si el usuario quiere gestionar los errores enviados por los usuarios</i> 3f.- Ver Tabla 85.
<b>Poscondiciones:</b> Los objetos del sistema modificados se han actualizado si no ha habido ningún error.

Tabla 78: Caso de uso extendido “Ejecutar funciones administrativas”

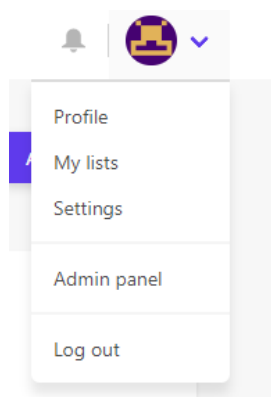


Figura 54: Opciones de los usuarios

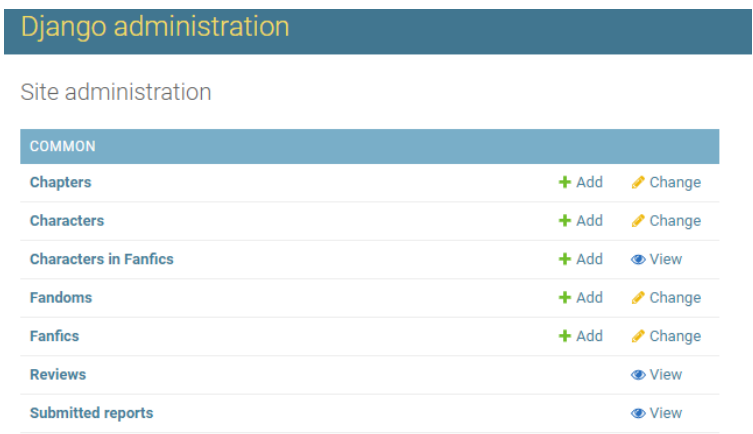


Figura 55: Panel de administración

<b><i>Gestionar capítulos</i></b>
<b>Descripción:</b> Aquellas acciones que permitan la gestión de los capítulos de los <i>fanfics</i> .
<b>Precondiciones:</b> Existe al menos un <i>fanfic</i> .
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b>
1.- El usuario selecciona “Chapters”. (Figura 56).
<i>Si el usuario quiere añadir un capítulo a un fanfic del sistema</i>
2a.- Presiona en “Add Chapter”.
3a.- Introduce los datos (Figura 57).
<i>Si el número del capítulo no es correcto o está repetido</i>
4aa.- Se muestra un mensaje de error por pantalla.
<i>Si el usuario quiere borrar un capítulo</i>
2b.- Se seleccionan los capítulos a borrar y se elige en “Action” la opción correspondiente.
3b.- Se confirma presionando “Yes, I’m sure”.
<i>Si el usuario quiere modificar un capítulo</i>
2c.- Se presiona el título del capítulo
3c.- Después de realizar los cambios se presiona en “Save”.
<i>Si el número del capítulo no es correcto o está repetido</i>
4ca.- Se muestra un mensaje de error por pantalla.
<b>Poscondiciones:</b> Los objetos del sistema modificados se han actualizado si no ha habido ningún error.

Tabla 79: Caso de uso extendido “Gestionar capítulos”

## Django administration

Site administration

COMMON		
Chapters	<a href="#">+ Add</a>	<a href="#">Change</a>
Characters	<a href="#">+ Add</a>	<a href="#">Change</a>
Characters in Fanfics	<a href="#">+ Add</a>	<a href="#">View</a>
Fandoms	<a href="#">+ Add</a>	<a href="#">Change</a>
Fanfics	<a href="#">+ Add</a>	<a href="#">Change</a>
Reviews		<a href="#">View</a>
Submitted reports		<a href="#">View</a>

Figura 56: Panel de administración

Add chapter

Title:

Num chapter:

Url chapter:

Fanfic:

[Save and add another](#) [Save and continue editing](#) [SAVE](#)

Figura 57: Añadir capítulo

<b><i>Gestionar personajes</i></b>
<b>Descripción:</b> Aquellas acciones que permitan la gestión de los personajes pertenecientes a un <i>fandom</i> .
<b>Precondiciones:</b> Existe al menos un <i>fandom</i> .
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b>
<i>Si el usuario quiere modificar los personajes de un fanfic</i>
1a.- El usuario selecciona “Characters in Fanfics”. (Figura 58).
<i>Si el usuario quiere asignar un personaje a un fanfic del sistema</i>
2aa.- Presiona en “Add Characters in Fanfics”. Selecciona el <i>fanfic</i> y el personaje. (Figura 59).
<i>Si el fanfic ya tiene asignado este personaje</i>
3aaa.- Se muestra un mensaje de error por pantalla.
<i>Si el usuario quiere borrar un personaje de un fanfic</i>
2ab.- Se seleccionan los personajes a borrar y se elige en “Action” la opción correspondiente. Se confirma presionando “Yes, I’m sure”.
<i>Si el usuario quiere crear un pairing entre dos personajes</i>
2ac.- Se seleccionan dos personajes y se elige en “Action” la opción correspondiente (“Create a pairing between the characters”).
<i>Si los personajes son de distinto fanfic o el pairing ya existe</i>
3aca.- Se muestra un mensaje de error por pantalla.
<i>Si el usuario quiere borrar un pairing</i>
2ad.- Se seleccionan dos personajes y se elige en “Action” la opción correspondiente (“Delete the pairing”).
<i>Si los personajes no pertenecen al mismo pairing</i>
3ada.- Se muestra un mensaje de error por pantalla.

Tabla 80: Caso de uso extendido “Gestionar personajes” (I)

## ***Gestionar personajes (II)***

*Sino*

1b.- El usuario selecciona “Characters”. (Figura 58).

*Si el usuario quiere añadir un personaje a un fandom del sistema*

2ba.- Presiona en “Add Character”.

3ba.- Introduce el nombre y apellido (Figura 60).

*Si el fandom tiene un personaje con el mismo nombre*

4baa.- Se muestra un mensaje de error por pantalla.

*Si el usuario quiere borrar un personaje*

2bb.- Se seleccionan los personajes a borrar y se elige en “Action” la opción correspondiente.

3bb.- Se confirma presionando “Yes, I’m sure”.

*Si el usuario quiere modificar un personaje*

2bc.- Se presiona el nombre del personaje

3bc.- Después de realizar los cambios se presiona en “Save”.

*Si el fandom tiene un personaje con el mismo nombre*

4bca.- Se muestra un mensaje de error por pantalla.

*Si el usuario quiere juntar dos personajes porque son el mismo*

2bd.- Se seleccionan dos usuarios y en “Action” se elige la opción “Merge the characters”.

*Si los personajes son de distinto fandom*

2bda.- Se muestra un mensaje de error por pantalla.

**Poscondiciones:** Los objetos del sistema modificados se han actualizado si no ha habido ningún error.

Tabla 81: Caso de uso extendido “Gestionar personajes” (II)

## Django administration

Site administration

COMMON		
Chapters	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Characters	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Characters in Fanfics	<a href="#">+ Add</a>	<a href="#">👁 View</a>
Fandoms	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Fanfics	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Reviews		<a href="#">👁 View</a>
Submitted reports		<a href="#">👁 View</a>

Figura 58: Panel de administración

Add Characters in Fanfics

Fanfic:

Character:

[Save and add another](#) [Save and view](#) [SAVE](#)

Figura 59: Añadir personaje a un *fanfic*



Add character

Name surname:

Fandom:

Save and add another

Save and continue editing

SAVE

Figura 60: Añadir personaje

<b><i>Gestionar fandoms</i></b>
<b>Descripción:</b> Aquellas acciones que permitan la gestión de los <i>fandoms</i> , ya sea cambiar el nombre, eliminarlos, etc.
<b>Precondiciones:</b> Los tipos (tv, películas,..) están definidos.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b>
1.- El usuario selecciona “Fandoms”. (Figura 61).
<i>Si el usuario quiere añadir un fandom</i>
2a.- Presiona en “Add Fandom”.
3a.- Introduce los datos (Figura 62).
<i>Si el fandom en ese tipo (tv, películas,..) ya existe</i>
4aa.- Se muestra un mensaje de error por pantalla.
<i>Si el usuario quiere borrar un fandom</i>
2b.- Se seleccionan los <i>fandoms</i> a borrar y se elige en “Action” la opción correspondiente.
3b.- Se confirma presionando “Yes, I’m sure”.
<i>Si el usuario quiere modificar un fandom</i>
2c.- Se presiona el título del <i>fandom</i> .
3c.- Después de realizar los cambios se presiona en “Save”.
<i>Si el fandom en ese tipo (tv, películas,..) ya existe</i>
4ca.- Se muestra un mensaje de error por pantalla.
<b>Poscondiciones:</b> Los objetos del sistema modificados se han actualizado si no ha habido ningún error.

Tabla 82: Caso de uso extendido “Gestionar *fandoms*”

## Django administration

Site administration

COMMON		
Chapters	<a href="#">+ Add</a>	<a href="#">Change</a>
Characters	<a href="#">+ Add</a>	<a href="#">Change</a>
Characters in Fanfics	<a href="#">+ Add</a>	<a href="#">View</a>
Fandoms	<a href="#">+ Add</a>	<a href="#">Change</a>
Fanfics	<a href="#">+ Add</a>	<a href="#">Change</a>
Reviews		<a href="#">View</a>
Submitted reports		<a href="#">View</a>

Figura 61: Panel de administración

Add fandom

Name:

Type:

Avatar:  Ningún archivo seleccionado

Figura 62: Añadir *fandom*

<b><i>Gestionar fanfics</i></b>
<b>Descripción:</b> Aquellas acciones que permitan la gestión de los <i>fanfics</i> del sistema.
<b>Precondiciones:</b> Existe al menos un <i>fandom</i> .
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b>
1.- El usuario selecciona “Fanfics”. (Figura 63).
<i>Si el usuario quiere añadir un fanfic</i>
2a.- Presiona en “Add Fanfic”.
3a.- Introduce los datos (Figura 64).
<i>Si alguno de los datos no es válido (web repetida, etc.)</i>
4aa.- Se muestra un mensaje de error por pantalla.
<i>Si el usuario quiere borrar un fanfic</i>
2b.- Se seleccionan los <i>fanfics</i> a borrar y se elige en “Action” la opción correspondiente.
3b.- Se confirma presionando “Yes, I’m sure”.
<i>Si el usuario quiere modificar un fanfic</i>
2c.- Se presiona el título del <i>fanfic</i> .
3c.- Después de realizar los cambios se presiona en “Save”.
<i>Si alguno de los datos no es válido (web repetida, etc.)</i>
4ca.- Se muestra un mensaje de error por pantalla.

Tabla 83: Caso de uso extendido “Gestionar *fanfics*”

## Django administration

Site administration

COMMON		
Chapters	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Characters	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Characters in Fanfics	<a href="#">+ Add</a>	<a href="#">👁 View</a>
Fandoms	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Fanfics	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Reviews		<a href="#">👁 View</a>
Submitted reports		<a href="#">👁 View</a>

Figura 63: Panel de administración

Add fanfic

**Name:**

**Author:**

**Web:**

Language:

Complete

Genre1:

Genre2:

Genre3:

Genre4:

Figura 64: Añadir *fanfic*

<b><i>Administrar reseñas</i></b>
<b>Descripción:</b> Aquellas acciones que permitan la gestión de las opiniones de los usuarios respecto de los <i>fanfics</i> .
<b>Precondiciones:</b> Ninguna.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b>
1.- El usuario selecciona “Reviews”. (Figura 65).
<i>Si el usuario quiere borrar una reseña</i>
2a.- Se seleccionan las reseñas a borrar y se elige en “Action” la opción correspondiente.
3a.- Se confirma presionando “Yes, I’m sure”.

Tabla 84: Caso de uso extendido “Administrar reseñas”

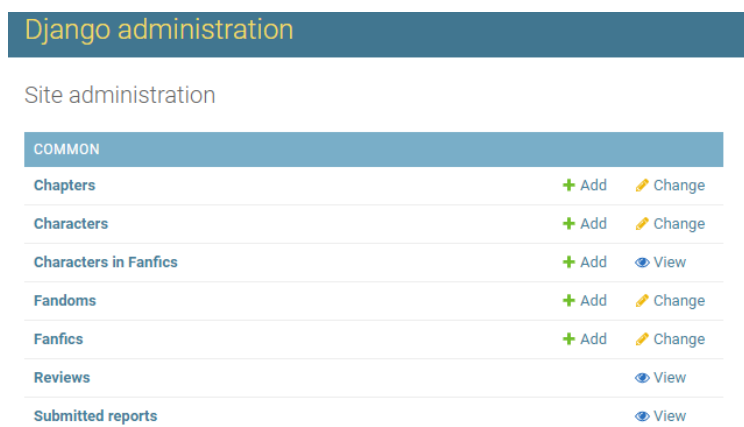


Figura 65: Panel de administración

<i>Gestionar errores</i>
<b>Descripción:</b> Aquellas acciones que permitan la gestión de errores enviados por el usuario desde la página de un <i>fanfic</i> .
<b>Precondiciones:</b> Ninguna.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b>
1.- El usuario selecciona “Submitted reports”. (Figura 66).
<i>Si el usuario quiere borrar uno de los errores enviados</i>
2a.- Se seleccionan los errores a borrar y se elige en “Action” la opción correspondiente.
3a.- Se confirma presionando “Yes, I’m sure”.

Tabla 85: Caso de uso extendido “Gestionar errores”

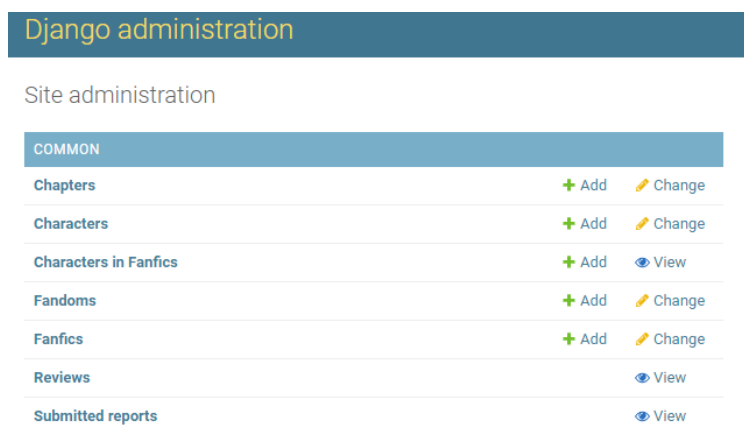


Figura 66: Panel de administración

### 12.3. Administrador

<i>Ver estadísticas e información de Ciencia de Datos</i>
<b>Descripción:</b> La aplicación contará con información de las interacciones del usuario con el sistema, ya sea el número de veces que una historia ha sido añadida, o el número de <i>fanfics</i> añadidos a un mismo <i>fandom</i> . Esta información en un principio no dice nada, pero tras aplicar técnicas de Ciencia de Datos se puede extraer conocimiento.
<b>Precondiciones:</b> Ninguna.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b> 1.- El usuario selecciona “Analytics” (Figura 67).  <i>Si no hay datos suficientes</i> 2a.- Se muestra un mensaje de error por pantalla
<b>Poscondiciones:</b> Si hay datos suficientes los gráficos se han creado y guardado.

Tabla 86: Caso de uso extendido “Ver estadísticas e información de Ciencia de Datos”

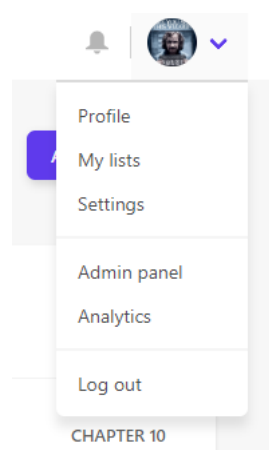


Figura 67: Opciones de los usuarios



## 12.4. Usuario identificado

<i>Cerrar sesión</i>
<b>Descripción:</b> Se dejará de estar identificado en el sistema y se pasará a ser un usuario anónimo.
<b>Precondiciones:</b> Ninguna.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b> 1.- El usuario selecciona “Log out” (Figura 68).
<b>Poscondiciones:</b> El usuario ya no estará identificado en el sistema.

