



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

Trabajo Fin de Grado

**DESARROLLO DE UN
SISTEMA DE
RECOMENDACIONES DE
LIBROS BASADO EN
MEMORIA**

Alumno: María Martínez Baeza

Tutores: Prof. D^a. Carmen Martínez Cruz
Prof. D. Carlos Porcel Gallego

Dpto: Informática

Junio, 2020



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Doña CARMEN MARTÍNEZ CRUZ y Don CARLOS PORCEL GALLEGO, tutores del Proyecto de Fin de Carrera: DESARROLLAR UN SISTEMA DE RECOMENDACIONES DE LIBROS BASADO EN MEMORIA, que presenta María Martínez Baeza, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Junio de 2020

El alumno:

Los tutores:

María Martínez Baeza

Carmen Martínez Cruz

Carlos Porcel Gallego

Agradecimientos

En este apartado me gustaría agradecer a los tutores Doña Carmen Martínez Cruz y Don Carlos Porcel Gallego su ayuda durante todo el proceso de elaboración de este proyecto. Y especialmente a mi madre, por su apoyo incondicional.

Índice de contenidos

1. INTRODUCCIÓN	5
1.1. Motivación	7
1.2. Objetivos y metodología	8
1.2.1. Objetivos	8
1.2.2. Metodología a desarrollar	9
1.3. Estructura de la memoria	9
2. SISTEMAS DE RECOMENDACIÓN	11
2.1. Definición de sistema de recomendación	11
2.2. Tipos de sistemas de recomendación	12
2.2.1. Sistemas de recomendación basados en contenido	12
2.2.2. Sistemas de recomendación colaborativos	18
2.2.3. Sistemas de recomendación demográficos	22
2.2.4. Sistemas de recomendación basados en conocimiento	23
2.2.5. Sistemas de recomendación híbridos	23
2.6. Estado del arte	24
2.6.1. Amazon.com	24
2.6.2. Goodreads	25
2.6.3. TheFussyLibrarian	27
2.6.4. WhatShould I ReadNext?	28
2.6.5. Whichbook	30
3. ANÁLISIS INICIAL DEL PROBLEMA A RESOLVER	31
3.1. Características generales	32
3.2. Determinación del alcance	33
3.3. Usuarios del sistema	33
3.4. Requisitos funcionales	34
3.4.1. Diagrama de casos de uso del sistema	34
3.4.2. Caso de uso: Registro de usuario	35
3.4.3. Caso de uso: Acceso a la aplicación	35
3.4.4. Caso de uso: Editar perfil de usuario	36
3.4.5. Caso de uso: Obtener recomendaciones	37
3.4.6. Caso de uso: Valorar un libro	38
3.4.7. Caso de uso: Ver la lista de libros valorados	39
3.4.8. Caso de uso: Ver la lista de las recomendaciones recibidas	39
3.4.9. Caso de uso: Cambiar una valoración hecha a un libro	40

3.4.10. Caso de uso: Desconectarse del sistema	41
3.4.11. Caso de uso: Darse de baja del sistema	41
3.5. Requisitos no funcionales	42
3.6. Modelo de dominio	42
3.7. Elección del dataset	43
3.8. Tecnologías utilizadas	44
3.8.1. Spring Framework	45
3.8.2. HTML5	45
3.8.3. Bootstrap	46
3.8.4. Entorno de desarrollo: Netbeans 8.2.	46
3.9. Metodología de trabajo escogida: Scrum	47
4. APLICACIÓN DE LA METODOLOGÍA SCRUM AL PROYECTO	49
4.1. Reparto de roles	50
4.2. Product backlog (PB) del proyecto	50
4.3. Planificación de las iteraciones	51
4.4. Presupuesto	52
4.5. Iteraciones	56
4.5.1. Iteración 1	56
4.5.2. Iteración 2	58
4.5.3. Iteración 3	59
4.5.4. Iteración 4	60
4.5.5. Gráficos de la evolución de las iteraciones y del proyecto	61
4.5.5.1. Gráfico de la evolución de la iteración 1	61
4.5.5.2. Gráfico de la evolución de la iteración 2	62
4.5.5.3. Gráfico de la evolución de la iteración 3	64
4.5.5.4. Gráficos de la evolución de la iteración 4	64
4.5.5.5. Gráfico de la evolución del proyecto	65
5. IMPLEMENTACIÓN DE MYRECOMMENDER	67
5.1. Página de inicio	67
5.2. Registro de usuario	68
5.3. Página de portada	70
5.4. Menú de navegación	71
5.5. Recibir recomendación	73
5.6. Tus recomendaciones	78
5.7. Tus valoraciones	79
5.8. Menú del usuario	81
5.8.1. Editar perfil	82
5.8.2. Darse de baja	84
5.8.3. Desconectar	85

5.9. Buscar un libro	86
6. PRUEBAS	89
6.1. Pruebas de caja negra	89
6.1.1. Pruebas de caja negra realizadas	89
6.1.1.1. Pruebas de caja negra de la iteración 1	90
6.1.1.2. Pruebas de caja negra de la iteración 2	91
6.1.1.3. Pruebas de caja negra de la iteración 3	93
7. CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS	96
7.1. Conclusiones	97
7.2. Líneas de trabajo futuras	98
8. BIBLIOGRAFÍA	100
ANEXO. MANUAL DE INSTALACIÓN	106

1. INTRODUCCIÓN

Con la expansión de las tecnologías de la información a toda clase de ámbitos, las personas que intentan obtener información sobre cualquier campo de conocimiento están expuestas a una continua sobrecarga de información que hace difícil distinguir la que se ajusta más a sus necesidades [21]. No hay más que hacer una consulta en un buscador de Internet para obtener millones de resultados, que no tienen por qué ser relevantes para nosotros, ya que los resultados no están adaptados a los gustos o necesidades de la persona que la realiza [16].

Y es que en la sociedad de la información tenemos disponibles miles de alternativas para elegir sobre cualquier cosa: dónde viajar, con qué compañía vuelo, qué veo, qué leo, qué me interesa más [18]. Los buscadores pueden ayudar siempre que se haga la búsqueda de manera adecuada, y, muchas veces, un usuario no tiene la formación para hacerlo: términos parecidos, uso de comodines y comillas, limitaciones de la búsqueda, tipo de recursos buscados, etc. Por otro lado, un buscador no conoce nuestros gustos y no puede sustituir a un conocido que los conozca y a quién podemos pedir que nos haga una recomendación basándose en ellos.

Aquí es donde entrarían en escena los sistemas de recomendación, que pretenden ser los modernos sustitutos del boca a boca, que resulta ineficaz con los volúmenes de información que se manejan en la actualidad, y que son una herramienta fundamental para ayudarnos a encontrar los productos más adecuados para nosotros, entre la gran cantidad [16]. Cualquiera que navegue habitualmente por Internet habrá podido comprobar que el uso de los sistemas de recomendación cada vez está más extendido. Si lees una noticia, un artículo o ves un vídeo, lo más probable es que la página te sugiera algo que esté relacionado y que te puede interesar.

Los sistemas de recomendación [13] actúan como un filtro para el usuario, proporcionándole sólo la información que se adapta a sus preferencias de entre todas las opciones disponibles, de manera que el acceso a esta información sea

personalizada [13]. Son especialmente útiles para recursos que el usuario no conoce previamente, opciones que el usuario no habría tomado en consideración y que pueden acabar siendo un gran descubrimiento. No se trata de obtener información sobre un ítem del que ya se ha oído hablar sino de tener acceso a nuevas posibilidades que se ajustan a nuestro perfil [21].

La construcción del perfil del usuario para que refleje sus preferencias de forma precisa es, por tanto, una parte fundamental para el éxito del sistema. El modo de construirlo varía de unos modelos de sistemas de recomendación a otros, centrándose, por ejemplo, en las valoraciones que los usuarios han dado de un ítem para, como ocurre en los modelos colaborativos [6,20], o en la descripción de lo que éste debe tener para encajar con lo que busca el usuario, como en el caso de los basados en contenido [6,9,12]. Estos serían los modelos principales pero para tratar de solucionar los problemas que presentan, surgen los sistemas de recomendación híbridos [4,5], que utilizan más de una técnica para obtener la recomendación ofrecida al usuario.

1.1. Motivación

Se propone desarrollar un sistema Web para la recomendación de libros, que permita a los aficionados a la lectura descubrir sugerencias personalizadas sobre qué libros leer.

El sistema estará basado en un enfoque de recomendación híbrido, entre un sistema de recomendación colaborativo, y uno basado en contenidos, en el que la principal novedad está en que el proceso de generación de recomendaciones se completa con una memoria para recordar recomendaciones previas, que por restricciones no se enviaron al usuario y que ahora podrían ser interesantes.

Los perfiles se formarán teniendo en cuenta las restricciones que los usuarios han determinado sobre la cantidad y tipo de recomendaciones que desean recibir.

Así, el usuario debe seleccionar la cantidad de recomendaciones que quiere recibir, que pueden ser baja, media o alta, correspondiéndose con una,

tres o cinco recomendaciones de lectura. También tendrá que seleccionar el tipo de recomendaciones, que pueden ser nuevas o mezcla. Las nuevas recomendaciones son las que se generan desde cero, mientras que en el otro tipo la recomendación que se ofrece es el resultado de recuperar de la memoria algunas de las recomendaciones no ofrecidas al usuario por las limitaciones de número, pero que ya habían sido generadas en una ronda de recomendación anterior, y el resto se completa con nuevas recomendaciones.

Esto es interesante en el caso de los libros, porque cada año se añaden millones de libros al mundo editorial, con lo que en un sistema real contaría con una fuente constante de nuevos ítems disponibles para el usuario pero que a la par pueden ser una gran fuente de ruido para el usuario. Además, cuando se trata de libros, algunas recomendaciones no tan recientes pueden ser más interesantes que las más actuales.

1.2. Objetivos y metodología

En este apartado se introducen los objetivos del proyecto y la metodología a seguir para su elaboración.

1.2.1. Objetivos

El objetivo general del proyecto sería el desarrollo de un sistema de recomendación basado en memoria. Este objetivo se puede desglosar, a su vez en otros cuatro objetivos específicos:

1. Propuesta, desarrollo e implementación de un esquema de recomendación híbrido que genere recomendaciones basándose en una memoria y que por tanto, tendrá varias rondas para la generación de recomendaciones.
2. Desarrollar e implantar una plataforma Web para la recomendación de libros.
3. Seleccionar una fuente de datos para los libros e incluirlos en el sistema.
4. Implantar el sistema y probarlo con una batería de libros de prueba.

1.2.2. Metodología a desarrollar

La realización del proyecto seguirá la siguiente metodología:

- Análisis detallado para la definición del problema y búsqueda de información para analizar la viabilidad del mismo.
- Estudio de las diferentes herramientas, tanto hardware como software (incluyendo lenguajes de programación) que permitan el desarrollo del sistema propuesto, así como su implantación.
- Desarrollo del sistema basado en una arquitectura cliente-servidor.
- Puesta en marcha del sistema para realizar distintas pruebas.
- Redacción de un documento final siguiendo las especificaciones establecidas en un proyecto de ingeniería software.