Tabla 87: Caso de uso extendido “Cerrar sesión”

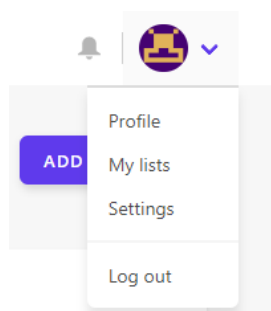


Figura 68: Opciones de los usuarios

<b><i>Gestionar listas</i></b>
<b>Descripción:</b> Se podrán crear, modificar y eliminar las listas que el usuario utilizará para organizar sus <i>fanfics</i> .
<b>Precondiciones:</b> Ninguna.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b>
1.- El usuario selecciona “My lists” (Figura 69).
<i>Si el usuario quiere añadir una lista</i>
2a.- Escribe el nombre de la lista a añadir.
<i>Si el usuario quiere añadir más de una lista a la vez</i>
3aa.- Presiona “Add” y escribe el nombre de la segunda lista.
4a.- El usuario presiona “Save” (Figura 70).
<i>Si existe una lista con el mismo nombre</i>
5aa.- Se muestra un mensaje de error al usuario.
<i>Si el usuario quiere borrar una lista</i>
2b.- Las listas que se quieran borrar se seleccionan, y se presiona el botón “Delete”(Figura 71).
<i>Si el usuario quiere modificar una lista</i>
2c.- Se cambian los nombres y se presiona el botón “Save”(Figura 72).
<i>Si algún nombre está repetido</i>
3ca.- Se muestra un mensaje de error al usuario.
<b>Poscondiciones:</b> Las listas se han actualizado si no ha habido ningún error.

Tabla 88: Caso de uso extendido “Gestionar listas”

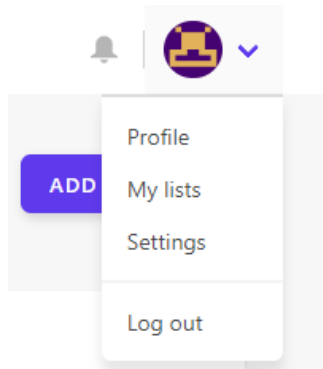


Figura 69: Opciones de los usuarios

  
[Add](#)  
[DELETE](#) [DISCARD](#) [SAVE](#)

Figura 70: Formulario para añadir una nueva lista

   
  
[Add](#)  
[DELETE](#) [DISCARD](#) [SAVE](#)

Figura 71: Borrar listas

■

Add

DELETE                      DISCARD                      **SAVE**

Figura 72: Cambiar el nombre de listas

<b><i>Leer notificaciones</i></b>
<b>Descripción:</b> Acción de leer los avisos recibidos por parte de la aplicación.
<b>Precondiciones:</b> Ninguna.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b> 1.- El usuario pulsa en el icono correspondiente (Figura 73), y se abre un contenedor con las últimas notificaciones.  <i>Si el usuario quiere marcar una notificación como leída</i> 2a.- Hace click en el círculo de la Figura 74.  <i>Si el usuario quiere marcar todas las notificaciones como leídas</i> 2b.- Hace click en “Mark all as read” (Figura 74).  <i>Si el usuario quiere ver todas las notificaciones</i> 2c.- Hace click en “See all” (Figura 74). <i>Si el usuario quiere marcar una notificación como leída</i> 3ca.- Hace click en el círculo de la Figura 75.  <i>Si el usuario quiere marcar todas las notificaciones como leídas</i> 3cb.- Hace click en “Mark all as read” (Figura 75).  <i>Si se produce un fallo de conexión</i> 4.- Se muestra un mensaje de error al usuario.
<b>Poscondiciones:</b> Ya no aparecerá el número de notificaciones por ver en el icono de las notificaciones.

Tabla 89: Caso de uso extendido “Leer notificaciones”

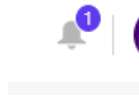


Figura 73: Icono notificaciones

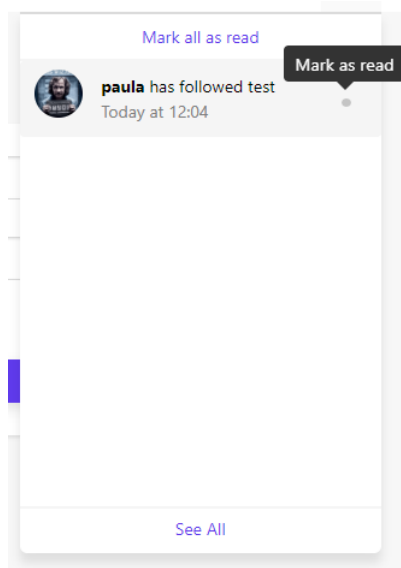


Figura 74: Marcar como leído

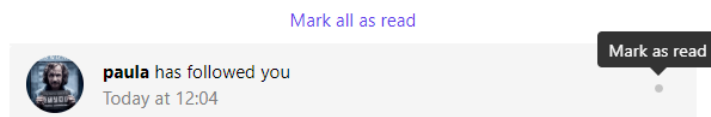


Figura 75: Marcar como leído

<b><i>Ver perfil propio</i></b>
<b>Descripción:</b> El usuario tendrá la posibilidad de ver su propio perfil.
<b>Precondiciones:</b> Ninguna.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b> 1.- El usuario selecciona “Profile” (Figura 76).  <i>Si pulsa sobre “See all stories”</i> 2a. Ver Tabla 91  <i>Si tiene fanfics favoritos y pulsa sobre uno</i> 2b. Ver Tabla 103  <i>Si pulsa sobre “Follows”</i> 2c. Ver Tabla 95  <i>Si pulsa sobre “Followers”</i> 2d. Ver Tabla 96
<b>Poscondiciones:</b> Ninguna.

Tabla 90: Caso de uso extendido “Ver perfil propio”

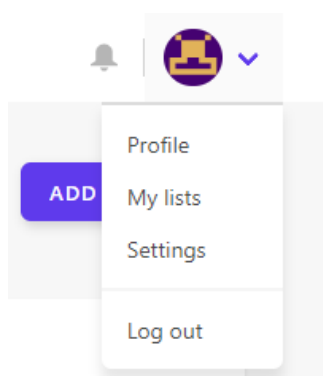


Figura 76: Opciones de los usuarios

<b><i>Ver fanfics de un usuario</i></b>
<b>Descripción:</b> Ver los fanfics de un usuario.
<b>Precondiciones:</b> Se tiene el nivel de privacidad requerido por dicho usuario.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b> 1.- El usuario selecciona “See all stories” (Figura 76). 2.- Se seleccionan el <i>fandom</i> , la lista y los filtros a utilizar.  <i>Si tiene fanfics y pulsa sobre “Mark as read”</i> 3a.- Ver Tabla 108  <i>Si se produce un fallo de conexión</i> 4aa.- Se muestra un mensaje de error al usuario.  <i>Si tiene fanfics y pulsa sobre el título de un fanfic</i> 3b.- Ver Tabla 103  <i>Si quiere añadir, cambiar o borrar el fanfic de una de sus listas</i> 3c.- Ver Tabla 107
<b>Poscondiciones:</b> Ninguna.

Tabla 91: Caso de uso extendido “Ver fanfics de un usuario”

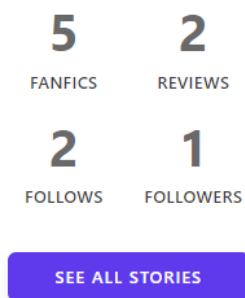


Figura 77: Panel lateral del perfil de un usuario



<b><i>Cambiar la configuración de la cuenta</i></b>
<b>Descripción:</b> Los ajustes de la cuenta podrán ser cambiados por los usuarios.
<b>Precondiciones:</b> Ninguna.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b>
1.- El usuario selecciona “Settings” (Figura 78).
<i>Si quiere cambiar la información de su perfil</i>
2a.- El usuario presiona “Profile” en el menú lateral izquierdo, y cambia los datos que crea oportunos.
<i>Si quiere cambiar la imagen de perfil</i>
3aa.- Selecciona una foto (.jpg, .jpeg, .gif o .png).
4a.- Presiona “Save” para guardar los cambios.
<i>Si alguno de los datos no es correcto</i>
5aa.- Se muestra un mensaje de error.
<i>Si quiere cambiar la configuración general</i>
2b.- Se presiona en “General” en el menú lateral izquierdo.
<i>Si se quiere crear una copia de seguridad</i>
3ba.- Ver Tabla 93
4b.- Cambia la configuración que desee y pulsa en “Save” (Figura 79).
<b>Poscondiciones:</b> Los datos del usuario se han actualizado si han sido correctos.

Tabla 92: Caso de uso extendido “Cambiar la configuración de la cuenta”

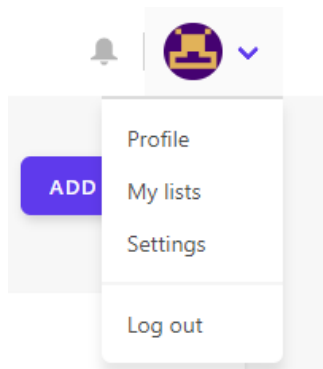


Figura 78: Opciones de los usuarios

### User Settings

- PROFILE
- GENERAL

**PROFILE**

Username: \*

Full Name: \*

Email: \*

New password:

MAX. 1MB




Figura 79: Configuración de la cuenta del usuario

<b><i>Exportar copia de seguridad</i></b>
<b>Descripción:</b> Posibilidad de crear una copia de seguridad de la información del usuario almacenada hasta el momento. Esta funcionalidad otorga una sensación de control al usuario, ya que puede respaldar sus datos cuando quiera.
<b>Precondiciones:</b> Ninguna.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b> 1.- El usuario selecciona “Create backup of my data” (Figura 80).
<b>Poscondiciones:</b> Se ha guardado un fichero <i>.json</i> con los datos del usuario.

Tabla 93: Caso de uso extendido “Exportar copia de seguridad”

GENERAL

Create backup of my data

Figura 80: Crear copia de seguridad

<b><i>Ver perfil usuario</i></b>
<b>Descripción:</b> Se tendrá acceso al perfil del resto de usuarios del sistema.
<b>Precondiciones:</b> Hay más de un usuario en el sistema.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b> 1.- El usuario pincha en el usuario que quiera ver (Figura 81).  <i>Si pulsa sobre "See all stories"</i> 2a.- Ver Tabla 91  <i>Si tiene fanfics favoritos y pulsa sobre uno</i> 2b.- Ver Tabla 103  <i>Si pulsa sobre "Follows"</i> 2c.- Ver Tabla 95  <i>Si pulsa sobre "Followers"</i> 2d.- Ver Tabla 96  <i>Si pulsa sobre "Follow/Unfollow"</i> 2e.- Ver Tabla 97
<b>Poscondiciones:</b> Ninguna

Tabla 94: Caso de uso extendido "Ver perfil usuario"

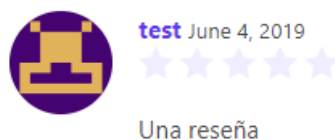


Figura 81: Reseña de un miembro de la aplicación

<b><i>Ver usuarios seguidos</i></b>
<b>Descripción:</b> Ver los usuarios que otro usuario sigue.
<b>Precondiciones:</b> Ninguna.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b> 1.- El usuario pincha en “Follows” (Figura 82).
<b>Poscondiciones:</b> Ninguna

Tabla 95: Caso de uso extendido “Ver usuarios seguidos”

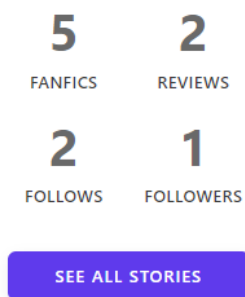


Figura 82: Panel lateral del perfil de un usuario

<b>Ver seguidores</b>
<b>Descripción:</b> Ver los usuarios que siguen a otro.
<b>Precondiciones:</b> Ninguna.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b> 1.- El usuario pincha en “Followers” (Figura 83).
<b>Poscondiciones:</b> Ninguna

Tabla 96: Caso de uso extendido “Ver seguidores”

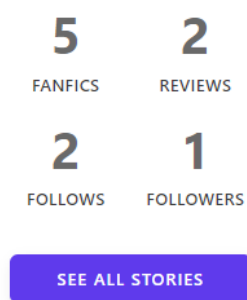


Figura 83: Panel lateral del perfil de un usuario

<b><i>Gestionar seguimiento</i></b>
<b>Descripción:</b> Seguir o dejar de seguir a un usuario.
<b>Precondiciones:</b> Hay más de un usuario en el sistema, y el usuario a seguir/dejar de seguir no es el mismo.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b> <i>Si no se está siguiendo y se quiere seguir a un usuario</i> 1a. - El usuario pincha en “Follow” (Figura 84).  <i>Si se está siguiendo y se quiere dejar de seguir a un usuario</i> 1b. - El usuario pincha en “Unfollow” (Figura 85).  <i>Si se produce un error</i> 2.- Se muestra un mensaje de error por pantalla.
<b>Poscondiciones:</b> El estado de la relación entre dos usuarios ha cambiado si no ha habido ningún error.

Tabla 97: Caso de uso extendido “Gestionar seguimiento”



Figura 84: Botón seguir a un usuario



Figura 85: Botón dejar de seguir a un usuario

<b><i>Enviar error</i></b>
<b>Descripción:</b> Avisar de un error en la información de un <i>fanfic</i> del sistema.
<b>Precondiciones:</b> Existe al menos un <i>fanfic</i> .
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b> 1.- El usuario presiona el botón “Report error” (Figura 86). 2.- El usuario rellena el formulario y presiona “Send” (Figura 87).
<b>Poscondiciones:</b> El error ha quedado registrado.

Tabla 98: Caso de uso extendido “Enviar error”

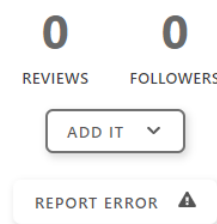


Figura 86: Botón para reportar error



Submit fanfic error ✕

What seems to be the issue?  
----- ▾

Please, tell us something more about it...

CANCEL SEND

Figura 87: Formulario para notificar de un error en el *fanfic*

<b><i>Explorar fanfics</i></b>
<b>Descripción:</b> El sistema dispondrá de un catálogo por categorías que se podrá explorar, buscar, filtrar y ordenar para localizar <i>fanfics</i> según gustos personales.
<b>Precondiciones:</b> Existen categorías para agrupar fandoms.
<b>Requisitos no funcionales:</b> Ninguno.
<p><b>Flujo de eventos:</b></p> <p>1.- El usuario pincha en “Browse” y selecciona una categoría. (Figura 88).</p> <p><i>Si quiere añadir un fanfic externo</i></p> <p>2a.- Ver Tabla 102</p> <p><i>Si no</i></p> <p>2b.- El usuario selecciona el <i>fandom</i> que quiera explorar (Figura 89).</p> <p>3b.- Se pincha en “Filter By” y se ajustan las opciones (Figura 90).</p> <p><i>Si pulsa sobre un fanfic</i></p> <p>4ba.- Ver Tabla 103</p> <p><i>Si quiere añadir, cambiar o borrar el fanfic de una de sus listas</i></p> <p>4bb.- Ver Tabla 107</p>
<b>Poscondiciones:</b> Ninguna

Tabla 99: Caso de uso extendido “Explorar fanfics”

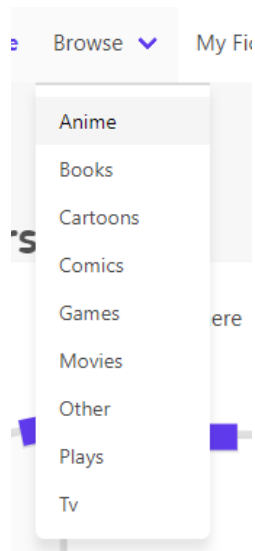


Figura 88: Categorías *fanfics*

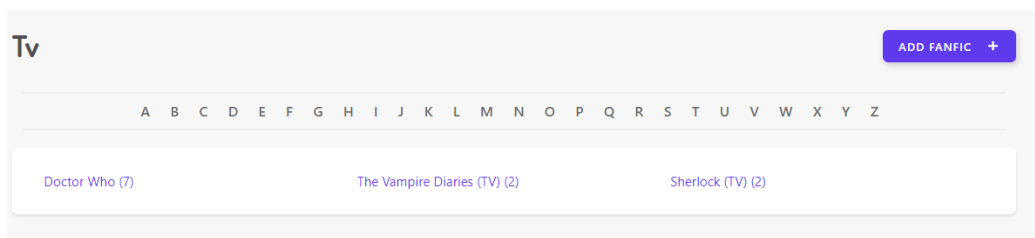


Figura 89: *Fandoms* de una categoría

### Filter fanfics ✕

Sort by: All <span>▼</span>	Genre: All <span>▼</span>
Language: All <span>▼</span>	Length: All <span>▼</span>
Status: All <span>▼</span>	Rating: All <span>▼</span>
Character a: All <span>▼</span>	Character b: All <span>▼</span>

CANCEL FILTER

Figura 90: Filtros para buscar *fanfics*

<b><i>Ver mis fanfics</i></b>
<b>Descripción:</b> Cada usuario dispondrá de una “biblioteca” compuesta de listas en las que estarán clasificados los <i>fanfics</i> . La agrupación principal se realizará mediante <i>fandoms</i> .
<b>Precondiciones:</b> Existen <i>fandoms</i> en el sistema.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b> 1.- El usuario pincha en “My Fics” (Figura 91). 2.- Se seleccionan el <i>fandom</i> , la lista y los filtros a utilizar (Figura 92).  <i>Si pulsa sobre un fanfic</i> 3a.- Ver Tabla 103  <i>Si quiere marcar el último capítulo como leído</i> 3b.- Ver Tabla 108  <i>Si quiere añadir, cambiar o borrar el fanfic de una de sus listas</i> 3c.- Ver Tabla 107  <i>Si quiere añadir un fanfic externo</i> 4.- Ver Tabla 102
<b>Poscondiciones:</b> Ninguna

Tabla 100: Caso de uso extendido “Ver mis fanfics”

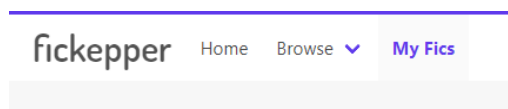



Figura 91: Parte del menú superior

My Fanfics > The Vampire Diaries (TV) ADD FANFIC +

Search fandom... SELECT LIST ▾ FILTER BY 

The Vampire Diaries (TV) |

**LETS CHEERS TO THIS**

★ ★ ★ ☆ ☆ 3.0

BY AFORGOTTENWEASLEY - T - ENGLISH - HUMOR / ROMANCE 0/29 CHAPTERS

LAST UPDATED: JUN 29, 2013 - LAST CHECKED: MAY 28, 2019 MARK AS READ

83391 WORDS TESTLIST ▾

[ Stefan Salvatore , OC ] - OC , Stefan Salvatore

Figura 92: Página mis fanfics

<b><i>Buscar fanfics y usuarios</i></b>
<b>Descripción:</b> Acción de buscar términos en la base de datos del sistema. El resultado de la búsqueda será un listado de <i>fanfics</i> o usuarios afines al término usado.
<b>Precondiciones:</b> Ninguna.
<b>Requisitos no funcionales:</b> Ninguno.
<p><b>Flujo de eventos:</b></p> <p>1.- El usuario escribe el término a buscar en el cuadro de búsqueda y presiona <i>Enter</i> o el botón de buscar (Figura 93).</p> <p><i>Si pulsa sobre un fanfic</i></p> <p>2a.- Ver Tabla 103</p> <p><i>Si quiere buscar un usuario</i></p> <p>3b.- Introduce el término de búsqueda y elige la opción correspondiente (Figura 94).</p> <p><i>Si pulsa sobre un usuario</i></p> <p>3ba.- Ver Tabla 94</p>
<b>Poscondiciones:</b> Ninguna

Tabla 101: Caso de uso extendido “Buscar fanfics”



Figura 93: Cuadro de búsqueda

término de búsqueda

Users  Fanfics

SEARCH

Figura 94: Opciones de búsqueda



<b><i>Añadir fanfic externo</i></b>
<b>Descripción:</b> Posibilidad de importar un <i>fanfic</i> a la aplicación.
<b>Precondiciones:</b> Ninguna.
<b>Requisitos no funcionales:</b> Internet.
<p><b>Flujo de eventos:</b></p> <p>1.- El usuario introduce la dirección web del <i>fanfic</i> a añadir y presiona “Continue” (Figura 95).</p> <p><i>Si el formato no es correcto, la web no está disponible o el fanfic ya está en el sistema</i></p> <p>2a.- Se le muestra un mensaje de error al usuario.</p> <p><b>Poscondiciones:</b> El <i>fanfic</i> se ha añadido a la cola para ser importado si no ha habido ningún error. Se notificará al usuario si se produce algún error obteniendo sus datos o se ha añadido correctamente.</p>

Tabla 102: Caso de uso extendido “Añadir fanfic externo”

**Add Fanfic**

URL \*

[See supported sites](#)

**CONTINUE**

Examples:

- <http://www.avengersfanfiction.com/Story/90426/lce-to-meet-you/>
- <https://archiveofourown.org/works/144365>
- <https://ficwad.com/story/267693>

Figura 95: Añadir *fanfic*

<b><i>Ver página de fanfic individual</i></b>
<b>Descripción:</b> Visualizar la página de un <i>fanfic</i> . Se dispondrá de los datos técnicos, la lista de capítulos y los comentarios de otros miembros registrados.
<b>Precondiciones:</b> Existe al menos un <i>fanfic</i> .
<b>Requisitos no funcionales:</b> Ninguno.
<p><b>Flujo de eventos:</b></p> <p>1.- El usuario hace click en el nombre de un <i>fanfic</i> (Figura 96).</p> <p><i>Si pulsa sobre un usuario en las reseñas</i></p> <p>2a.- Ver Tabla 94</p> <p><i>Si quiere marcar/desmarcar un capítulo como leído</i></p> <p>2b.- Ver Tabla 104</p> <p><i>Si quiere escribir, modificar o borrar una nota privada</i></p> <p>2c.- Ver Tabla 105</p> <p><i>Si quiere escribir, modificar o borrar su reseña</i></p> <p>2d.- Ver Tabla 106</p> <p><i>Si quiere añadir, cambiar o borrar el fanfic de una de sus listas</i></p> <p>2e.- Ver Tabla 107</p> <p><i>Si quiere reportar un error</i></p> <p>2f.- Ver Tabla 98</p>
<b>Poscondiciones:</b> Ninguna.

Tabla 103: Caso de uso extendido “Ver página de fanfic individual”

The Vampire Diaries (TV)  
**LETS CHEERS TO THIS**  
LAST UPDATED: JUN 29, 2013  
[Stefan Salvatore, OC] - OC , Stefan Salvatore

0/29 CHAPTERS

MARK AS READ

Figura 96: Un *fanfic*

<b><i>Gestionar capítulos leídos</i></b>
<b>Descripción:</b> Marcar/Desmarcar como leídos capítulos de un <i>fanfic</i> .
<b>Precondiciones:</b> Hay un <i>fanfic</i> con capítulos.
<b>Requisitos no funcionales:</b> Ninguno.
<p><b>Flujo de eventos:</b></p> <p><i>Si el usuario quiere marcar todos los capítulos como leídos</i></p> <p>1a.- El usuario selecciona “Mark all as read” (Figura 97).</p> <p><i>Sino</i></p> <p>1b.- El usuario hace click en el icono de cada capítulo para marcarlo como leído/no leído (Figura 97).</p> <p><i>Si se produce un error de conexión</i></p> <p>2ba.- Se muestra un mensaje de error.</p>
<b>Poscondiciones:</b> El estado de los capítulos se ha actualizado si no ha habido ningún error.

Tabla 104: Caso de uso extendido “Gestionar capítulos leídos”



<a href="#">CLEAR ALL</a>   <a href="#">CLEAR PRIVATE NOTES</a>   <a href="#">MARK ALL AS READ</a>		
CHAPTER	PRIVATE NOTES	READ
1 <a href="#">EDIT</a>		
2 <a href="#">EDIT</a>		

Figura 97: Gestionar capítulos de un *fanfic*

<b><i>Gestionar notas privadas</i></b>
<b>Descripción:</b> Añadir, modificar y eliminar notas privadas de capítulos.
<b>Precondiciones:</b> Hay un <i>fanfic</i> con capítulos.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b> <i>Si el usuario quiere borrar todas las notas privadas</i> 1a.- El usuario selecciona “Clear private notes” (Figura 98).  <i>Si el usuario quiere crear, editar o eliminar una nota privada</i> 1b.- El usuario hace click en “Edit”, escribe o edita la nota, y pulsa en “Save”. (Figura 98).  <i>Si se produce un error de conexión o el usuario ha superado los 200 caracteres</i> 2ba.- Se muestra un mensaje de error.
<b>Poscondiciones:</b> Las notas se han actualizado.

Tabla 105: Caso de uso extendido “Gestionar notas privadas”

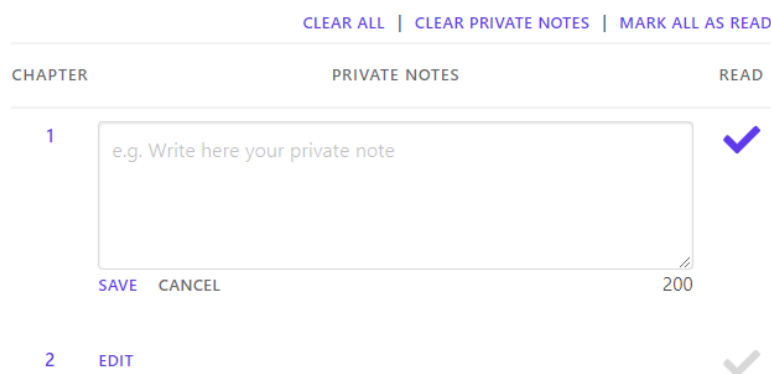
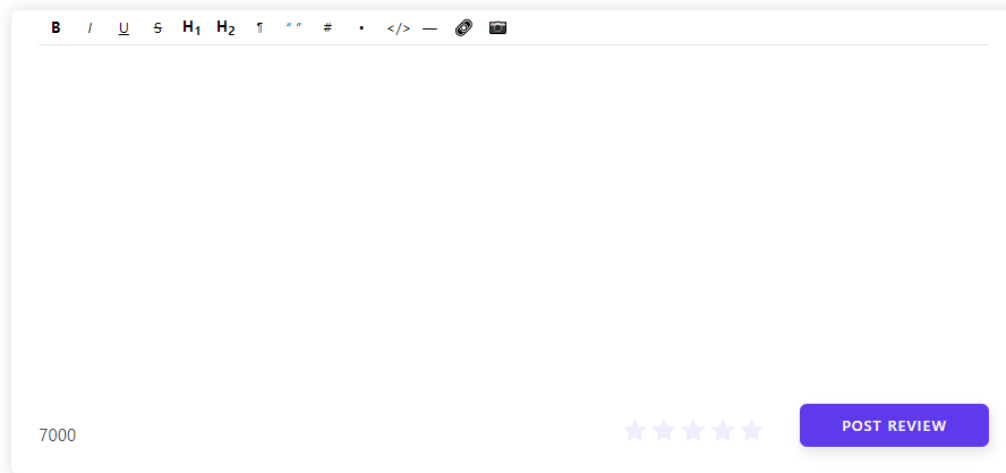




Figura 98: Notas privadas en los capítulos

<b><i>Gestionar reseñas</i></b>
<b>Descripción:</b> Añadir, modificar y eliminar reseñas de los <i>fanfics</i> .
<b>Precondiciones:</b> Hay un <i>fanfic</i> .
<b>Requisitos no funcionales:</b> Ninguno.
<p><b>Flujo de eventos:</b></p> <p><i>Si el usuario todavía no ha escrito una reseña en el fanfic y quiere</i></p> <p>1a.- El usuario escribe la reseña y pulsa en “Post Review” (Figura 99).</p> <p><i>Si la reseña está vacía o ha superado los 7000 caracteres</i></p> <p>2aa.- Se le muestra un mensaje de error al usuario.</p> <p><i>Si el usuario ha escrito una reseña y quiere borrarla</i></p> <p>1b.- El usuario presiona en el icono de borrar. (Figura 100).</p> <p><i>Si el usuario quiere editar la reseña</i></p> <p>1c.- El usuario presiona en el icono de modificar, edita la reseña y pulsa en “Save”. (Figura 100).</p> <p><i>Si la reseña está vacía o ha superado los 7000 caracteres</i></p> <p>2ca.- Se le muestra un mensaje de error al usuario.</p>
<b>Poscondiciones:</b> La reseña se ha actualizado.

Tabla 106: Caso de uso extendido “Gestionar reseñas”

## Reviews (1)



B / U S H<sub>1</sub> H<sub>2</sub> ¶ "" # • </> —  

7000

☆☆☆☆☆ [POST REVIEW](#)

Figura 99: Formulario nueva reseña



Figura 100: Una reseña

<b><i>Gestionar fanfic en lista</i></b>
<b>Descripción:</b> Añadir y/o eliminar un <i>fanfic</i> de una lista.
<b>Precondiciones:</b> El usuario tiene alguna lista.
<b>Requisitos no funcionales:</b> Ninguno.
<p><b>Flujo de eventos:</b></p> <p><i>Si el fanfic no está añadido y lo quiere añadir</i></p> <p>1a.- El usuario pincha en “Add it” y selecciona una lista (Figura 101).</p> <p><i>Si el fanfic está añadido y quiere cambiar la lista</i></p> <p>1b.- El usuario pincha en el botón correspondiente y selecciona una nueva lista (Figura 102).</p> <p><i>Si el fanfic está añadido y quiere borrarlo de la lista</i></p> <p>1c.- El usuario hace click en el icono correspondiente (Figura 102) y desmarca la lista actual.</p> <p><i>Si se produce un error de conexión</i></p> <p>2.- Se le muestra un mensaje de error al usuario.</p>
<b>Poscondiciones:</b> Los <i>fanfics</i> de las listas de los usuarios han sido actualizados.

Tabla 107: Caso de uso extendido “Gestionar *fanfic* en lista”

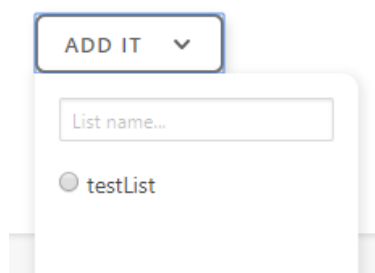


Figura 101: Añadir fanfic a una lista



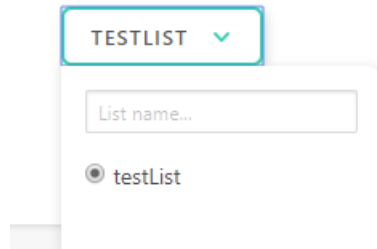


Figura 102: Cambiar fanfic de lista

<b><i>Marcar último capítulo de un fanfic como leído</i></b>
<b>Descripción:</b> El usuario puede marcar el último capítulo no leído de un <i>fanfic</i> como leído.
<b>Precondiciones:</b> Existe un <i>fanfic</i> en las listas del usuario del que no se han leído capítulos.
<b>Requisitos no funcionales:</b> Ninguno.
<b>Flujo de eventos:</b> 1.- El usuario hace click en “Mark as read” (Figura 103).  <i>Si se ha producido un error de conexión</i> 2.- Aparece un mensaje de error.
<b>Poscondiciones:</b> El capítulo se ha marcado como leído.