1.3. Estructura de la memoria

La memoria se estructura en los siguientes capítulos, además del introductorio:

- Capítulo 2: En él se habla de los sistemas de recomendación, su tipología y las principales técnicas que utilizan. Además, se mencionan algunos recomendadores de libros disponibles en línea y se habla de su funcionamiento y del modo en el que ofrecen sus recomendaciones al lector.
- Capítulo 3: Se analiza el problema a resolver y las técnicas y utilidades empleadas para hacerlo.
- Capítulo 4: Se centra en la metodología Scrum, metodología ágil que se ha empleado para elaborar el proyecto.
- Capítulo 5: Plasma el resultado de la aplicación de Scrum y se centra en las iteraciones en las que se dividió el desarrollo del proyecto.
- Capítulo 6: Recoge un manual de la aplicación web e incluye imágenes de su funcionamiento.
- Capítulo 7: Este capítulo se centra en las pruebas realizadas en cada iteración a medida que se iba avanzando en la elaboración de la aplicación web que sirve de apoyo al sistema de recomendación.
- Capítulo 8: Se exponen aquí las conclusiones extraídas de este trabajo y

se proponen líneas de trabajo futuras.

- Capítulo 9: En este capítulo se localiza la bibliografía utilizada para la elaboración de la memoria.

Además, se incluye un anexo con una guía de instalación que muestra los pasos a seguir para poder utilizar la base de datos que maneja el sistema.

2. SISTEMAS DE RECOMENDACIÓN

En esta sección, primero se define lo que es un sistema de recomendación y, a continuación, se explican algunos de los tipos de recomendadores que existen.

2.1. Definición de sistema de recomendación

En primer lugar, hay que definir lo que es un sistema de recomendación. Se entiende como tal cualquier sistema que produce como salida una recomendación individualizada o que guía al usuario de forma personalizada hasta objetos útiles a través de un gran espacio de posibles opciones [4].

Por tanto, un sistema de recomendación es un agente software que ayuda a la toma de decisiones de un usuario a partir de sus intereses y preferencias, y su función es la de emparejar usuarios con ítems [23].

La sobrecarga de información a la que está expuesto un usuario de cualquier sistema, hace necesario su uso para poder filtrar la información, para recibir sólo la que realmente se adapte a nuestros gustos e intereses [21].

Esto hace que el uso de recomendadores sea especialmente atractivo en el caso del comercio electrónico, en el que resultan útiles tanto a los empresarios que quieren vender sus productos como a los clientes que quieren encontrar los que más les interesen [23].

Si nos centramos en el ámbito de la lectura, si un usuario quiere comprar un libro no puede leer los resúmenes de los millones de libros disponibles en Amazon [24] para ver cuál le interesa. Es necesario un filtro que le ayude a reducir esa cantidad. Y ese filtro es el usuario, sus preferencias e intereses, que el sistema tendrá que recoger y analizar de alguna manera para generar recomendaciones personalizadas.

Por tanto, sus principales características son dos:

1. Sus recomendaciones son personalizadas, lo que los diferencia de los resultados ofrecidos por un motor de búsqueda, que se esperan que sean relevantes por igual para todos los usuarios que realicen una determinada consulta.

2. No utilizan una técnica única, sino que se pueden distinguir distintos tipos de recomendadores según el tipo de entrada que utilizan para dar como salida esa recomendación.

2.2. Tipos de sistemas de recomendación

Según [21], desde el punto de vista teórico los sistemas de recomendación se pueden clasificar en dos categorías principales: sistemas de recomendación basados en contenido y sistemas de recomendación colaborativos. Sin embargo, en [20] se mencionan, además, otros dos tipos de recomendadores, los basados en demografía y los basados en conocimiento:

1. Los sistemas de recomendación basados en contenido recomiendan ítems o productos al usuario basándose en la información del contenido del mismo y en la relación del usuario con otros ítems parecidos en pasadas interacciones con el sistema.
2. Los sistemas de recomendación colaborativos hacen recomendaciones al usuario basadas en las que han sido hechas por usuarios que tienen perfiles similares.
3. Los sistemas de recomendación basados en demografía utilizan diferentes tipos de información demográfica de los usuarios para recomendar un ítem.
4. Los sistemas de recomendación basados en conocimiento realizan una recomendación a partir de la información que infieren sobre las características proporcionadas por los usuarios.

Estas técnicas de recomendación tienen sus ventajas y sus desventajas, que se comentarán en los siguientes apartados, por lo que a menudo se usan técnicas híbridas, que combinan dos o más técnicas de recomendación para tratar de paliarlas.

2.2.1. Sistemas de recomendación basados en contenido

Los sistemas de recomendación basados en contenido utilizan las características de un producto que ha sido elegido o valorado por el usuario con

el fin de buscar productos similares a éste en el sistema para hacer la recomendación. Esto requiere que un producto sea descrito para obtener un conocimiento significativo de los gustos del usuario, descripción que, idealmente, sería de igual nivel de profundidad para todos los objetos del sistema [6]. Se utiliza una representación del contenido de los ítems vistos, comprados o votados por los usuarios para ofrecer una recomendación al usuario. Esta representación suele hacerse en forma de vector, en el que cada elemento es un término sacado de las palabras clave, los metadatos o el texto descriptivo del objeto o producto [12].

Por ejemplo, si un usuario ha comprado los dos primeros libros de una trilogía escrita por un autor, el sistema le recomendará que compre el tercero cuando esté disponible.

La información sobre los gustos del usuario se puede obtener implícita o explícitamente. Pongamos el caso de un sistema de recomendación musical.

Si se quisiera recoger información del usuario de forma implícita, el sistema analizaría, por ejemplo, qué tipo de canciones ha escuchado el usuario o de qué grupos o cantantes son esas canciones u otra clase de información relevante, como el idioma de la canción.

Si se quisiera recoger información de forma explícita, el sistema solicitaría al usuario que valorara, por ejemplo, su interés sobre distintos géneros musicales, mediante un formulario o eligiendo un número determinado de géneros de entre los disponibles en una lista.

Como se indica en [9], para poder analizar un ítem, es necesario poder representarlo en base a sus características. La opción clásica es representarlo de forma plana, como un vector en el que cada posición corresponde a una posible característica y su contenido indicaría su valor, para cada uno de los ítems del sistema, como se muestra en la tabla 2.1.

Título	Autor	Editorial	Fecha de publicación	Tipo	Palabras clave
--------	-------	-----------	----------------------	------	----------------

Cartas en el asunto	Pratchett, Terry	Debolsillo	2012	Pasta blanda	Fantasía, humor, ficción, Mundodisco
---------------------	------------------	------------	------	--------------	--------------------------------------

Tabla 2.1.: Representación de un ítem mediante un vector de características.

[Elaboración propia]

Pero también se puede representar como una estructura jerárquica como en el formato XML o incluso con un texto en formato libre, que el sistema tendría que analizar para extraer características que sean útiles para realizar la recomendación, como se puede ver en la figura 2.2.

```
<personaje>
  <nombre>Aloy</nombre>
  <género>Femenino</género>
  <oficio>Cazadora</oficio>
  <cultura>Nora</cultura>
  <altura>1,69 cm.</altura>
</personaje>
```

Figura 2.2.: XML descriptivo del personaje de un videojuego. [Elaboración propia]

Respecto al cálculo de la similitud entre ítems en los sistemas de recomendación basados en contenido, en estos sistemas resulta especialmente útil la similitud que se basa en la coincidencia de las palabras clave que representan el contenido de un producto, con las que representan a otro dentro del sistema.

Para ello, se utiliza la siguiente fórmula para calcular la similitud entre el producto i y el producto j [14]:

$$Similitud (p_i , p_j) = \frac{2 \times | palabras clave (p_i) \cap palabras clave (p_j) |}{| palabras clave (p_i) | + | palabras clave (p_j) |}$$

Fórmula 2.1.: Similitud entre objetos basada en las palabras clave

Donde: p_i y p_j son dos productos

Pero la aplicación de esta fórmula de similitud simple presenta dos problemas principales:

1. No todas las palabras tienen la misma importancia dentro de un texto o documento
2. Los documentos que tienen una mayor longitud tienen más probabilidad de contener palabras que casen con las que interesan al usuario.

Por esta razón, se introdujo otra medida, TF-IDF [1]. Es una medida numérica que determina la relevancia de una palabra dentro de un texto o documento, que es muy usada en motores de búsqueda.

Supongamos que queremos hacer una búsqueda en el catálogo de una biblioteca electrónica. Si buscamos "guerra civil española". Lo primero que hará el sistema será tener en cuenta sólo los documentos que contengan esas tres palabras, pero todavía seguiría ofreciendo una gran cantidad de resultados. Sería el momento de utilizar TF-IDF para filtrarlos aún más.

Para cada palabra clave de un documento se calcularía su TF-IDF, que establece la importancia de un término en un documento y en el conjunto de estos. TF (Term Frequency) mide la frecuencia con la que es usado un término en un texto o documento, como se muestra en la fórmula 2.2., mientras que IDF (Inverse Term Frequency), mide la relevancia de un término dentro del texto o documento, como muestra la fórmula 2.3.

$$TF = \frac{Frecuencia(i, j)}{\max_z Frecuencia(z, j)}$$

Fórmula 2.2.: TF frecuencia del término i

Donde i es un término

j es un documento

z es el término que más veces aparece en el documento j .

$$IDF = \log \frac{1 + N}{n(i)}$$

Fórmula 2.3.: IDF inversa de la frecuencia del término i

Donde N es el total de documentos candidatos a ser recomendados
 n(i) el número total de documentos de N en los que aparece el término i
 1 es un valor de corrección añadido a la fórmula.

Con estos dos elementos, se puede calcular el TF-IDF, como se muestra en la fórmula 2.4.

$$TF-IDF(i, j) = TF(i, j) \times IDF(i, j)$$

Fórmula 2.4.: TF-IDF

Supongamos que tenemos un documento con quinientas palabras clave en el que el término "dinosaurio" aparece dos veces y en el que el término más frecuente en ese documento es "fósil", que aparece diez veces, y que hay cien documentos, pero el término "dinosaurio" sólo está presente en uno de estos documentos. Si aplicamos las fórmulas anteriores, el resultado sería el que podemos ver en la tabla 2.2.

TF	TF=2/10	0,2
IDF	IDF=log (1+100/1)	2
TF-IDF	TF-IDF=TF*IDF	0,4

Tabla 2.2.: Ejemplo de cálculo de TF, IDF y TF-IDF

Supongamos ahora que tenemos 3 documentos en los que se quiere comprobar su relevancia para la búsqueda "guerra civil española" y que la matriz

que agrega los vectores para los resultados de la frecuencia de los términos es la mostrada en la tabla 2.3.

Frecuencia	Documento 1	Documento 2	Documento 3
guerra	18	5	10
civil	9	2	1
española	1	2	0

Tabla 2.3.: Frecuencia de los términos en los documentos. [Elaboración propia]

Si aplicamos las fórmulas para el cálculo del TF sobre estos documentos en base a esas tres palabras clave, obtendremos las cifras mostradas por la tabla 2.4.