Tabla 108: Caso de uso extendido “Marcar último capítulo de un fanfic como leído”

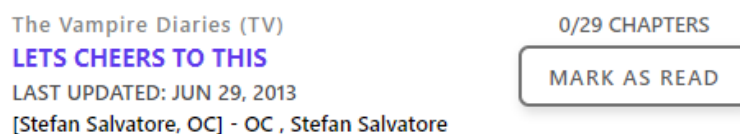


Figura 103: Un *fanfic*

### 13. Anexo II: Diagramas de secuencia

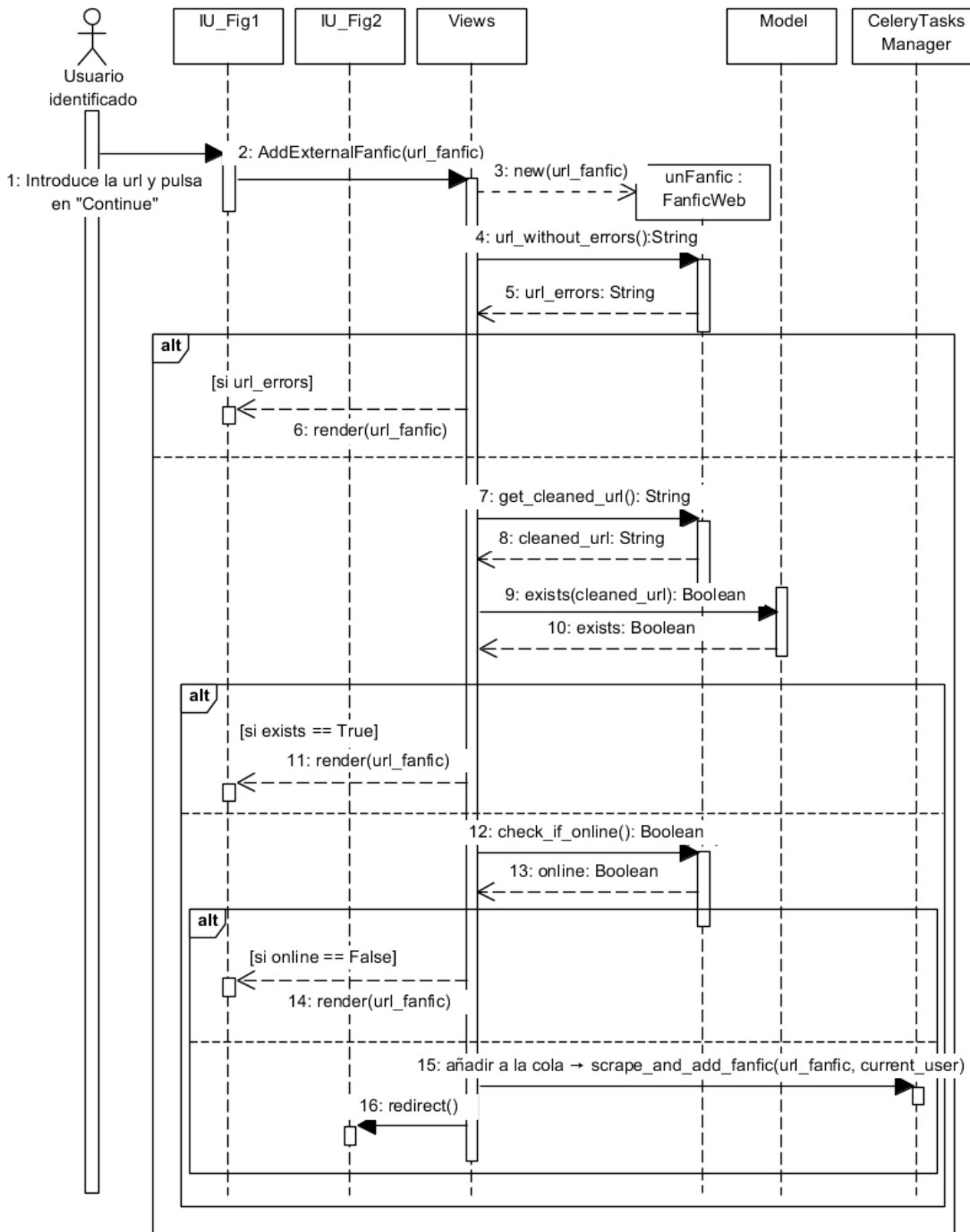


Figura 104: Añadir tarea para importar *fanfic*

## Add Fanfic

URL \*

[See supported sites](#)


[CONTINUE](#)

Examples:

- <http://www.avengersfanfiction.com/Story/90426/lce-to-meet-you/>
- <https://archiveofourown.org/works/144365>
- <https://ficwad.com/story/267693>

Figura 105: IU\_Fig1

fickepper Home Browse ▾ My Fics Search fanfics... 🔍 🔔 👤 ▾



### You did it!

We'll send you a notification as soon as we add it to the system.

[ADD MORE](#)

[Go Home](#)

Figura 106: IU\_Fig2

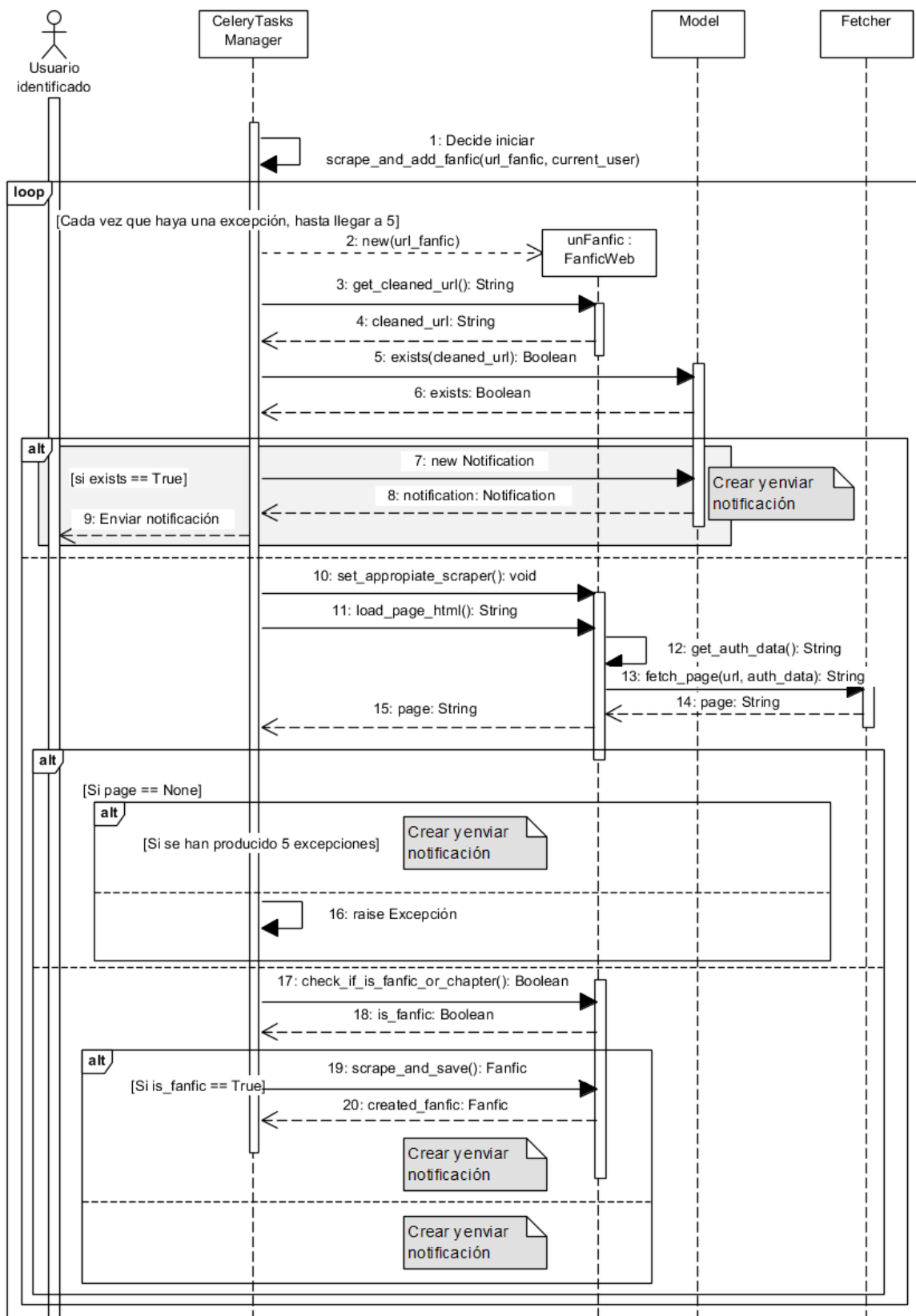


Figura 107: Iniciar importación de *fanfic* externo

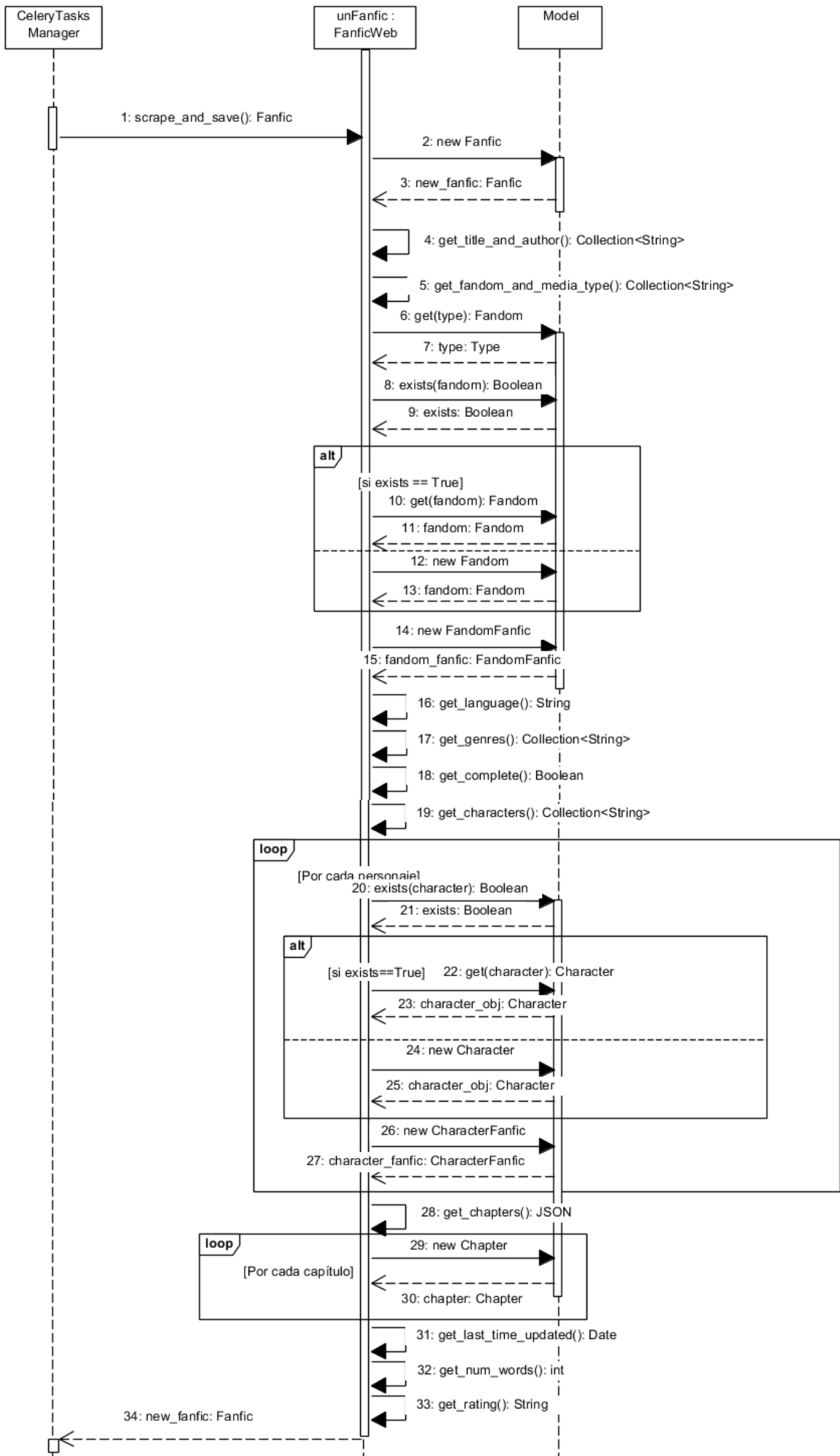


Figura 108: Subdiagrama scrapear y añadir fanfic externo

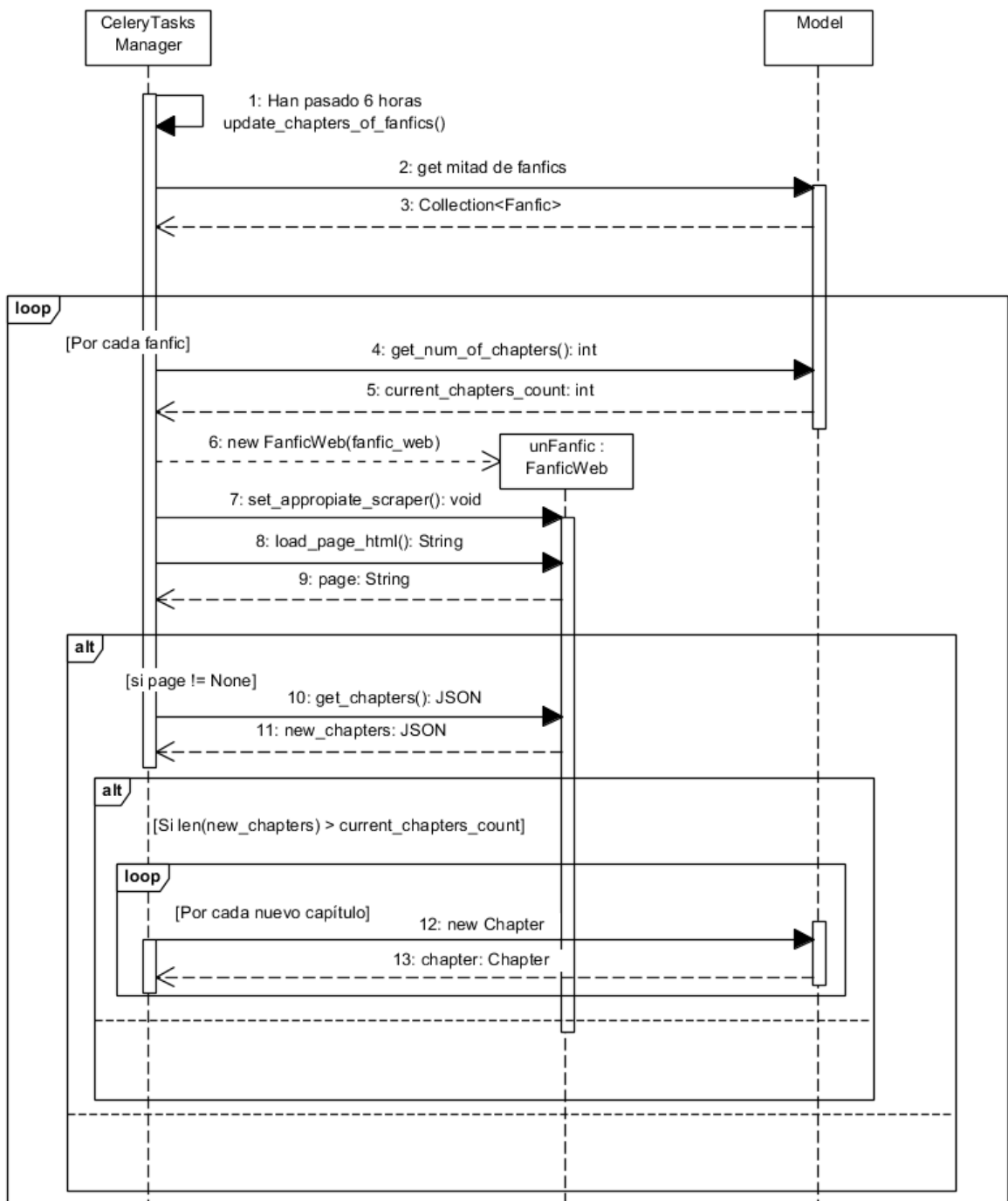


Figura 109: Comprobar si hay capítulos nuevos

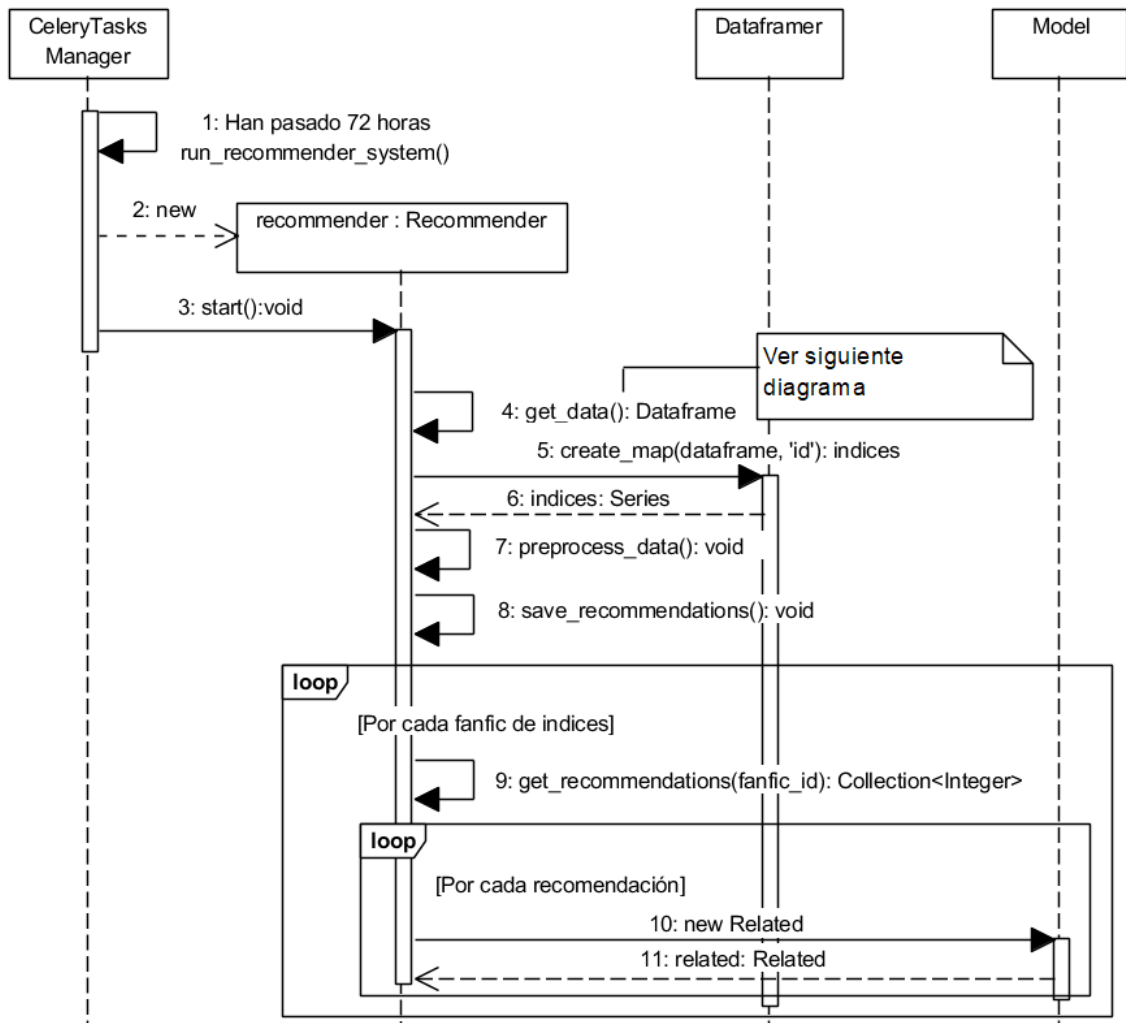


Figura 110: Entrenar sistema de recomendación de *fanfics*



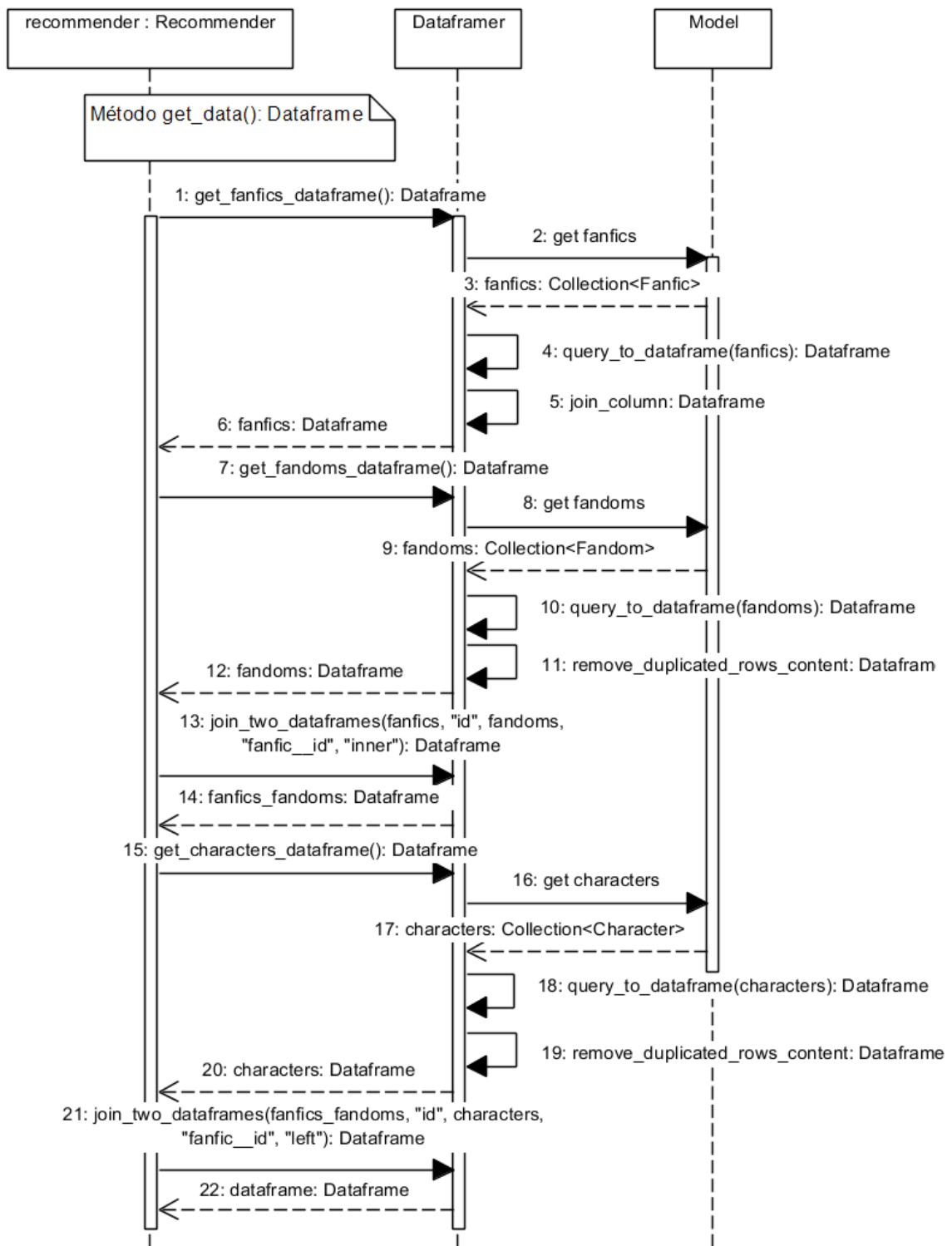


Figura 111: Subdiagrama de entrenar sistema de recomendación de *fanfics*

## 14. Anexo III: Prototipos baja fidelidad

Sign up

LOGO

JOIN FICKEEPER

FULL NAME

\_\_\_\_\_

USERNAME \_\_\_\_\_

EMAIL ADDRESS \_\_\_\_\_

GÉNERO FEMANAMUNIONA

PASSWORD \_\_\_\_\_ CONFIRM PASSWORD \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

(SIGN UP)