TF(i,j)	Documento 1	Documento 2	Documento 3
guerra	1	1	1
	TF=18/18	TF=5/5	TF=10/10
civil	0,5	0,4	0,01
	TF=9/18	TF=2/5	TF=1/10
española	0,06	0,4	0
	TF=1/18	TF=2/5	TF=0/10

Tabla 2.4.: Cálculo del TF en los documentos. [Elaboración propia]

Ahora, supongamos que en el sistema sólo existen estos 3 documentos recomendables, así que volveríamos a aplicar la fórmula para obtener el IDF, como muestra la tabla 2.5.

IDF(i,j)	Documento 1	Documento 2	Documento 3
guerra	0,12	0,12	0,12

	IDF=log (1+3/3)	IDF=log (1+3/3)	IDF=log (1+3/3)
civil	0,12	0,12	0,12
	IDF=log (1+3/3)	IDF=log (1+3/3)	IDF=log (1+3/3)
española	0,3	0,3	0,6
	IDF=log (1+3/2)	IDF=log (1+3/2)	IDF=log (1+3/0)

Tabla 2.5.: Cálculo del IDF en los documentos. [Elaboración propia]

Y por último, aplicaríamos la fórmula para el cálculo del TF-IDF, tal y como muestra la tabla 2.6.

TF-IDF(i,j)	Documento 1	Documento 2	Documento 3
guerra	0,12	0,12	0,12
	TF-IDF=1*0,12	TF-IDF=1*0,12	TF-IDF=1*0,12
civil	0,06	0,05	0,12
	TF-IDF=0,5*0,12	TF-IDF=0,4*0,12	TF-IDF=1*0,12
española	0,02	0,12	0
	TF-IDF=0,06*0,3	TF-IDF=0,4*0,3	TF-IDF=0*0,6

Tabla 2.6.: Cálculo del TF-IDF en los documentos. [Elaboración propia]

Como es lógico, los resultados indican que la mejor opción recomendable es el Documento 2, seguida por el Documento 1. Ya que ambos contienen los 3 términos buscados, pero se tiene en cuenta el peso de una palabra dentro del documento, cuando ésta no es la que más veces aparece.

2.2.2. Sistemas de recomendación colaborativos

Los sistemas de recomendación colaborativos utilizan a las valoraciones que han hecho los usuarios sobre los ítems del sistema para hacer sus recomendaciones. Buscan a los usuarios que más se asemejan a un usuario para recomendarle un producto infiriendo la valoración que le daría el usuario,

en caso de haber valorado el producto, de las valoraciones dadas por los otros usuarios.

Para ello el sistema de recomendación colaborativo ha de tener usuarios, valoraciones e ítems a recomendar, dados en un dominio de valoración determinado: restaurantes, canciones, películas, etc.

Se pueden distinguir dos clases de sistemas de recomendación colaborativos, los basados en memoria y los basados en modelos [17].

Los primeros utilizan la información de las valoraciones de los usuarios del sistema para calcular la similitud entre pares de usuarios analizando las valoraciones que ambos han hecho de los mismos ítems.

Los segundos, crean un modelo a partir de la matriz de valoraciones para establecer cuáles son los usuarios similares al usuario al que se quiere hacer una recomendación.

Un ejemplo de sistema de recomendación colaborativo muy conocido es la web Filmaffinity [29]. A partir de las votaciones de un usuario, lo empareja con otros usuarios similares y le ofrece recomendaciones de películas que éstos han visto y el usuario no, de modo que el usuario descubre nuevos contenidos que le pueden interesar.

Ahora bien, una vez que se han calculado los usuarios más similares al usuario, se necesita inferir la valoración que tendría un ítem para el usuario en aquellos ítems que éste no ha valorado todavía. Para ello, hay que calcular la similitud entre usuarios, seleccionar los más afines y generar una recomendación en base a las que hicieron éstos usuarios.

Por tanto, hay que realizar tres tareas principales. La primera tarea consistiría en elegir una forma de calcular la similitud. Las dos medidas más usadas para calcularla son el Coeficiente de Correlación de Pearson y la Coeficiente del Coseno [14].

La segunda tarea sería determinar el número de usuarios similares cuyas votaciones se van a considerar para hacer recomendaciones al usuario sobre ítems no valorados. La medida más utilizada es la de los k-vecinos más

cercanos, esto es, los k usuarios con mayor similitud al usuario a considerar.

Y la tercera tarea, una vez escogido el número de usuarios que van a formar parte de este vecindario, se ha de inferir la valoración para el ítem no valorado.

El coeficiente de Correlación de Pearson sirve para medir el grado de similitud de dos variables, siempre y cuando ambas sean cuantitativas y continuas, obteniéndose una cifra entre -1 y 1, y su cálculo se realiza de la manera que se muestra en la fórmula 2.5.

$$\text{similitud}(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

Fórmula 2.5.: Coeficiente de Correlación de Pearson

Donde \bar{x} es la media aritmética de los valores de x

\bar{y} es la media aritmética de los valores de y

x_i valor del ítem i en el vector x

y_i valor del ítem i en el vector y

n se nota el número de usuarios

El coeficiente del Coseno es un método que necesita que los ítems sean representados mediante vectores. La similitud entre dos vectores x e y viene determinada por el coseno del ángulo que forman y su fórmula (fórmula 2.6.) es la siguiente:

$$\text{similitud}(x, y) = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n (x_i)^2} \sqrt{\sum_{i=1}^n (y_i)^2}}$$

Fórmula 2.6.: Coeficiente del Coseno

Donde x_i es el valor del ítem i en el vector x

Desarrollar un sistema de recomendaciones de libros basado en memoria

y_i es el valor del ítem i en el vector y

n es el número de usuarios

Por ejemplo, en la tabla 2.7. tenemos 3 películas y 3 usuarios. 2 de las películas han sido valoradas por todos los usuarios pero la película 3 no ha sido valorada por el usuario 2. ¿Es esta película una candidata a ser recomendada a este usuario?

	Película 1	Película 2	Película 3
Usuario 1	5	3	5
Usuario 2	4	3	?
Usuario 3	2	2	1

Tabla 2.7.: Ejemplo de valoraciones de usuarios en un sistema de recomendación de películas. [Elaboración propia]

Para saberlo habría que calcular la similitud entre los usuarios, que en este ejemplo se ha calculado utilizando la similitud del coseno, con los resultados que se muestran en la tabla 2.8.

Sim (Usuario 2, Usuario 1)	0,99
Sim (Usuario 2, Usuario 3)	0,68

Tabla 2.8.: Similitud entre usuarios en un sistema de recomendación de películas. [Elaboración propia]

Como se puede ver, la similitud con el usuario 1 es mucho mayor y tendría más posibilidad de ser elegido entre los k-vecinos del usuario 2.

Restaría utilizar una de las técnicas existentes para inferir cuál sería la valoración hecha por el usuario del ítem para sugerírselo o no como recomendación. Según [10], la forma más sencilla de hacerlo es calcular una media ponderada de las valoraciones, aunque una de las aproximaciones más utilizadas es calcular una suma media ajustada de dichas puntuaciones, utilizando la media de valoración de cada individuo y las correlaciones como pesos.

2.2.3. Sistemas de recomendación demográficos

Los sistemas de recomendación demográficos, utilizan el perfil demográfico de los usuarios para realizar sus recomendaciones, identificando el tipo de ítem

que prefiere un tipo de usuario [19]. Este perfil contendría información sobre la edad, la formación académica, el género, empleo, lugar en el que vive, etc., y sería utilizada para crear grupos demográficos [2] que suponen tendrán unos intereses comunes.

Suponen, por ejemplo, que usuarios que viven en un mismo lugar y que pertenecen a una misma franja de edad y hablan el mismo idioma, tienen un gusto común, y por lo tanto, se les puede recomendar lo mismo, por ejemplo, un restaurante.

Los autores coinciden en que si bien esta técnica ha demostrado ser eficaz para algunos sectores, es difícil recopilar esta clase de información de los usuarios.

2.2.4. Sistemas de recomendación basados en conocimiento

Los sistemas de recomendación basados en conocimiento no dependen de las valoraciones y sus recomendaciones son independientes de los gustos particulares de los usuarios [3]. Estos recomendadores localizan un ítem que cumple con las características que el usuario indica que le interesan, es decir, buscan la recomendación que sea más útil para las necesidades del usuario.

Por ejemplo, un usuario está interesado en localizar un restaurante que cumpla con dos características: que sirva comida china y tailandesa. El sistema le ofrece un resultado pero el dinero que cuesta comer en ese restaurante le parece demasiado caro al usuario, por lo que añade a las características deseadas el dinero máximo que está dispuesto a pagar. En una segunda recomendación, se le recomienda un restaurante, que si bien sólo ofrece comida tailandesa en su menú, cumple con la característica del precio.

2.2.5. Sistemas de recomendación híbridos

Los sistemas de recomendación híbridos tratan de aprovechar las ventajas de dos o más técnicas de recomendación para tratar de evitar o minimizar sus desventajas [4,5].

Por ejemplo, como se recoge en [11], la técnica de recomendación basada en demografía presenta problemas cuando se tienen pocos usuarios a

comparar al inicio, el llamado "*cold-start problem*" o arranque en frío, mientras que las recomendaciones colaborativas los tienen cuando no hay suficientes votaciones para los ítems incorporados al sistema recientemente.

2.6. Estado del arte

En la red, se pueden encontrar muchas webs que incluyen recomendadores para libros, ya sea con un objetivo comercial o no. Para esta sección se han seleccionado algunos ejemplos, eligiendo una muestra de recomendadores cuyo funcionamiento, al menos en lo que respecta a la interacción con el usuario, es diferente.

2.6.1. Amazon.com



Figura 2.2.: Logo de amazon.

Sin duda, cualquier repaso a sistemas de recomendación tiene que incluir a amazon [24], cuyo logo se puede ver en la figura 2.2., que ha desarrollado un sistema de recomendación propio, que ha denominado recomendación colaborativa ítem a ítem y que logra ser eficaz a pesar de tener que analizar datos de decenas de millones de usuarios.

En [15] podemos ver cómo funciona. Utiliza los productos comprados o valorados por un usuario para crear las listas de recomendaciones. Calcula qué ítems son similares al comprado por un usuario a partir de los ítems que han aparecido junto a éste en un pedido, carrito de la compra o lista de deseos de otros usuarios.

Por eso, si la compra de un usuario incluye 3 objetos y la de otro incluye 2

de ellos, es probable que el tercer objeto se le ofrezca como recomendación de compra al segundo usuario.

Si realizamos una búsqueda de un libro y pulsamos sobre él para ver sus detalles se nos informará de qué libros compran otros usuarios que compraron el libro y qué libros miran los usuarios que miraron los detalles del libro, tal y como muestra el ejemplo de la figura 2.3.

Comprados juntos habitualmente



Precio total: **EUR 28,35**
[Añadir los tres a la cesta](#)

- Este producto:** El Color de la Magia (Mundodisco 1) por Terry Pratchett Libro de bolsillo **EUR 9,45**
- La Luz Fantástica (Mundodisco 2) por Terry Pratchett Tapa blanda **EUR 9,45**
- Ritos Iguales (Mundodisco 3) por Terry Pratchett Libro de bolsillo **EUR 9,45**

Los clientes que vieron este producto también vieron

Página 1 de 12



Mort (Mundodisco 4)
› Terry Pratchett
★★★★☆ 38
Tapa blanda
8,50 € ✓prime



¡Guardias! ¡Guardias!
(Mundodisco 8)
› Terry Pratchett
★★★★☆ 24
Tapa blanda
9,45 € ✓prime



Ritos Iguales (Mundodisco 3)
› Terry Pratchett
★★★★☆ 16
Libro de bolsillo
9,45 € ✓prime



Brujerías (Mundodisco 6)
› Terry Pratchett
★★★★☆ 18
Tapa blanda
9,45 € ✓prime



La Luz Fantástica
(Mundodisco 2)
› Terry Pratchett
★★★★☆ 17
Tapa blanda
9,45 € ✓prime

Figura 2.3.: Información sobre libros comprados o vistos por los que ven un libro en amazon.com

2.6.2. Goodreads



Figura 2.4.: Logo de goodreads.

En esta web [25], cuyo logo se puede ver en la figura 2.4., una vez que el usuario se ha registrado, se le presentan una lista de géneros para elegir. En el siguiente paso, se invita al usuario a valorar una serie de títulos de las categorías elegidas, de los que tiene que puntuar al menos veinte para recibir una buena recomendación.

Cuando se le muestran los libros recomendados de cada categoría, el usuario puede valorarlos o indicar que no son de su interés, como se muestra en el ejemplo de la figura 2.5., lo que evitará que se le muestren libros similares en futuras recomendaciones. Las recomendaciones irán siendo mejores si el usuario valora más libros de los géneros de su interés.

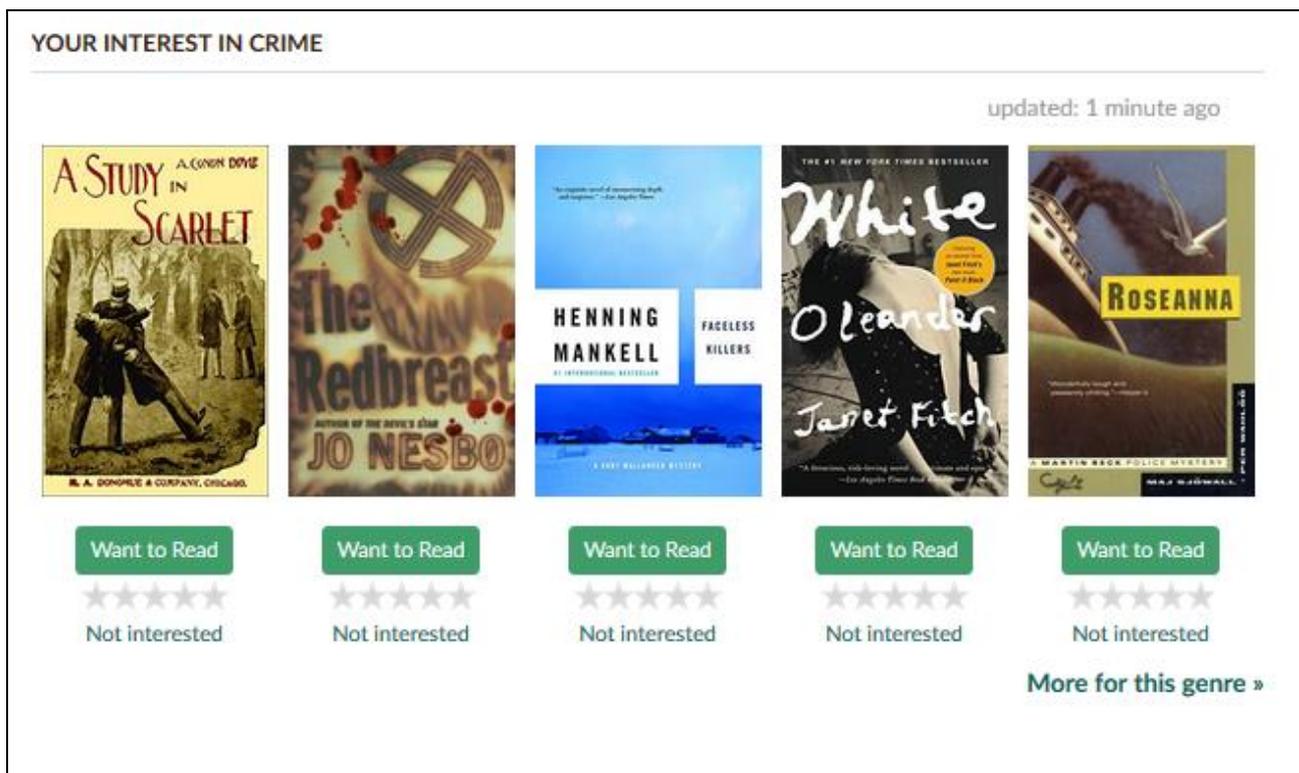


Figura 2.5.: Recomendaciones por género en goodreads

2.6.3. TheFussyLibrarian



Figura 2.6.: Logo de TheFussyLibrarian

En la web de TheFussyLibrarian [26], cuyo logo se puede ver en la figura 2.6., los usuarios tienen la opción de suscribirse a dos boletines de novedades, de ebooks gratuitos y de ebooks de pago, proporcionando un e-mail. En ambas opciones el usuario puede expresar sus preferencias en cuanto a géneros que

le interesa y con qué frecuencia quiere recibir las recomendaciones. Pero sólo en la opción de suscripción a ebooks de pago se puede afinar en las preferencias del usuario, añadiendo información sobre contenido de violencia, sexual o tipo de lenguaje.

2.6.4. WhatShould I ReadNext?



Figura 2.7.: Logo de Whatshould I readnext?

La web de WhatShould I ReadNext [27], cuyo logo se puede ver en la figura 2.7., tiene un cuadro de búsqueda en el que si se escribe el título de un libro, una palabra clave o el nombre de un autor, se muestra un listado de libros que lo contiene, que aparecen ordenados por relevancia, como muestra la figura 2.8. No es necesario registrarse para utilizar esta opción.



Figura 2.8.: Recomendaciones en What I shouldreadnext?

Si queremos una recomendación más personalizada, hay que proporcionar información como género del usuario, país, edad, géneros literarios de interés, hábitos lectores, etc., mediante un formulario muy detallado.

2.6.5. Whichbook



Figura 2.9.: Logo de whichbook.

En Whichbook.net [28], cuyo logo se muestra en la figura 2.9., hay unos controles deslizantes con los que podemos indicar el grado de coincidencia entre las parejas de adjetivos contrarios disponibles, por ejemplo divertido-triste, que buscamos en nuestra próxima lectura, tal y como muestra la figura 2.10.

Además, en los resultados obtenidos se nos indica cómo de buena es la recomendación que se nos ha hecho en cuanto al grado de coincidencia con las preferencias introducidas.

The screenshot shows the 'whichbook' website interface. On the left, there is a vertical stack of sliders for filtering books. Each slider has two labels and a central triangle marker. The sliders are: 'happy' vs 'sad', 'funny' vs 'serious', 'safe' vs 'disturbing', 'expected' vs 'unpredictable', 'larger than life' vs 'down to earth', 'beautiful' vs 'disgusting', 'gentle' vs 'violent', 'easy' vs 'demanding', 'no sex' vs 'sex', 'conventional' vs 'unusual', 'optimistic' vs 'bleak', and 'short' vs 'long'. The 'funny' slider is currently positioned towards 'funny', and the 'easy' slider is positioned towards 'demanding'. The main content area on the right is titled 'Good matches' and lists two books: 'The Prophet Murders' by Mehmet Murat Somer and 'The First Bad Man' by Miranda July. Each book entry includes a short review and links for 'Extract', 'Parallels', 'Profile', and 'Find similar'. On the far right, there are promotional banners for 'The Prophet Murders' and 'The First Bad Man', with 'Borrow' and 'Buy' buttons.

Figura 2.10.: Controles deslizantes y recomendaciones en wichbook

3. ANÁLISIS INICIAL DEL PROBLEMA A RESOLVER

En este apartado, partiendo de unas características generales, se extraen los requisitos funcionales y no funcionales del proyecto, se determinan los casos de uso, el alcance, los usuarios a los que va dirigido, y se explica cómo se ha desarrollado, incluyendo casos de uso, el modelo de dominio y las tecnologías, los datasets y los métodos utilizados.

3.1. Características generales

El proyecto consiste en realizar una aplicación web que incorpora un sistema de recomendación de libros basados en memoria, dirigido a cualquier persona interesada en la lectura, que debe incluir las opciones que se mencionan a continuación. Se propone un nuevo esquema de recomendación híbrido que se ha integrado en la plataforma web para obtener recomendaciones más personalizadas y de mayor utilidad para los usuarios.

Un usuario tiene que registrarse en el sistema para poder acceder al mismo. Los usuarios podrán registrarse dando un nombre de usuario, que no esté siendo utilizado por otro usuario, y un password o contraseña. Una vez registrado un usuario, ya puede entrar a la aplicación proporcionando las credenciales con las que se ha registrado.

Para poder recibir recomendaciones, es necesario que la primera vez que solicite recibir recomendaciones, los usuarios valoren una serie de libros, los veinte más valorados por otros usuarios en el sistema. De este modo, el sistema trata de paliar el problema de arranque en frío (cold-start problem) de los sistemas colaborativos y se pueden ofrecer recomendaciones, basadas en estas votaciones iniciales, al usuario.

Además tendrá que especificar el tipo y número de recomendaciones que quiere recibir, que como ya se ha indicado en un capítulo anterior podrán ser de tipo nuevas recomendaciones o mezcla de recomendaciones nuevas y extraídas de la memoria, y el número bajo, medio o alto. El número de recomendaciones

de la mezcla que proceden de la memoria será la mitad del número de recomendaciones a recibir seleccionado y el resto se completarán con recomendaciones nuevas.

Las valoraciones se harán utilizando un juego de etiquetas manejado por el sistema para este fin.

El sistema guardará un registro de las recomendaciones hechas a un usuario y también se tendrá un registro de los libros valorados por él. Estas dos informaciones podrán ser consultadas por el usuario

El usuario podrá modificar su perfil, cambiando alguno de los datos que lo componen: tipo y número de recomendaciones que desea recibir, nombre de usuario y password.

Además del perfil, el usuario podrá modificar la valoración hecha a un libro y esta nueva valoración vendrá a sustituir a la que hizo anteriormente.

Un usuario que ha hecho login en el sistema podrá seleccionar desconectarse del mismo y también podrá darse de baja del sistema, si decide que ya no quiere seguir utilizando la aplicación.

La web tendrá una interfaz sencilla e intuitiva y recogerá las características mencionadas anteriormente.

3.2. Determinación del alcance

Se trata de crear la web de un recomendador de libros basado en memoria, que ofrezca recomendaciones personalizadas a los usuarios registrados, basándose en su similitud con otros usuarios.