login

Gayciful

Figura 112: Prototipo de registrarse

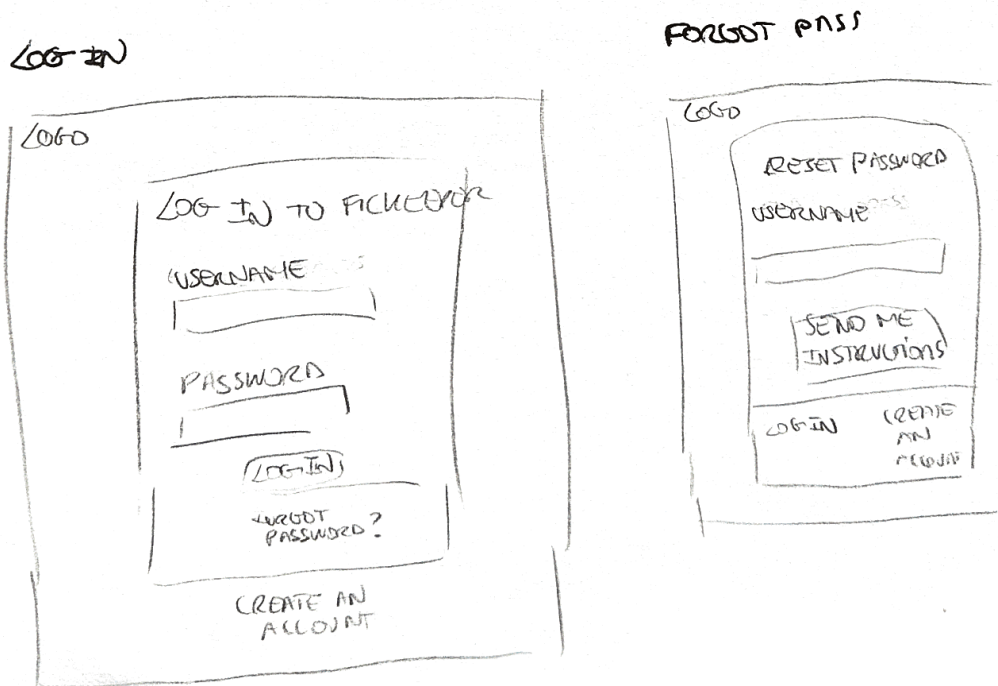


Figura 113: Prototipo de iniciar sesión y recuperar contraseña

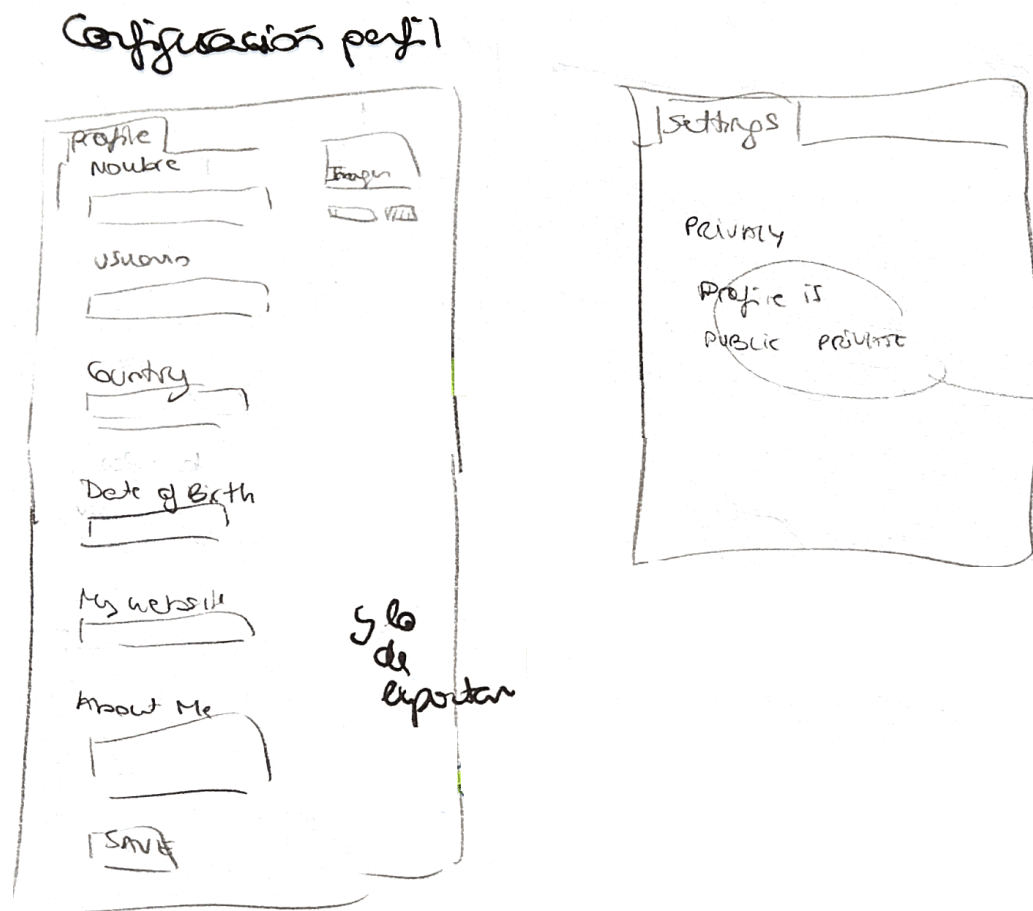


Figura 114: Prototipo de configuración del perfil

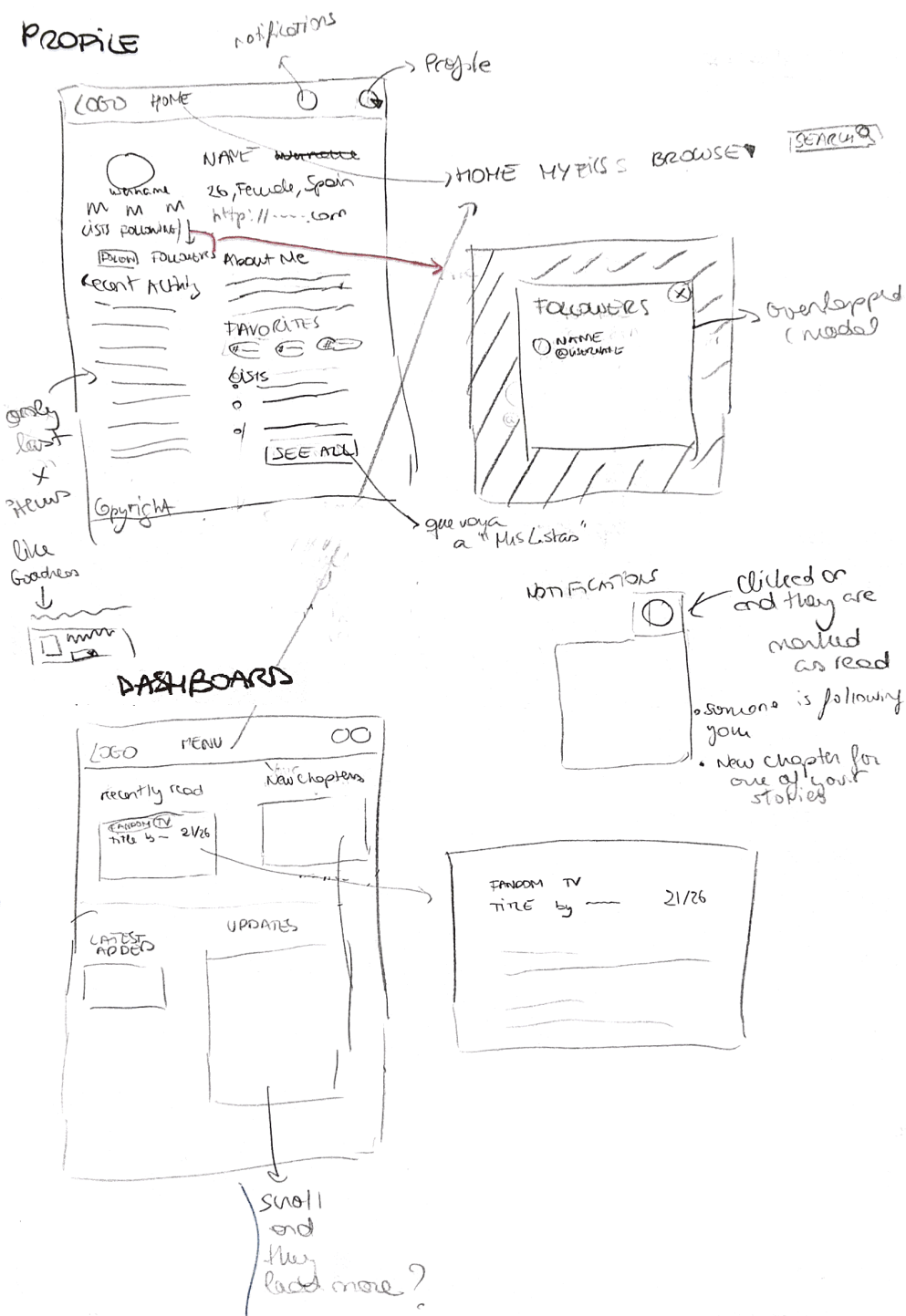


Figura 115: Prototipo de dashboard y perfil

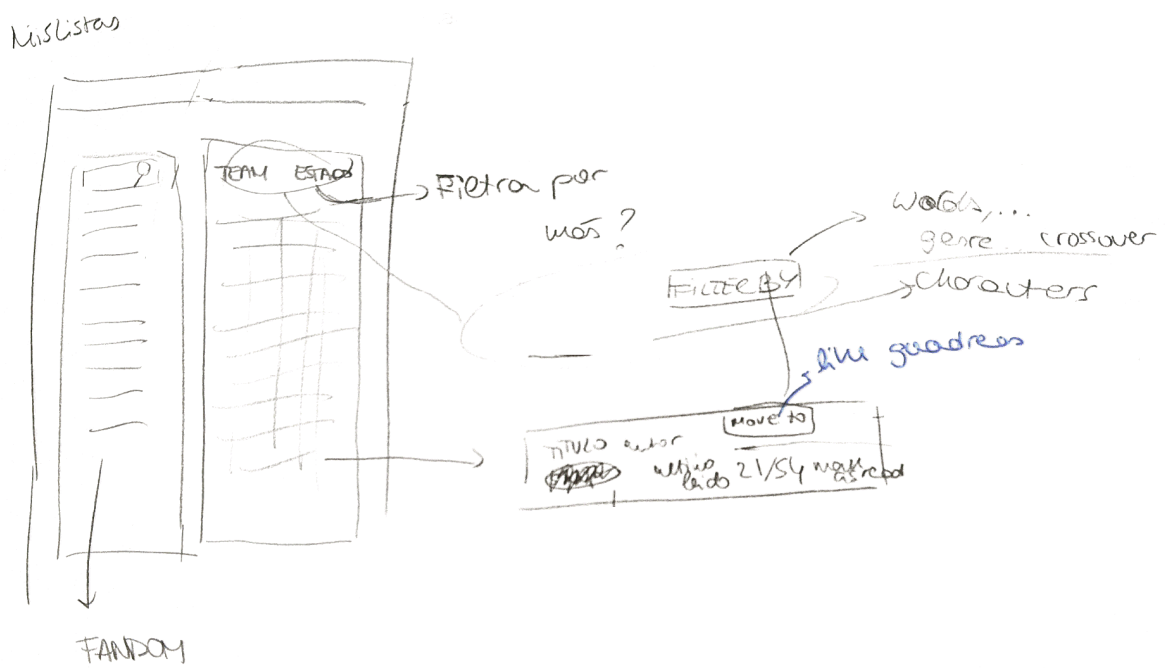
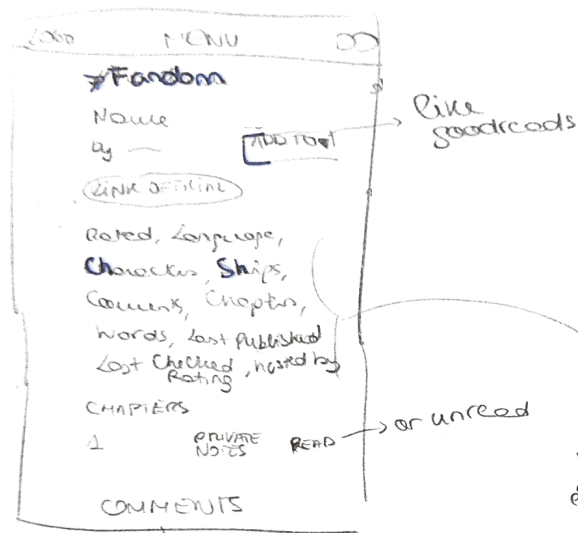


Figura 116: Prototipo de *fanfics* propios

## Página de fanfic individual



## Catálogo

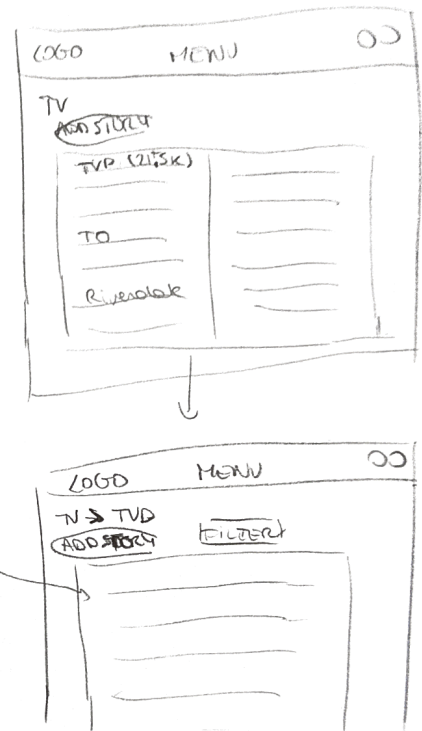


Figura 117: Prototipo de explorar *fanfics* y ver página de *fanfic*

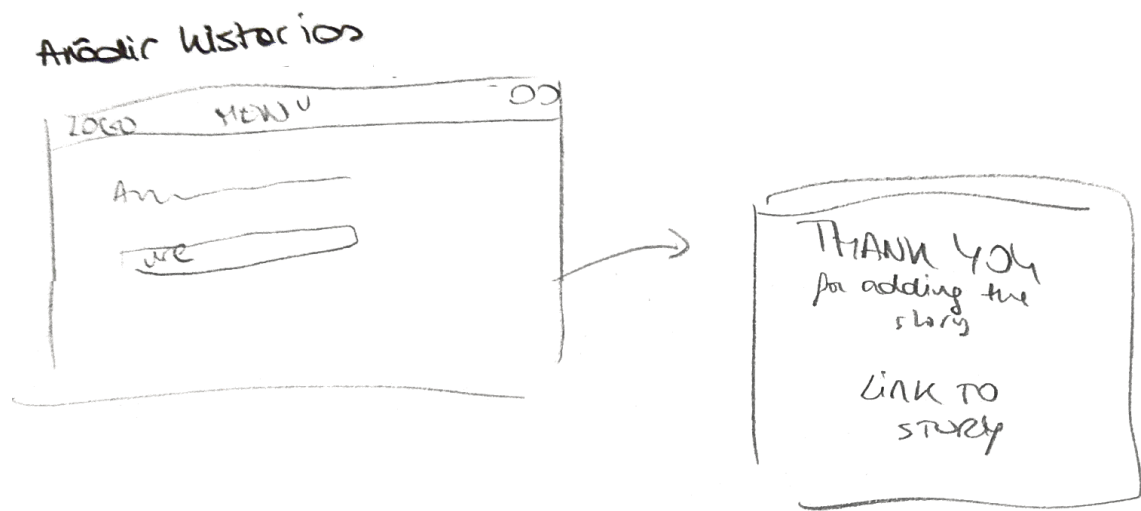


Figura 118: Prototipo de añadir *fanfic* externo



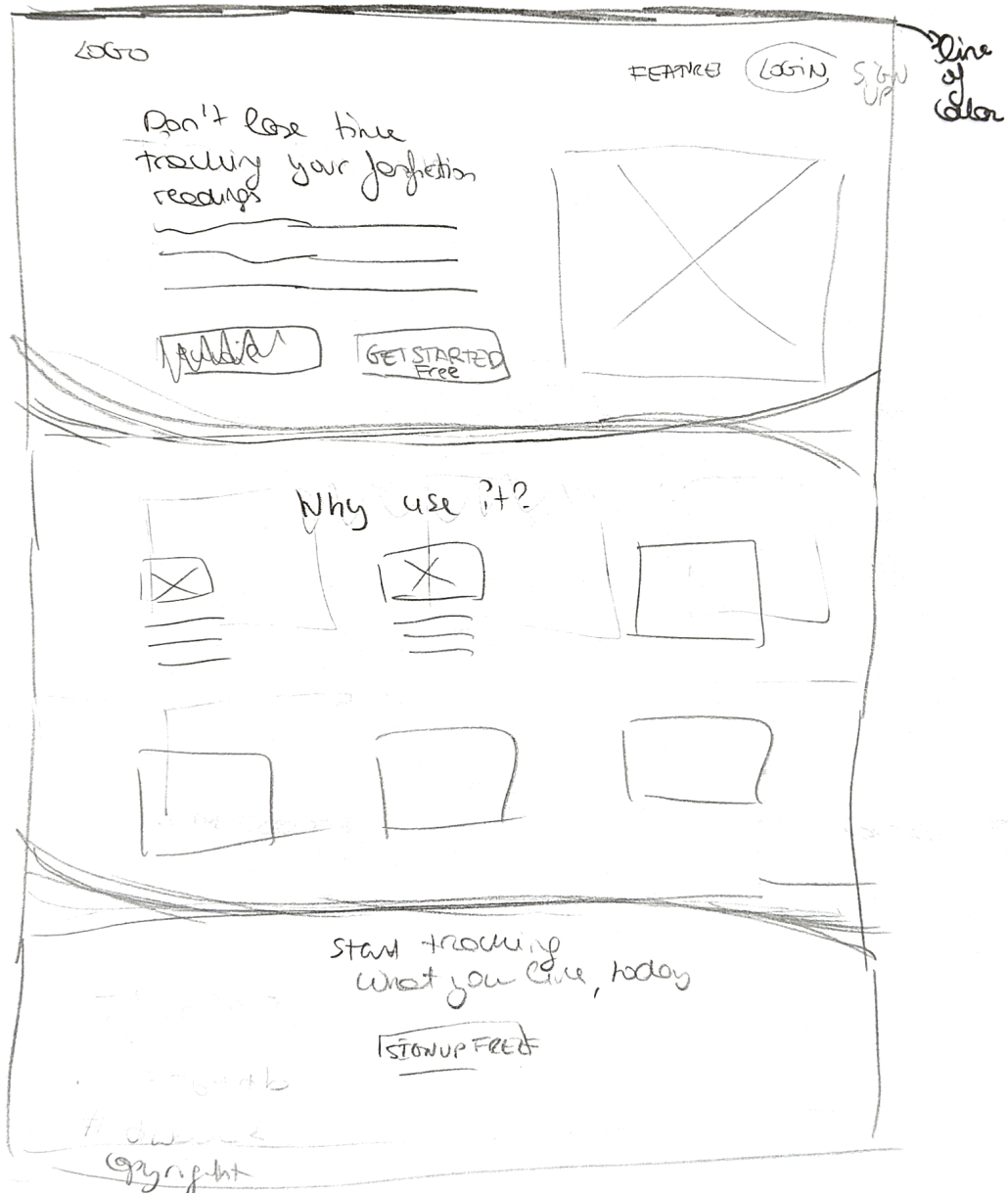


Figura 119: Prototipo de la página de aterrizaje

## 15. Anexo IV: Prototipos alta fidelidad

---

fickepper

# Log In

WELCOME BACK!

Username

Password

**LOG IN**

Don't have an account? [Sign up](#)  
Forgot Password?

© 2018 Copyright. Fickepper.

Figura 120: Prototipo de la identificación

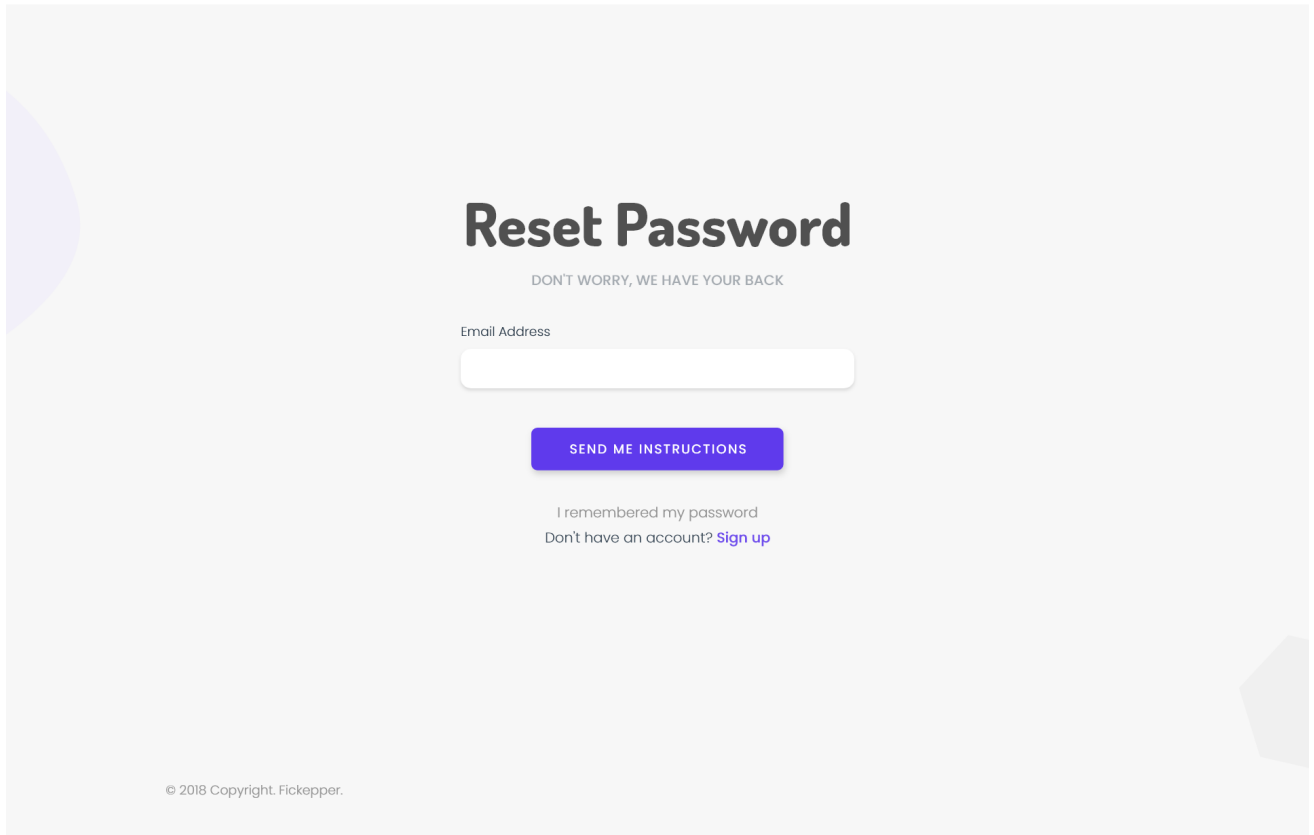


Figura 121: Prototipo de recuperar contraseña

# Sign Up

HI THERE! JOIN US HERE

Full Name\*

Username\*

✓ ✗

User Exists

Email Address\*

Country\*

Birthday\*

Password\* Confirm Password\*

**SIGN UP**

Already have an account? [Log in](#)

© 2018 Copyright. Fickepper.