El sistema de recomendación va a ser híbrido, a medio camino entre un sistema de recomendación colaborativo y uno basado en conocimiento y apoyándose en el uso de una memoria, que contendrá recomendaciones generadas pero no ofrecidas al usuario hasta el momento.

3.3. Usuarios del sistema

El grupo de usuarios del sistema lo forman personas interesadas en la lectura, que quieran recibir información sobre sus próximas lecturas en base a

las valoraciones hechas por otros usuarios.

3.4. Requisitos funcionales

En base a las características generales analizadas en la sección 3.1., se han extraído una serie de casos de uso, que se irán detallando en las siguientes subsecciones.

3.4.1. Diagrama de casos de uso del sistema

La figura 3.1. muestra el diagrama de casos de usos del sistema, que se comentarán en las subsecciones siguientes.

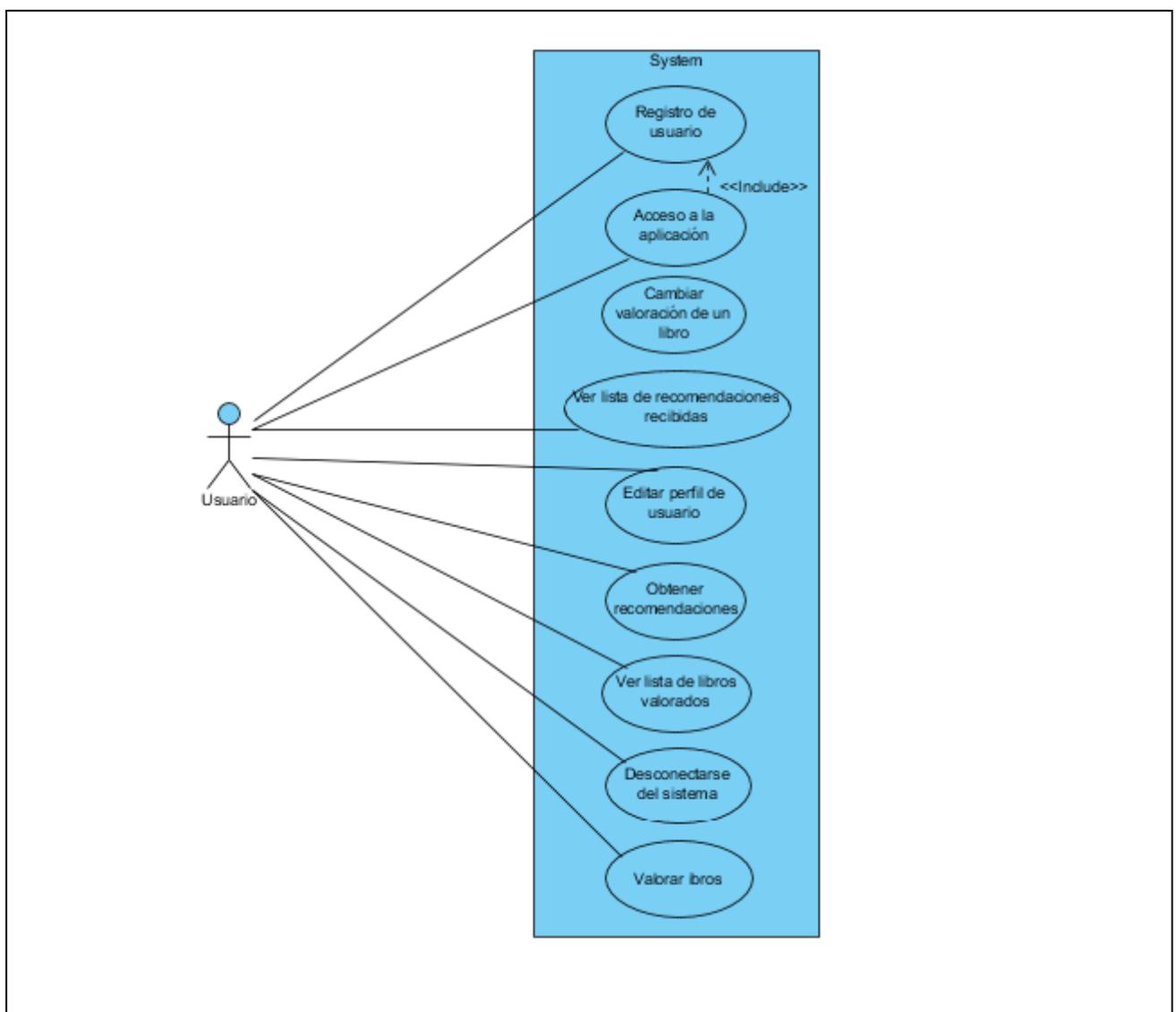


Figura 3.1.: Diagrama de casos de uso del sistema. [Elaboración propia]

3.4.2. Caso de uso: Registro de usuario

Un usuario podrá registrarse en el sistema, proporcionando un nombre de usuario, que no podrá estar siendo utilizado por otro usuario, y una contraseña, como se detalla en la tabla 3.1.

Caso de uso principal 1	
Nombre	Registro de usuario
Descripción	Un usuario se registra en el sistema para poder tener credenciales que le permitan utilizar la aplicación
Actor/es participantes	Usuario
Condiciones de entrada	1. Nombre de usuario 2. Contraseña
Flujo de eventos	1. El usuario selecciona acceder al registro de usuario 2. El usuario introduce un nombre de usuario 3. El usuario introduce una contraseña 4. El usuario selecciona crear usuario 5. El usuario visualiza un mensaje de usuario creado
Condiciones de salida	Se crea un nuevo usuario en el sistema, con las credenciales proporcionadas

Tabla 3.1.: Caso de uso principal 1. [Elaboración propia]

3.4.3. Caso de uso: Acceso a la aplicación

Como muestra la tabla 3.2., un usuario podrá acceder al sistema proporcionando el usuario y la contraseña con los que se registró.

Caso de uso principal 2	
Nombre	Acceso a la aplicación
Descripción	El usuario accede a la aplicación, introduciendo un nombre de usuario y una contraseña en la página de login
Actor/es participantes	Usuario
Condiciones de entrada	1. Nombre de usuario 2. Contraseña
Flujo de eventos	1. El usuario selecciona acceder a la aplicación desde la página de inicio 2. El usuario introduce su nombre de usuario 3. El usuario introduce su contraseña 4. El usuario pulsa sobre "Login"
Condiciones de salida	Se abre la página de portada

Tabla 3.2.: Caso de uso principal 2. [Elaboración propia]

3.4.4. Caso de uso: Editar perfil de usuario

Un usuario podrá editar su perfil de usuario en cualquier momento, cambiando alguno de los datos que lo forman, de modo que los cambios queden guardados en el sistema. Los detalles de este caso de uso se muestran en la tabla 3.3.

Caso de uso principal 3	
Nombre	Editar perfil de usuario
Descripción	El usuario puede editar su perfil, modificando el número y tipo de recomendaciones que quiere recibir.

Actor/es participantes	Usuario
Condiciones de entrada	Número y tipo de recomendaciones.
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario selecciona editar perfil en su menú de usuario 2. El usuario selecciona el número de recomendaciones a recibir 3. El usuario selecciona el tipo de recomendaciones a recibir 4. El usuario selecciona guardar
Condiciones de salida	El sistema actualiza las opciones seleccionadas por el usuario en su perfil

Tabla 3.3.: Caso de uso principal 3 [Elaboración propia]

3.4.5. Caso de uso: Obtener recomendaciones

En este caso de uso, detallado en la tabla 3.4, un usuario podrá recibir recomendaciones personalizadas en la aplicación. Para ello, la primera vez que las solicite, deberá expresar su interés acerca de la lista de los 20 libros más valorados por los usuarios del sistema.

Caso de uso principal 4	
Nombre	Obtener recomendaciones
Descripción	El usuario solicita recibir recomendaciones de libros
Actor/es participantes	Usuario
Condiciones de entrada	Ninguna
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario selecciona recibir recomendaciones 2. El usuario recibe una lista de

	recomendaciones
Condiciones de salida	Lista de libros recomendados por el sistema, en número establecido por el usuario en su perfil

Tabla 3.4.: Caso de uso principal 4. [Elaboración propia]

3.4.6. Caso de uso: Valorar un libro

Un usuario podrá valorar su interés en un libro y esta valoración quedará registrada en el sistema. La valoración se hará mediante la selección de una etiqueta a elegir entre cinco posibles: nada, bajo, medio, alto y total. Esta información se guardará en la base de datos en una tabla en la que cada registro tendrá el identificador de la valoración, el identificador del libro y la valoración del usuario. Este caso de uso se detalla en la tabla 3.5.

Caso de uso principal 5	
Nombre	Valorar un libro
Descripción	Un usuario puede dar una valoración para un libro
Actor/es participantes	Usuario
Condiciones de entrada	Valoración para el libro
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario selecciona ver los detalles de un libro 2. El usuario selecciona su valoración para el libro 3. El usuario selecciona valorar el libro
Condiciones de salida	La valoración será almacenada en la base de datos

Tabla 3.5.: Caso de uso principal 5. [Elaboración propia]

3.4.7. Caso de uso: Ver la lista de libros valorados

Un usuario podrá ver un listado de los libros que ha valorado en el sistema, con la valoración que ha dado a cada uno, según muestra la tabla 3.6.

Caso de uso principal 6	
Nombre	Ver la lista de libros valorados
Descripción	El usuario puede visualizar un listado de los libros que ha valorado en el sistema, con detalles de la valoración que ha dado a cada uno de ellos
Actor/es participantes	Usuario
Condiciones de entrada	Ninguna
Flujo de eventos	1. El usuario selecciona ver sus valoraciones 2. El usuario puede visualizar los libros que ha valorado
Condiciones de salida	Lista de libros valorados por el usuario

Tabla 3.6.: Caso de uso principal 6. [Elaboración propia]

3.4.8. Caso de uso: Ver la lista de las recomendaciones recibidas

Un usuario podrá ver una lista de los libros que le ha recomendado el sistema hasta el momento, y este caso de uso se detalla en la tabla 3.7.

Caso de uso principal 7	
Nombre	Ver la lista de recomendaciones recibidas
Descripción	El usuario puede visualizar un listado de las

	recomendaciones recibidas en alguna ronda de recomendación
Actor/es participantes	Usuario
Condiciones de entrada	Ninguna
Flujo de eventos	1. El usuario selecciona ver sus recomendaciones 2. El usuario puede visualizar los libros que se le han recomendado
Condiciones de salida	Lista de libros recomendados al usuario

Tabla 3.7.: Caso de uso principal 7. [Elaboración propia]

3.4.9. Caso de uso: Cambiar una valoración hecha a un libro

En este caso de uso, detallado en la tabla 3.8., un usuario podrá cambiar, desde la lista de libros valorados, la valoración hecha a un libro, de modo que la nueva valoración quede registrada en el sistema, sustituyendo a la anterior.

Caso de uso principal 8	
Nombre	Cambiar valoración hecha a un libro
Descripción	Un usuario puede cambiar la valoración que le ha dado a un libro
Actor/es participantes	Usuario
Condiciones de entrada	Ninguna
Flujo de eventos	1. Un usuario selecciona ver sus valoraciones 2. Un usuario selecciona ver los detalles de un libro valorado 3. Un usuario selecciona una valoración para el libro

	4. Un usuario selecciona guardar
Condiciones de salida	La valoración para el libro es modificada en el sistema

Tabla 3.8.: Caso de uso principal. [Elaboración propia]

3.4.10. Caso de uso: Desconectarse del sistema

Un usuario puede solicitar desconectarse del sistema con lo que se cerrará su sesión en el mismo. Los detalles de este caso de uso se pueden ver en la tabla 3.9.

Caso de uso principal 9	
Nombre	Desconectarse del sistema
Descripción	El usuario puede solicitar desconectarse del sistema
Actor/es participantes	Usuario
Condiciones de entrada	Ninguna
Flujo de eventos	1. El usuario despliega el menú de usuario 2. El usuario selecciona desconectar
Condiciones de salida	1. El usuario deja de ser el usuario activo y se abre la página de inicio 2. Se abre la página de inicio

Tabla 3.9.: Caso de uso principal 9. [Elaboración propia]

3.4.11. Caso de uso: Darse de baja del sistema

En este último caso de uso, detallado en la tabla 3.10., un usuario puede darse de baja del sistema, con lo que sus datos, sus recomendaciones y sus valoraciones se eliminarán del sistema.

Caso de uso principal 10	
Nombre	Darse de baja del sistema
Descripción	Un usuario puede darse de baja del sistema
Actor/es participantes	Usuario
Condiciones de entrada	Ninguna
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario despliega el menú de usuario 2. El usuario selecciona baja del sistema
Condiciones de salida	<ol style="list-style-type: none"> 1. El usuario deja de ser el usuario activo y se borra del sistema, así como sus valoraciones y recomendaciones 2. Se abre la página de inicio

Tabla 3.10.: Caso de uso principal 10. [Elaboración propia]

3.5. Requisitos no funcionales

Se ha identificado un sólo requisito no funcional, y es que la interfaz sea sencilla e intuitiva, fácil de utilizar para los usuarios.

3.6. Modelo de dominio

El modelo de dominio, que se muestra en la figura 3.2., representa la relación que se da entre las clases conceptuales del sistema en la fase de análisis de los requisitos, aunque puede ser revisada en etapas posteriores.

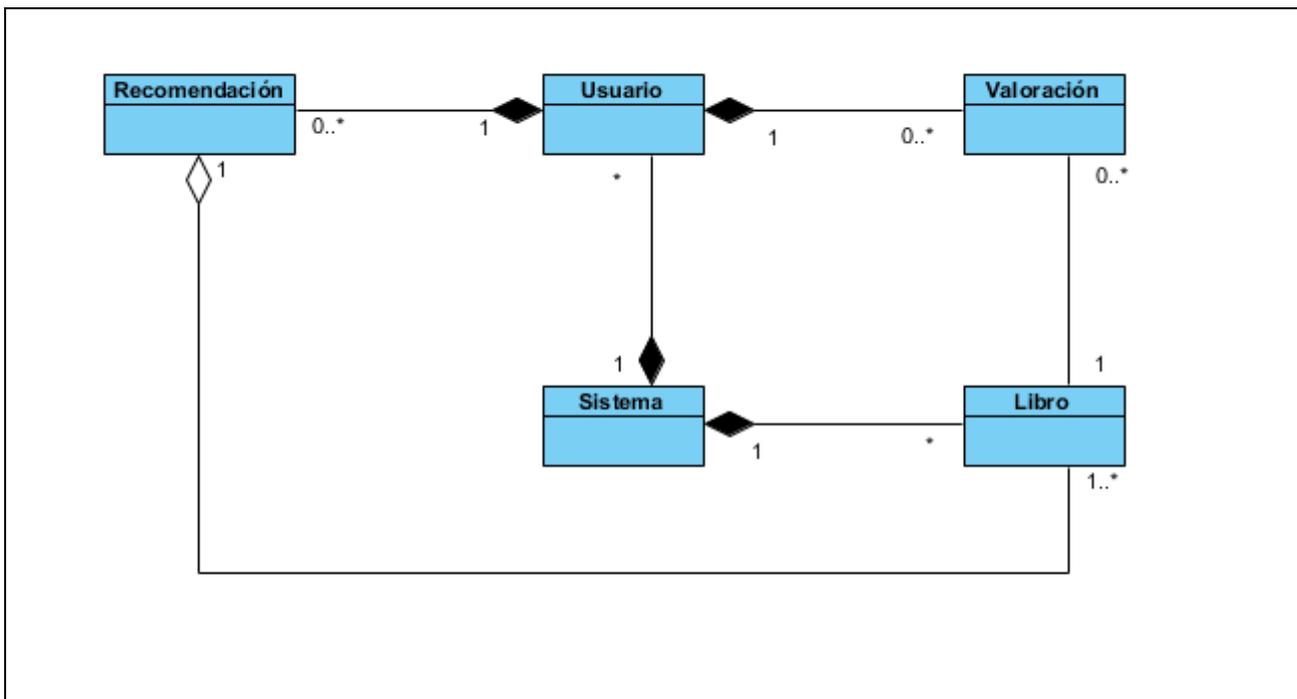


Figura 3.2.: Modelo de dominio del sistema. [Elaboración propia]

3.7. Elección del dataset

Para la realización de este proyecto era necesario escoger un dataset con información sobre libros y valoraciones de usuarios dadas a estos libros. Lo que en principio parecía una tarea sencilla, acabó por no serlo tanto. Si bien hay disponibles muchos datasets sobre otros ítems, como películas o canciones, no son tantos los que recogen información sobre libros.

Se valoraron dos conjuntos de datos: BookCrossing-dataset [30] y goodbooks-10k [31].

El primero es el resultado del trabajo realizado por Cai-NicolasZiegler, recopilando datos de la comunidad de Book-Crossing sobre libros, usuarios y recomendaciones.

El segundo, que finalmente fue el escogido, por contener información más detallada, recoge datos de libros, usuarios, recomendaciones y valoraciones de la web de recomendación de libros Goodreads, comentada en un capítulo anterior, y cuya información fue extraída del formato XML con el que esta página almacena sus datos.

El dataset de Goodreads-10k está dividido en 5 archivos en formato csv: books.csv, book_tags.csv, tags.csv, to_read.csv y ratings.csv.

- books.csv: Contiene datos de 10.000 libros.
- book_tags.csv: Relaciona la id de un libro con la id de una etiqueta.
- tags.csv: Contiene el nombre de etiquetas o géneros literarios y un identificador para cada una de ellas.
- to_read.csv: Contiene recomendaciones de usuarios sobre libros, ambos representados con su correspondiente id.
- ratings.csv: Recoge las valoraciones, representadas con una cifra de 1 a 5, de un usuario a un libro. Tanto los libros como los usuarios están identificados por su id.

Para este proyecto se han utilizado tres de los cinco archivos, prescindiéndose del archivo de recomendaciones (to_read.csv) y del de la relación de etiquetas temáticas de los libros, ya que hacer las recomendaciones de los usuarios es la finalidad de la aplicación de este proyecto y no se utiliza la descripción temática para ofrecer recomendaciones.

3.8. Tecnologías utilizadas

Para este proyecto, se han seleccionado sólo herramientas de código abierto, que no tengan coste adicional para el cliente, y que además son tecnologías que se han utilizado en las prácticas de diferentes asignaturas a lo largo de la carrera.

3.8.1. Spring Framework



Figura 3.3.: Logo de spring.

Spring Framework [34], cuyo logo se muestra en la figura 3.3., es un entorno de desarrollo open source pensado para trabajar con Java, que tiene herramientas y utilidades que ayudan al desarrollador a crear aplicaciones web, especialmente pensadas para la parte del back-end de la aplicación.

3.8.2. HTML5



Figura 3.4.: Logo de HTML5

HTML5, cuyo logo se puede ver en la figura 3.4., es la quinta versión de HTML (Hyper Text Markup Language), que es el lenguaje utilizado para la construcción de las páginas web. Es un estándar a cargo del World Wide Web Consortium (W3C), que ha sido adaptado por todos los navegadores para la

visualización de las páginas web.

La mayor novedad que aporta HTML5, si se compara con versiones anteriores, es que se han añadido nuevas etiquetas para organizar el código, como son <div>, <footer>, <nav> y otras como , <audio> o <video>, para incluir elementos específicos.

La correcta visualización de estas etiquetas requiere el uso de un navegador actualizado ya que las nuevas etiquetas no son compatibles con anteriores versiones de HTML.

3.8.3. Bootstrap



Figura 3.5.: Logo de Bootstrap.

Bootstrap [32], cuyo logo se puede ver en la figura 3.5., es una biblioteca de código abierto, que permite utilizar sus componentes de forma sencilla para aplicar estilos y objetos para construir páginas web, utilizando CSS y HTML, centrándose en la parte del front-end de las aplicaciones.

3.8.4. Entorno de desarrollo: Netbeans 8.2.

Netbeans [35], cuyo logo se puede ver en la figura 3.6., es un entorno de desarrollo integrado, gratuito y de código abierto, que permite programar aplicaciones en los sistemas operativos Windows, Mac, Linux y Solaris. El IDE simplifica el desarrollo de aplicaciones web, corporativas, de escritorio y móviles que utilizan plataformas Java y HTML5 y ofrece soporte para el desarrollo de aplicaciones PHP y C/C++ [33].

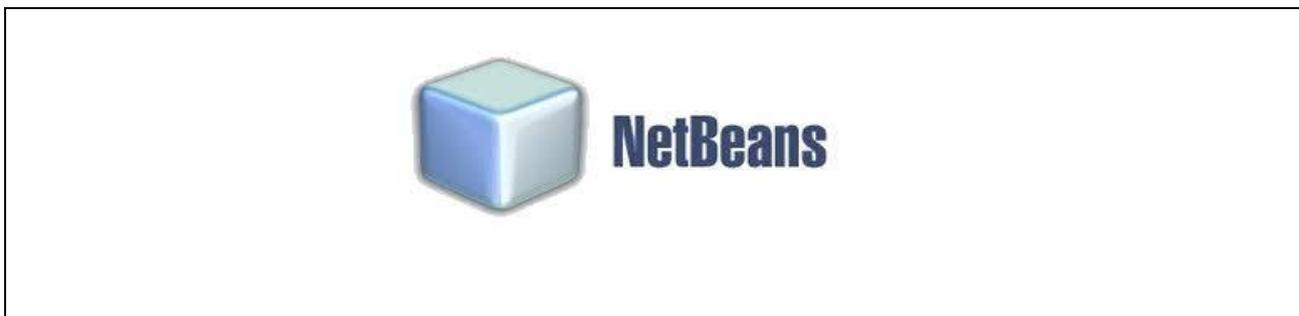


Figura 3.6.: Logo de Netbeans

3.9. Metodología de trabajo escogida: Scrum

A la hora de escoger el método de trabajo para llevar a cabo este proyecto, se optó por una metodología ágil [7,8]. Una metodología ágil de desarrollo de software se caracteriza por su flexibilidad, ya que se adaptan a los cambios que puedan producirse en cada revisión del proyecto y por tratar cada parte del mismo como si fuera independiente. Se producen entregas frecuentes, cada dos o seis semanas y tras cada entrega se produce una retroalimentación ya que el equipo decide qué procesos están funcionando adecuadamente y cuáles se pueden mejorar.