Figura 122: Prototipo de registrarse

+ ADD FANFIC

### Last Chapters

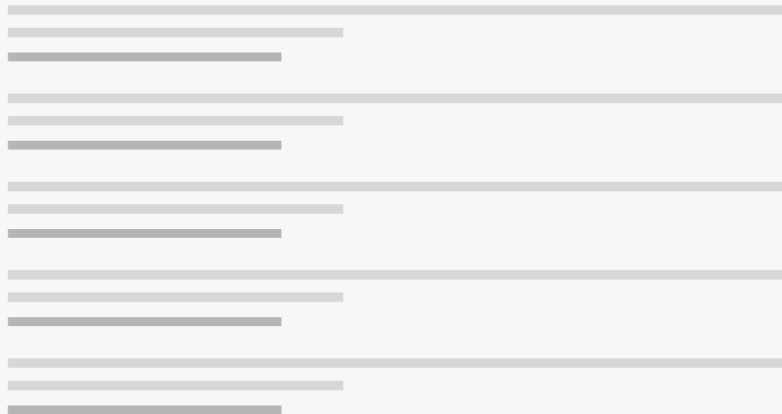
The Secret Circle <b>NOMBRE DE LA HISTORIA</b> LAST UPDATED: DEC 23 [Sirius Black, OC] - Sirius B., Severus S.	2/42 CHAPTERS MARK AS READ
The Secret Circle <b>NOMBRE DE LA HISTORIA</b> LAST UPDATED: DEC 23 [Sirius Black, OC] - Sirius B., Severus S.	2/42 CHAPTERS MARK AS READ
The Secret Circle <b>NOMBRE DE LA HISTORIA</b> LAST UPDATED: DEC 23 [Sirius Black, OC] - Sirius B., Severus S.	2/42 CHAPTERS MARK AS READ
The Secret Circle <b>NOMBRE DE LA HISTORIA</b> LAST UPDATED: DEC 23 [Sirius Black, OC] - Sirius B., Severus S.	2/42 CHAPTERS MARK AS READ

### Recently Read

The Secret Circle <b>NOMBRE DE LA HISTORIA</b> LAST UPDATED: DEC 23	CHAPTER 21
The Secret Circle <b>NOMBRE DE LA HISTORIA</b> LAST UPDATED: DEC 23	CHAPTER 21
The Secret Circle <b>NOMBRE DE LA HISTORIA</b> LAST UPDATED: DEC 23	CHAPTER 21
The Secret Circle <b>NOMBRE DE LA HISTORIA</b> LAST UPDATED: DEC 23	CHAPTER 21
The Secret Circle <b>NOMBRE DE LA HISTORIA</b> LAST UPDATED: DEC 23	CHAPTER 21
The Secret Circle <b>NOMBRE DE LA HISTORIA</b> LAST UPDATED: DEC 23	CHAPTER 21

### Recently Added

The Secret Circle (TV) <b>NOMBRE DE LA HISTORIA</b> BY AUTHOR
The Secret Circle (TV) <b>NOMBRE DE LA HISTORIA</b> BY AUTHOR
The Secret Circle (TV) <b>NOMBRE DE LA HISTORIA</b> BY AUTHOR
The Secret Circle (TV) <b>NOMBRE DE LA HISTORIA</b> BY AUTHOR



PAGINACIÓN

Figura 123: Prototipo del dashboard

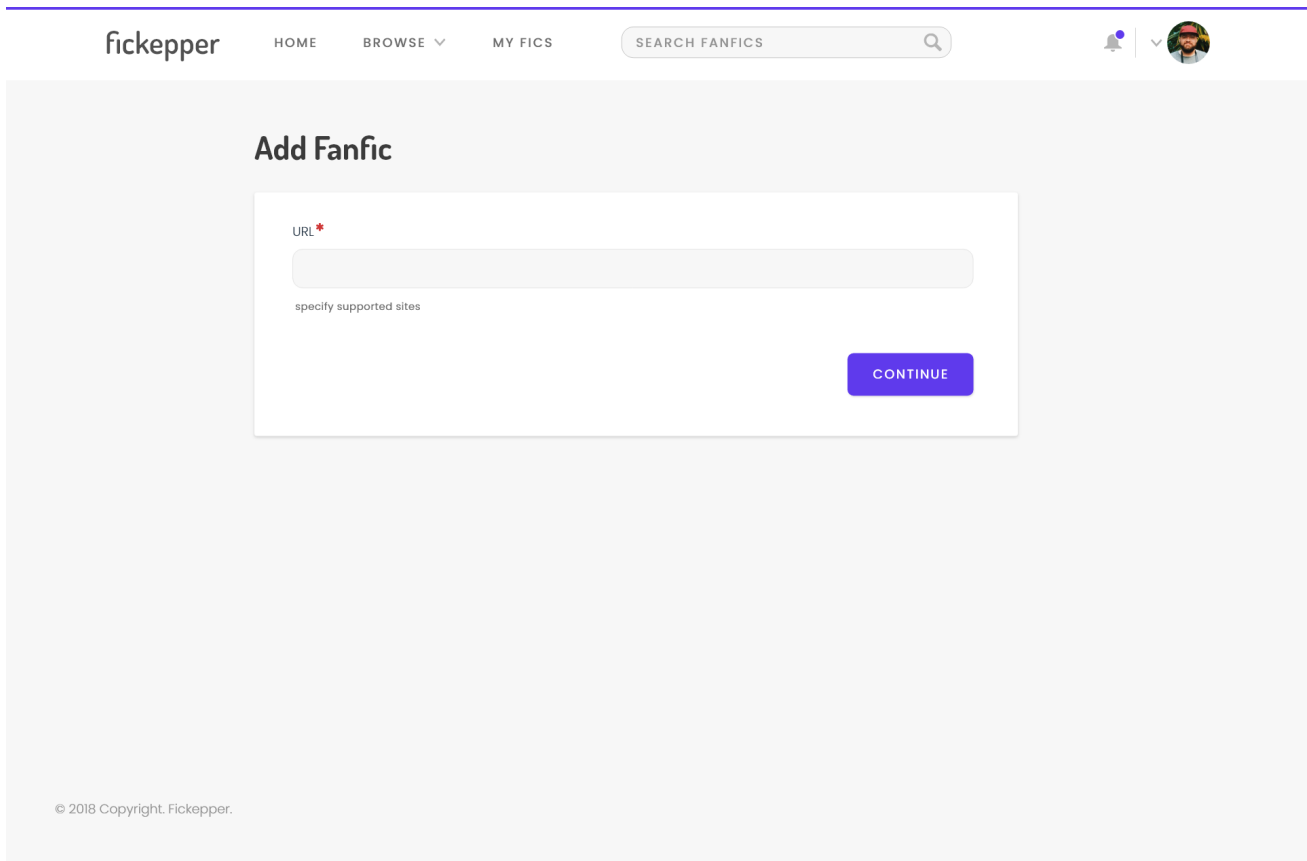


Figura 124: Prototipo de añadir *fanfics*

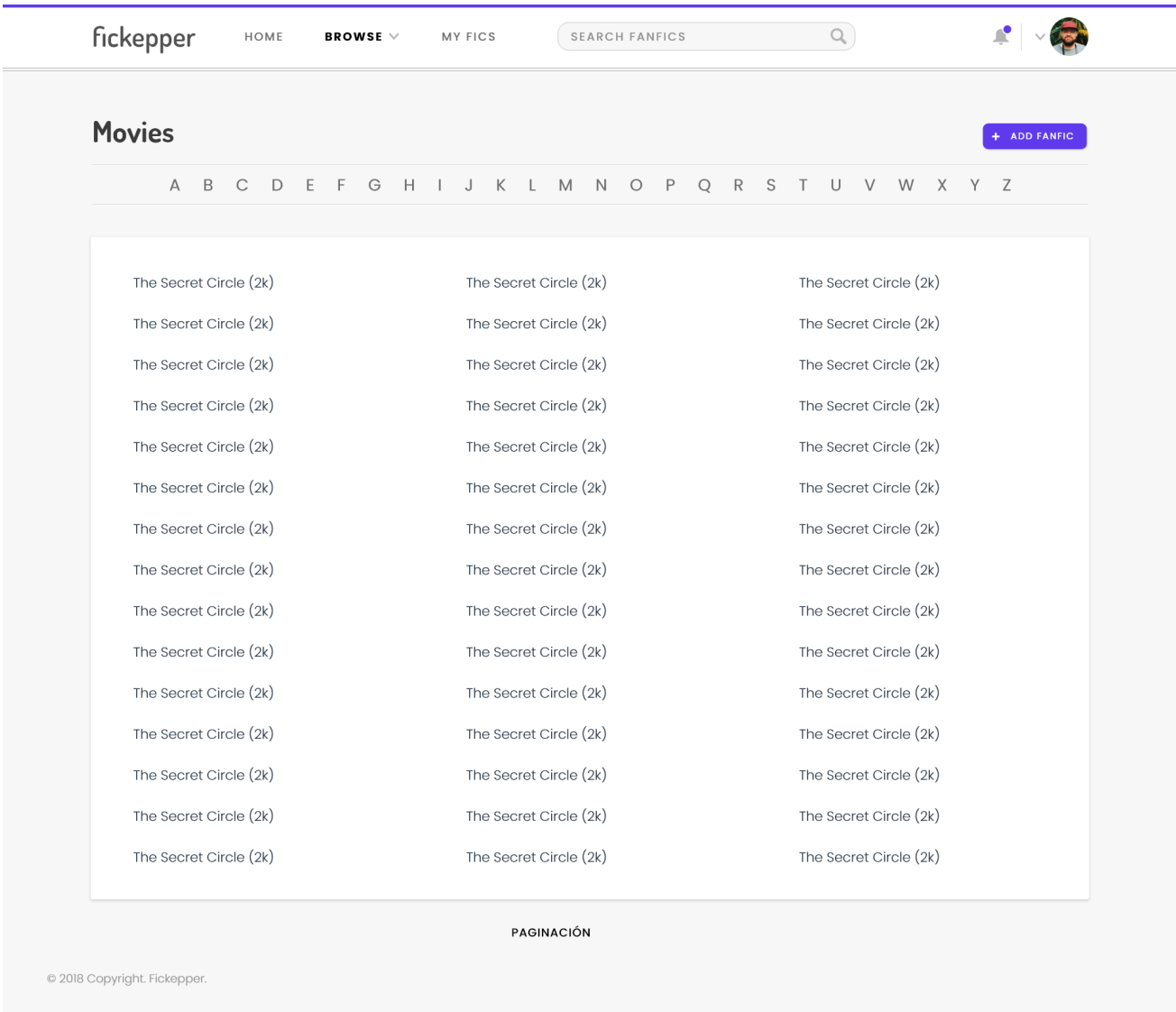


Figura 125: Prototipo de explorar *fanfics*

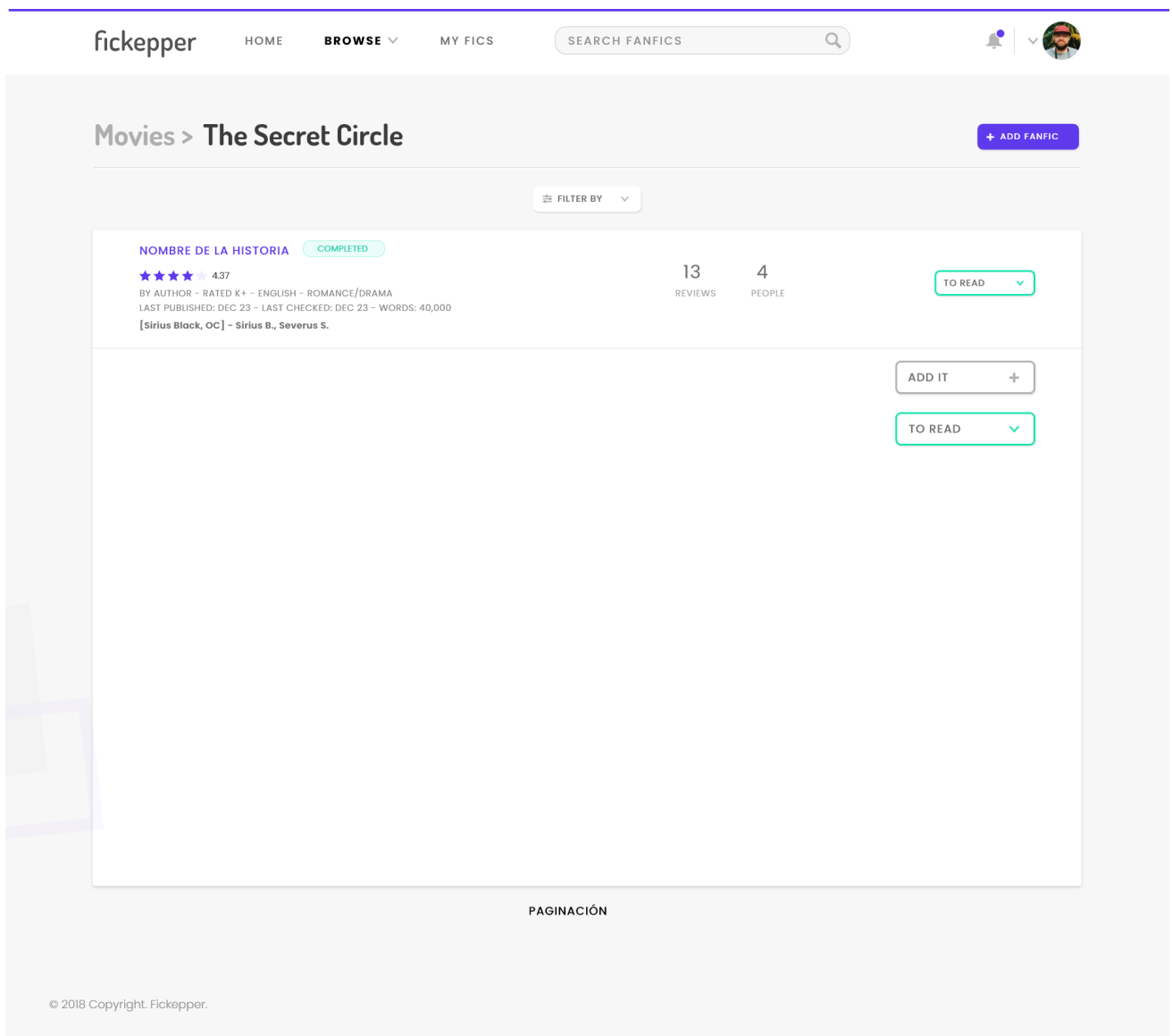


Figura 126: Prototipo de explorar *fanfics* de un *fandom*





## Movies &gt; The Secret Circle

Name of Fanfic COMPLETED

Name of Author

★★★★☆ 4.37

RATED K+ - ENGLISH - ROMANCE/DRAMA  
LAST PUBLISHED: DEC 23 - LAST CHECKED: DEC 23  
21 CHAPTERS - WORDS: 40,000

[Sirius Black, OC] - Sirius B., Severus S.