Scrum [7,8,21] es la metodología ágil más popular. No es exclusiva de proyectos software, sino que puede aplicarse a toda clase de proyectos, ya que sirve para crear un entorno de gestión de procesos. En la Guía de Scrum [22], se define esta metodología como un marco de trabajo a través del cual las personas pueden abordar problemas complejos adaptativos, a la vez que se entregan productos de forma eficiente y creativa con el máximo valor. Se fundamenta en:

- Trabajo en equipos pequeños
- Uso de roles en el equipo, que confieren a quien los desempeña unas responsabilidades determinadas
- Eventos, en los que se revisa y adapta el trabajo que se ha realizado hasta el momento
- Utilidades o artifacts, que permiten que la información sea compartida por todos los miembros del equipo

- Unas reglas sencillas, sobre cómo desarrollar los eventos, las funciones de cada rol y el uso de las utilidades.

Los roles en un equipo Scrum son 3: Product owner, Scrum master y equipo de desarrollo. El Product owner es el que se relaciona con el cliente y traslada sus necesidades al backlog del producto, del que es el único responsable.

El backlog del producto (PB) es una lista ordenada de tareas pendientes a realizar para un producto. El PB puede cambiar a lo largo del tiempo, ya que puede ser refinado, se pueden añadir o quitar elementos.

Los elementos del PB son las historias de usuario (HU), que tienen un título, una descripción, un valor en puntos de historia (PH) y unos criterios de validación. Además, si una HU es compleja, se dividirá en tareas.

Los miembros del equipo de desarrollo, formado por 3-9 personas, son los únicos que trabajan en el producto. Tienen la capacidad de autogestionarse y es un equipo multifuncional, saben hacer un poco de todo lo necesario para el proyecto. Además, todos los miembros del equipo tienen el mismo rango y los equipos no se subdividen.

El Scrum master, facilita la aplicación de las reglas de Scrum dentro del equipo y de la empresa. Organiza los eventos Scrum, ayuda a solucionar problemas que puedan surgir y ayuda al Product owner con la gestión del PB.

Respecto a los eventos Scrum, son 5: Sprint, Sprint Planning, Daily scrum, Sprint review y Sprint retrospective.

- Sprint: Tiene una duración de más o menos un mes y cuando finaliza se tiene un incremento usable.
- Sprint planning: El objetivo es decidir qué se va a entregar al final del Sprint y cómo se va a trabajar durante el mismo. Se seleccionan las tareas del PB que van a pasar al Sprint backlog (SB), que sólo puede ser modificado por el equipo de desarrollo.
- Daily scrum: Es una reunión diaria, de unos 15 minutos, que el equipo mantiene para que el equipo se mantenga informado de cómo se va avanzando en el proyecto. Cada miembro expone qué hizo ayer, qué

problemas ha encontrado y qué va a hacer hoy.

- **Sprint review:** El equipo, junto a personas interesadas, como el cliente, hace una revisión del incremento generado en el Sprint y se adapta el PB si es necesario, por ejemplo, se puede añadir una historia de usuario que no se había considerado que fuera necesaria hasta ese momento, o, por el contrario, eliminar una historia.
- **Sprint retrospective:** El equipo reflexiona sobre cómo ha trabajado en el último Sprint y qué se puede hacer para mejorar.

En la figura 3.7. se muestra un esquema de cómo funciona el proceso de realización de los proyectos elaborados usando Scrum.

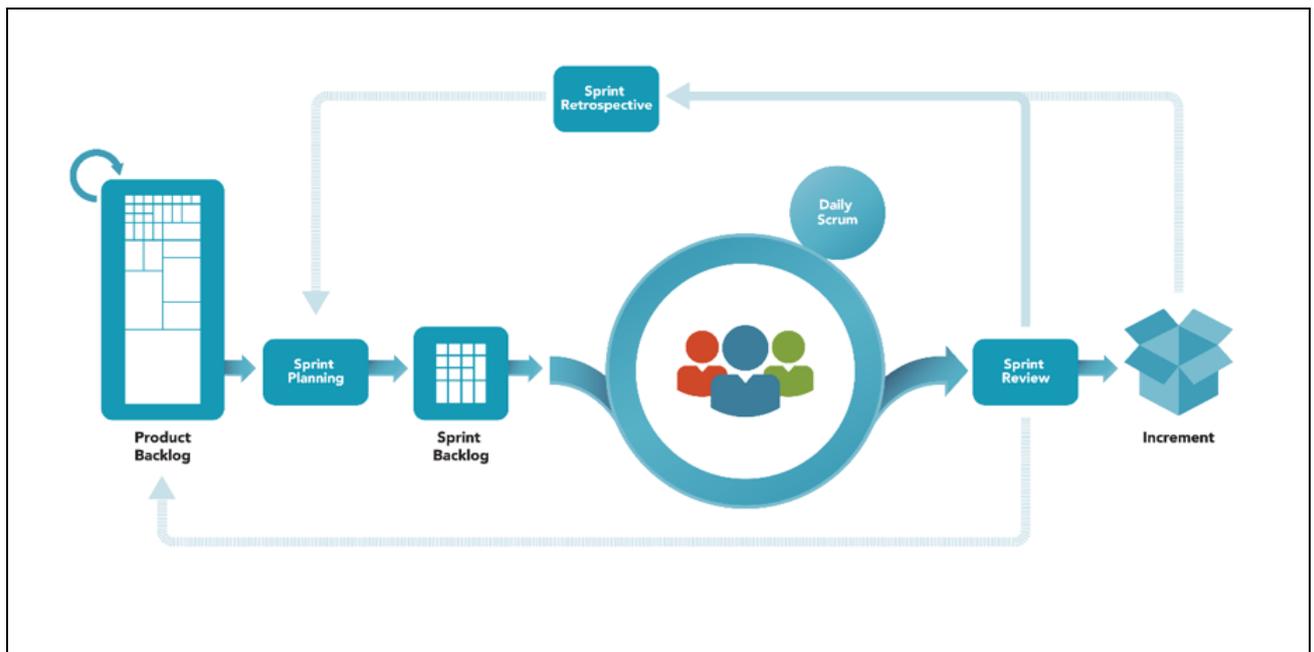


Figura 3.7.: Proceso de Scrum.

4. APLICACIÓN DE LA METODOLOGÍA SCRUM AL PROYECTO

En esta sección se explica el proceso de aplicación de la metodología ágil Scrum al proyecto. En ella se explica el reparto de roles y cómo se han planteado el Product backlog del proyecto, el presupuesto, la planificación y el desarrollo de las diferentes iteraciones.

4.1. Reparto de roles

Lo primero que habría que hacer es el reparto de roles para el proyecto. En este caso, los tutores, Doña Carmen Martínez Cruz y Don Carlos Porcel Gallego, actuarían como Product owners y como Scrum masters. Mientras que el equipo de desarrollo está formado únicamente por mí.

4.2. Product backlog (PB) del proyecto

A continuación, habría que detallar las historias de usuario que componen el Product backlog del proyecto y expresar su dificultad asignándoles una cantidad de puntos de historia.

Para asignar los puntos de historia, hay que utilizar una de las técnicas que se utilizan para ello, en este caso se ha seleccionado la progresión de Fibonacci (1, 2, 3, 5, 8, etc.), para asignarlos.

Se ha dividido el backlog en dos partes. La primera corresponde a la aplicación web y a la lógica necesaria para su funcionamiento, como muestra la tabla 4.1. La segunda se refiere a la preparación del documento final y a la presentación que lo resume, como muestra la tabla 4.2.

Web	
Historia de usuario	Puntos de Historia
Registro de usuario	8
Acceso a la aplicación	8
Ver lista de libros valorados	5
Ver lista de recomendaciones	5
Recibir recomendación	13
Editar perfil	5
Cambiar valoración de un libro	8
Darse de baja	8

Desconectarse de la aplicación	3
Diseñar capa de persistencia	13
Diseñar modelo de datos	13
Página de portada	5
Página principal	5
Página de registro	5
Página de perfil de usuario	5
Página de recomendaciones	5
Página de valoraciones	5
Menú de navegación	3

Tabla 4.1.: Historias de usuario relacionadas con la web. [Elaboración propia]

Documentación	
Historia de usuario	Puntos de Historia
Documento TFG	55
Presentación	8

Tabla 4.2.: Historias de usuario relacionadas con la documentación del proyecto.

4.3. Planificación de las iteraciones

Con esta asignación de puntos de historia (PH), se tienen 106 puntos de historia para la primera tabla y 63 para la segunda, lo que suma un total de 169 puntos de historia.

La velocidad de un Sprint es el número de PH que se consiguen terminar en un Sprint. El Proyecto está pensado para realizarlo durante el segundo cuatrimestre del curso 2019-2020, para terminarlo en mayo. Por tanto, si se comenzó a realizarlo en febrero y se quiere terminar en mayo, se disponen de 4 meses para terminarlo. Si se trabaja con sprints de un mes, al dividir 169 entre 4 nos da 42,25 PH por Sprint, 43 PH si se redondea hacia arriba. Los 4 sprints o

Iteraciones tendrían las fechas de comienzo y fin que se muestra en la tabla 4.3:

Sprint	Fechas de comienzo y fin
Sprint 1	1/02/2020-29/02/2020
Sprint 2	1/03/2020-31/03/2020
Sprint 3	1/04/2020-30/04/2020
Sprint 4	1/05/2020-31/05/2020

Tabla 4.3.: Planificación de inicio-fin de los sprints del proyecto. [Elaboración propia]

4.4. Presupuesto

Para calcular el presupuesto inicial, hay que determinar qué días se consideran laborables y cuántas horas de esos días forman la jornada laboral.

Para este proyecto, no se van a considerar días laborables los sábados ni los domingos, por lo que en un mes, hay 20 días laborables y se van a considerar jornadas de 8 horas, sumando 160 horas por iteración, con un total de 640 horas para las 4 iteraciones.

Por lo tanto, para calcular las horas que se van a emplear en conseguir realizar cada punto de historia, hay que dividir las 640 horas entre los 169 puntos de historia, lo que da 3,78 horas por cada punto de historia.

El proyecto requiere la intervención de una sola persona, que debido a su categoría de analista/programador, cobrará un sueldo de 28.500 euros al año. Toda esta información se recoge en la tabla 4.4.

Analista/programador	Precio
Sueldo anual	28.500,00 €
Sueldo mensual	2.375,00 €
Sueldo diario	118,75 €
Sueldo por hora	14,84 €

Tabla 4.4.: Resumen de sueldos de un analista trabajador. [Elaboración propia]

Al cliente se le van a facturar 640 horas por lo que el precio total por este concepto es de 9.497,6 €.

A esta cifra habría que sumar el coste de la parte correspondiente al equipo informático necesario para realizar el trabajo. Como se ha visto en la asignatura de Auditoría Informática, si quien va a realizar un trabajo necesita adquirir algún equipo para realizarlo durante 4 meses y el equipo tiene una vida útil de 5 años, se puede cobrar a la empresa la parte equivalente.