HOSTED BY

[http://your-web\\_\(Name\\_of\\_site\)](http://your-web_(Name_of_site))

YOU MAY ALSO LIKE

[Name of Fanfic \(fandom\)](#)[Name of Fanfic \(fandom\)](#)[Name of Fanfic \(fandom\)](#)

7

REVIEWS

5

FOLLOWERS

ADD IT

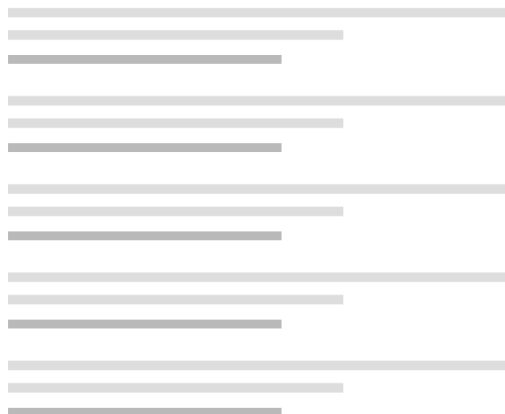


## Chapters

[CLEAR ALL](#) | [CLEAR PRIVATE NOTES](#) | [MARK ALL AS READ](#)

CHAPTER	PRIVATE NOTES	READ
1	Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea. <a href="#">EDIT</a>	✓
2	Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea. <a href="#">EDIT</a>	✓

## Reviews (1,523)



PAGINACIÓN

Figura 127: Prototipo de ver la página de un *fanfic*

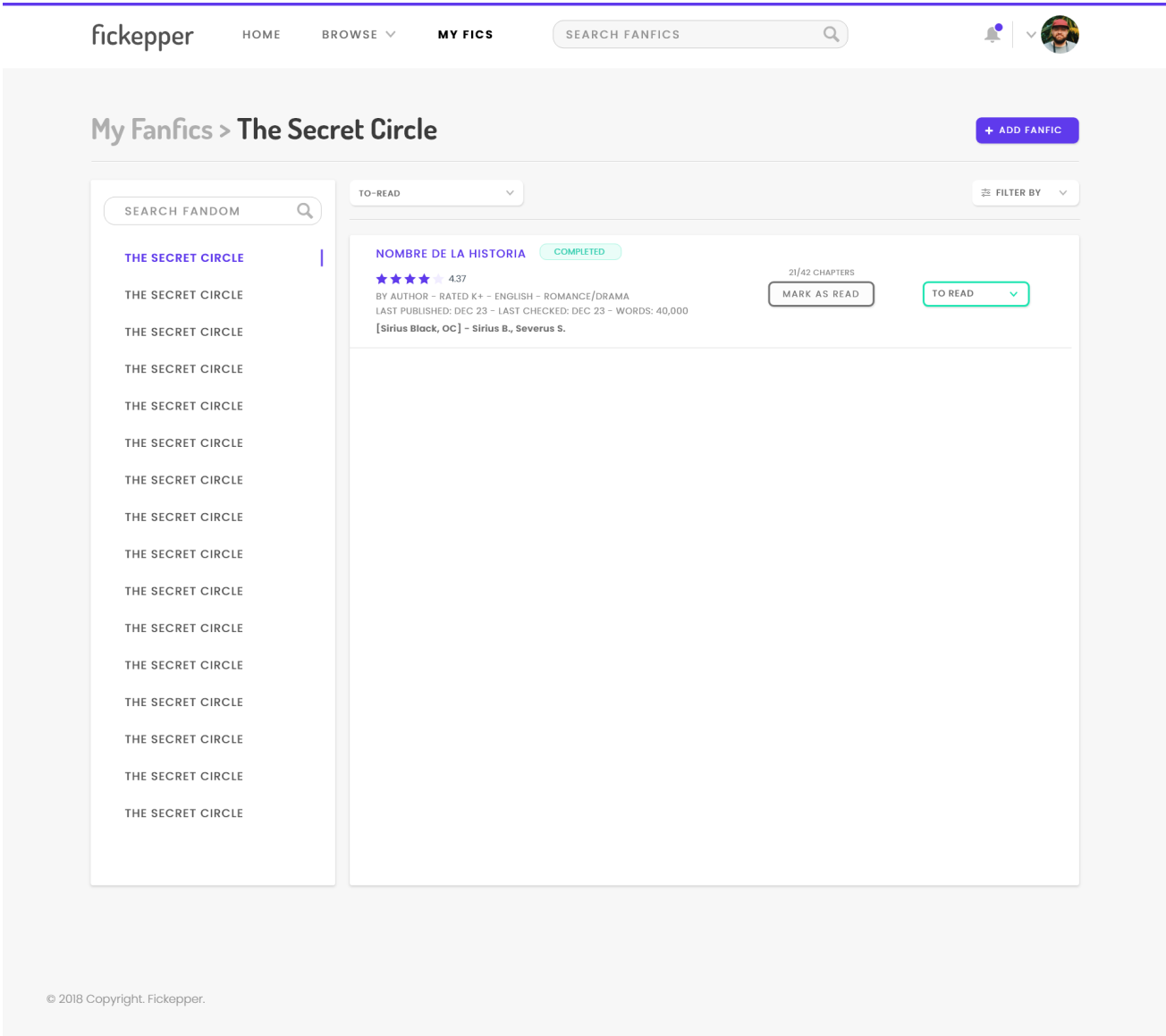


Figura 128: Prototipo de los *fanfics* propios



@johndoe22

27

FANFICS

7

REVIEWS

16

FOLLOWS

5

FOLLOWERS

[SEE ALL STORIES](#)

## John Doe

26, Male, Spain

<http://your-website.com>

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy.

### ♥ Fandoms

One, Two, Three with link

### ♥ Fanfics

One, Two, Three with link

### Recent Activity

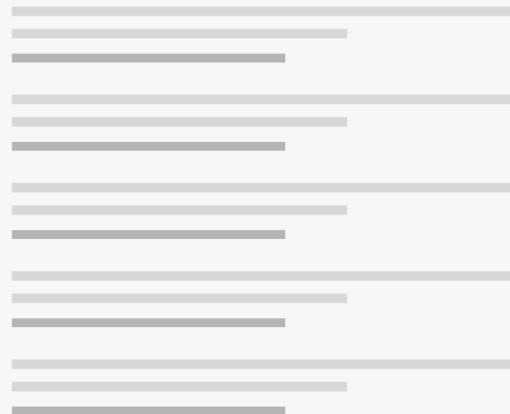


Figura 129: Prototipo de la página del perfil

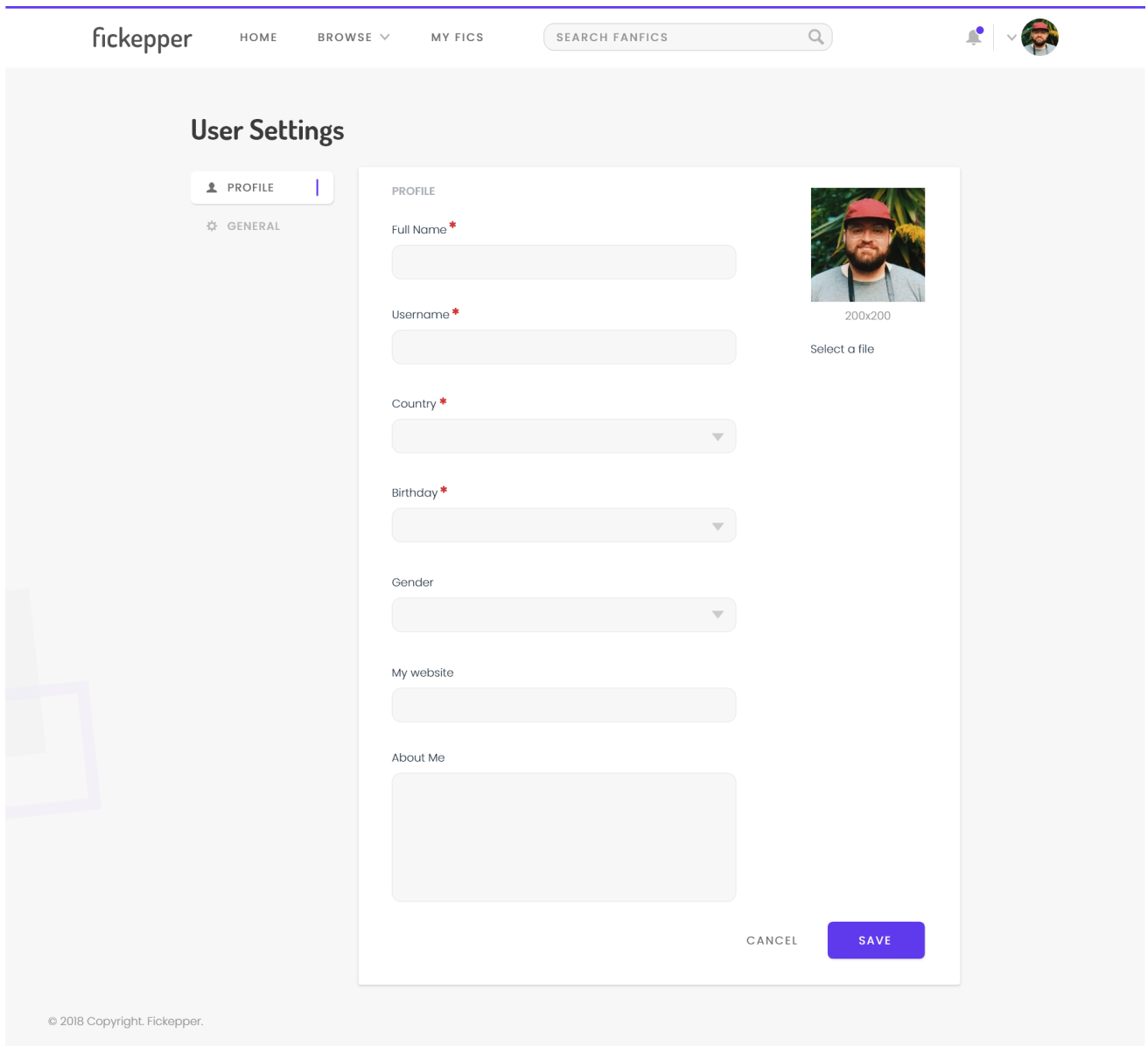


Figura 130: Prototipo de la configuración de la cuenta

## Track and follow your favorite fanfic stories.

Stop worrying about what you have available to read and just read.

GET STARTED FREE



### FEATURES



## Organize your fanfics

Keep all your favorites stories in one place and use lists to organize them. You will know what fics are on your to-read shelf right away.

## Track what you read

Know what chapters you have read and have left. You don't have to write it down anymore.



## New chapter alerts

If one of your added fanfics has a new chapter receive a notification and read it right away. Stop wasting time checking all of them.



Figura 131: Prototipo de la página de aterrizaje (I)

## Socialize with friends

Follow the people you want and see their favorite fandoms and fanfics. You can also see their public shelves.



## Start tracking the fanfics you read

Stop using notebooks and bookmarks and start doing it automatically.

GET STARTED FREE

FEATURES

CONTACT US

fickepper

TERMS OF SERVICE

PRIVACY POLICY

JOIN US ON



© 2018 Copyright. Fickepper.

Figura 132: Prototipo de la página de aterrizaje (II)

## 16. Anexo V: Manual de instalación

Este manual de instalación está dirigido a usuarios con Windows 10 (64 bits).

Se requiere tener instalada la versión 3.6.5 (64 bits) de *Python*, y tener instalado *XAMPP* para poder acceder a una base de datos. Una vez cumplidos los requisitos, se procederá a configurar la base de datos.

### 16.1. Configuración de la base de datos

Antes de crear el usuario *fickeep* y la base de datos correspondiente, si al intentar iniciar la consola de MySQL desde *XAMPP* se produce un error, entonces hay que acceder al fichero “my.ini” (Config), y debajo de la etiqueta “mysqld” añadir `skip-grant-tables`. Tras reiniciar el servicio se podrá acceder a MySQL.

Se introducirán los siguientes comandos en la consola de MySQL:

```
CREATE USER 'fickeep'@'%';
CREATE DATABASE fickeep;
GRANT ALL PRIVILEGES ON fickeep.* TO 'fickeep'@ '%' IDENTIFIED BY '';
```

### 16.2. Instalación de librerías necesarias

Se abrirá una consola de comandos de Windows, y se navegará a la carpeta *install* dentro del proyecto.

Se instala el cliente de MySQL:

```
pip install mysqlclient-1.4.2-cp36-cp36m-win_amd64.whl
```

Se instala el conector de MySQL y *python*, pulsando dos veces sobre el fichero “mysql-connector-python-8.0.16-py3.6-windows-x86-64bit.exe”.

Para evitar errores de dependencias, se instalará ‘Microsoft Visual Studio Community’ desde la carpeta *install* (“vs.community\_\_219878474.1559474219.exe”).

Cuando este fichero se ejecute, se descargarán e instalarán varios ficheros, y después de abrirá un panel. En este panel, se tendrá que seleccionar solamente lo mostrado en la pantalla de la Figura 133.

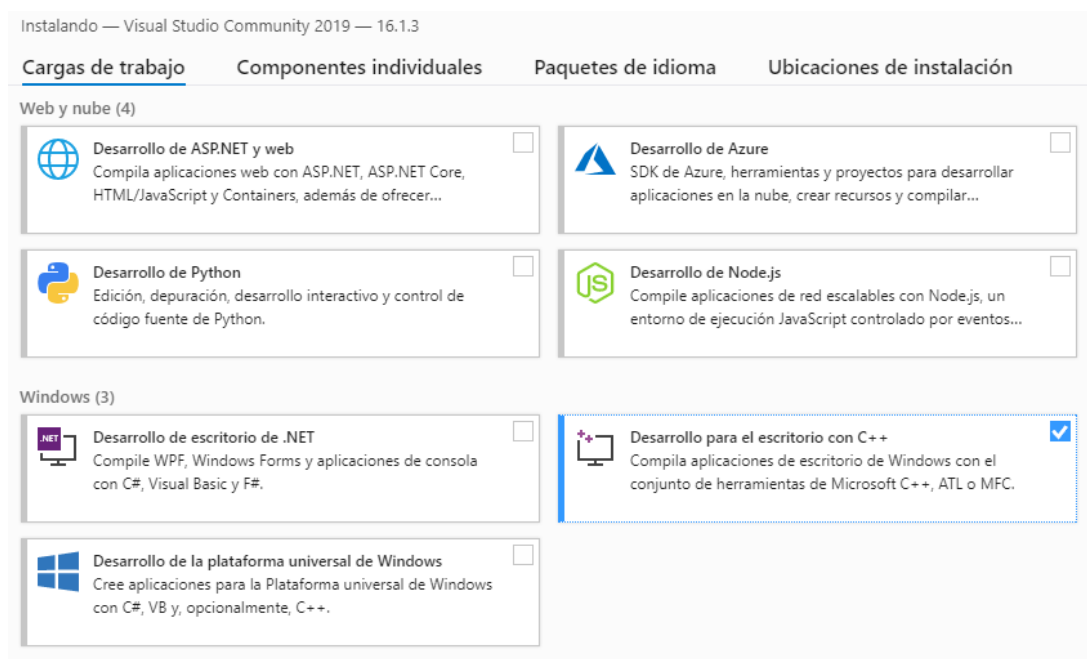


Figura 133: Instalación de Visual Studio

Tras instalar lo anterior, se introduce lo siguiente en una consola de comandos de Windows:



```
python -m pip install --upgrade pip
pip install -r requirements.txt
```

A continuación, se abre el directorio ‘code/fickeep’ en la consola de comandos anterior, y se ejecuta lo siguiente:

```
python manage.py migrate
python manage.py loaddata initial.json
```

Se procederá a instalar el servidor de *Redis* utilizando WSL (Windows Subsystem for Linux). Se va a suponer que dicho sistema ya está activado y está utilizando Ubuntu 18.04, por lo que solamente se abre la consola correspondiente y se ejecuta:

```
sudo apt-get update
sudo apt-get install redis-server
sudo service redis-server restart
```

Para comprobar que el servidor está activo, se abre una nueva terminal de WSL sin cerrar la anterior y se ejecuta lo siguiente:

```
redis-cli ping
```

Si la respuesta es *PONG*, el servidor funciona correctamente.

Para iniciar *Celery* se abren otras dos terminales de Windows dentro de la carpeta del proyecto que contenga al fichero *manage.py* (code/fickeep), y en cada una se escribe uno de los siguientes comandos:

```
celery -A fickeep worker -l info -P gevent
celery -A fickeep beat -l info
```

Por último, se abre otra terminal de Windows donde se encuentre el archivo *manage.py* y se ejecuta lo siguiente:

```
python manage.py runserver
```

La aplicación estará activa en la siguiente dirección:

```
http://127.0.0.1:8000/fickeeper
```

### 16.3. Credenciales

Los datos del administrador serán los siguientes:

```
Usuario: admin
Contraseña: A1.aaaaa
```

Y los de un bibliotecario:

```
Usuario: librarian
Contraseña: A1.aaaaa
```