Para realizar el proyecto necesitamos un equipo informático, así que se ha seleccionado un ordenador portátil Acer Aspire 5, con las características que se muestran en la tabla 4.5.

Equipo	Acer Aspire 5
Características	15.6" Full HD LED (Intel Core i7-8565U, 8 GB de RAM, 128 GB SSD + 1 TB HDD, Tarjeta gráfica Nvidia MX130 2 GB, Windows 10 Home) Silver - Teclado QWERTY Español
Precio	699,00 €

Tabla 4.5.: Características del equipo informático. [Elaboración propia]

Teniendo en cuenta que los equipos tienen una vida útil de 5 a 7 años, de la que se va a tomar la inferior para hacer los cálculos, tendríamos que pagar 140 euros por año por el equipo y los 4 meses empleados en el proyecto equivaldrían 46,66 euros, que se pueden añadir a la factura del cliente, como se muestra en la tabla 4.6.

Coste Acer	Vida útil	Coste por año	Duración en	Cifra
-------------------	------------------	----------------------	--------------------	--------------

Aspire 5			meses del proyecto	amortizable
699,00 €	5 años	140,00 €	4	46,66 €

Tabla 4.6.: Cálculo de la amortización del equipo informático. [Elaboración propia]

Respecto al software, las herramientas utilizadas no suponen ningún coste adicional, por lo que los costes del proyecto serían los que se muestran en la tabla 4.7. La herramienta utilizada para desarrollar la aplicación, Netbeans 8.2., es de código abierto y se puede descargar directamente desde la página. También son de código abierto Bootstrap y Spring Framework.

Costes del proyecto	
Sueldo del analista/programador durante el proyecto	9.497,60 €
Amortización del equipo informático	46,66 €
Total costes	9.544,26 €

Tabla 4.7.: Costes del proyecto. [Elaboración propia]

A los conceptos anteriores habrá que añadir una parte de beneficio, que puede oscilar entre 20-50% del coste, en este caso se va a coger un 30%, y un 21% de IVA. El presupuesto final es el que se recoge en la tabla 4.8.

Presupuesto	
Base imponible (costes)	9.544,26 €
30% de beneficio sobre los costes	2.863,27 €
21% de IVA	2.004,29 €

Desarrollar un sistema de recomendaciones de libros basado en memoria

Total costes	14.411,82 €
--------------	-------------

Tabla 4.8.: Presupuesto total del proyecto. [Elaboración propia]

4.5. Iteraciones

En esta sección, se analizan las historias de usuario que se han escogido para cada uno de los Sprints, así como los problemas que se han presentado a lo largo de su desarrollo y si ha surgido algún cambio durante el mismo.

Para cada una, se presentan gráficas sobre su evolución, que permiten ver con claridad si se han sufrido retrasos o todo ha ido según lo planeado. Además, al principio de las iteraciones que lo permiten, se comentarán los cambios que se ha considerado introducir en el Product backlog, como resultado del Sprint retrospective.

4.5.1. Iteración 1

En la iteración 1, se incluyen las siguientes tareas:

1. Diseñar capa de persistencia: Una vez que se ha decidido el modelo de datos, se tiene que analizar cómo se van a guardar estos datos.
2. Diseñar modelo de datos: El diseño del modelo de datos es una tarea intelectual, en la que se decide qué información necesita manejar el sistema y cómo se va a codificar esa información.
3. Página principal: La página principal o de inicio es la página que se abre cuando el usuario accede a la web. Al usuario se le presentan dos opciones, la de acceder a la aplicación y la de registro de usuario. Si el usuario ya se ha registrado puede acceder a la aplicación con su nombre de usuario y contraseña. Si no lo ha hecho todavía, ha de registrarse para obtener esas credenciales.
4. Página de registro: Esta página permite el registro de un nuevo usuario, proporcionando un nombre de usuario y una contraseña. El nombre de usuario no podrá estar repetido, de forma que no se creará un nuevo usuario si el nombre está siendo utilizado por otro usuario del sistema.
5. Menú de navegación: El menú de navegación permite que el usuario, una vez que se ha identificado y haya accedido a la página principal, se

desplace entre las distintas páginas de la web. El menú debe ser intuitivo y fácil de entender para el usuario, que, con un sólo vistazo, ha de comprender cómo se organiza la página y qué opciones de interacción tiene en ella. El menú estará presente, además, en las demás páginas, lo que le da un aspecto de uniformidad.

6. Página de portada: La portada es la página a la que accede el usuario una vez que ha introducido sus credenciales, nombre de usuario y password proporcionados durante el registro. En esta página, a través del menú de navegación, se podrá acceder a las distintas páginas que componen la web.

En la tabla 4.9. muestra el Sprint backlog de la iteración 1.

Historia de usuario	PH
Diseñar capa de persistencia	13
Diseñar modelo de datos	13
Página principal	5
Página de registro	5
Menú de navegación	3
Página de portada	5
Total	44

Tabla 4.9.: Sprint backlog de la Iteración 1. [Elaboración propia]

En este primer sprint, se ha planificado hacer las páginas principales de la aplicación así como el menú de navegación común a todas las páginas. Además, se incluye el diseño de datos y el de la persistencia, fundamentales a la hora de

decidir qué datos se van a manejar en el programa y cómo se van a estructurar y almacenarlos de forma permanente en una base de datos.

4.5.2. Iteración 2

En esta iteración, se han incluido las historias de usuario siguientes:

1. Registro de usuario: El registro de usuario, es la función que hace que los datos de usuario que se introducen en la página de registro, se usen para crear un nuevo usuario con esas credenciales.
2. Página de perfil de usuario: El usuario puede ver su perfil y cambiar alguna de sus opciones relativas a preferencias desde esta página, número de recomendaciones que desea recibir, tipo de recomendaciones.
3. Página de recomendaciones: A través de esta página el usuario puede ver las recomendaciones personalizadas que la aplicación ha seleccionado para él.
4. Página de valoraciones: El usuario podrá visualizar las valoraciones que ha dado a diferentes libros, ya sea sus valoraciones iniciales, dadas la primera vez que seleccionó recibir una recomendación o las dadas posteriormente.
5. Acceso a la aplicación: Esta función permite que el usuario acceda a la aplicación si introduce unas credenciales válidas, que correspondan a un usuario registrado en el sistema.
6. Darse de baja: Esta función hace que el usuario deje de formar parte del sistema, eliminándolo de la base de datos, así como sus valoraciones y sus recomendaciones.
7. Editar perfil: Esta función permite que los cambios que se han introducido en la página de perfil se cambien en la tabla de usuarios de la base de datos.

En la tabla 4.10. se puede ver el Sprint backlog de la iteración 2.

Historia de usuario	PH
---------------------	----

Registro de usuario	8
Página de perfil de usuario	5
Página de recomendaciones	5
Página de valoraciones	5
Acceso a la aplicación	8
Darse de baja	8
Editar perfil	5
Total	44

Tabla 4.10.: Sprint backlog de la Iteración 2. [Elaboración propia]

En este sprint se ha decidido dar prioridad a las historias de usuario que están relacionadas con la parte lógica de la aplicación, es decir, a las funciones y controladores que van a hacer que las páginas que se han creado en la iteración anterior. Además, se va a terminar un segundo bloque de páginas.

4.5.3. Iteración 3

En esta iteración se incluyen las tareas:

1. Ver lista de libros valorados: La función permite seleccionar la lista de libros que ha valorado un usuario en el sistema.
2. Ver lista de recomendaciones: La función permite seleccionar la lista de recomendaciones que el sistema le ha hecho al usuario en alguna ronda de recomendación.
3. Recibir recomendación: Cuando el usuario selecciona por primera vez esta opción, ha de valorar una serie de títulos, los veinte más recomendados, de modo que el sistema tenga una base para poder ofrecerle recomendaciones. Además, tendrá que decidir qué tipo de recomendaciones y cuántas le gustaría recibir en una ronda de recomendación. Si no es la primera vez que selecciona esta opción, el sistema utilizará la información del perfil del usuario para ofrecerle una

recomendación, que el usuario podrá valorar, comunicando al sistema si le parece o no adecuada para él, de forma que las recomendaciones futuras se ajusten más a sus preferencias.

4. Cambiar valoración de un libro: Un usuario puede cambiar la valoración que ha hecho de un libro seleccionando la opción "ver" desde la lista de libros valorados.
5. Desconectarse de la aplicación: Esta función permite que el usuario deje de ser el usuario activo y sale de la aplicación, abriendo la página principal.

Este sprint tiene la peculiaridad de que se queda corto respecto a la planificación de puntos de historia por sprint. La explicación está en que de las dos tareas restantes, una tiene muchos PH y la otra depende de la anterior. Por lo que se ha decidido que la iteración 4 pueda abarcar más puntos de historia, mientras que la 3 incluye menos. Su Sprint backlog se muestra en la tabla 4.11.

Historia de usuario	PH
Ver lista de libros valorados	5
Ver lista de recomendaciones	5
Recibir recomendación	13
Cambiar valoración de un libro	8
Desconectarse de la aplicación	3
Total	34

Tabla 4.11.: Sprint backlog de la Iteración 3. [Elaboración propia]

4.5.4. Iteración 4

En esta iteración se incluyen las siguientes tareas:

1. Documento TFG: En este documento se plasma la planificación del proyecto y se documentan los pasos dados a lo largo de su elaboración,

así como las conclusiones extraídas del trabajo realizado y la propuesta de líneas de trabajo futuras a partir del mismo.

2. **Presentación:** La presentación se utiliza para ofrecer al tribunal de valoración un resumen del trabajo realizado durante el proyecto del TFG.

El Sprint backlog de la iteración 4 se muestra en la tabla 4.12.:

Historia de usuario	PH
Documento TFG	55
Presentación	8
Total	63

Tabla 4.12.: Sprint backlog de la Iteración 4. [Elaboración propia]

Este último sprint está dedicado a la elaboración del documento final así como de la preparación de la presentación de cara a su exposición. Como es lógico, la segunda tarea depende de la primera, por lo que deberá acabarse antes de poder empezar con su elaboración.

4.5.5. Gráficos de la evolución de las iteraciones y del proyecto

A continuación, se presentan los gráficos que muestran la evolución de las diferentes iteraciones del proyecto, que en general ha ido como se planeó. Además, se muestra una gráfica que representa la distribución en el tiempo del desarrollo del proyecto. En todas las gráficas de las iteraciones, se muestran la fecha de comienzo y fin de cada proceso y los puntos de historia que se han desarrollado, mientras que en la del proyecto, se ha consignado la iteración y no la fecha.

4.5.5.1. Gráfico de la evolución de la iteración 1

Para este mes, al ser el más corto se tuvo que adaptar la equivalencia entre PH y días, al contar además con un día festivo. Por lo tanto, en vez de utilizar 2PH por día, como en el resto de los meses, hubo que hacer algo más de

trabajo para poder acabarlo en los días laborables disponibles. En el gráfico 4.1. se muestra como fue la evolución del proyecto.

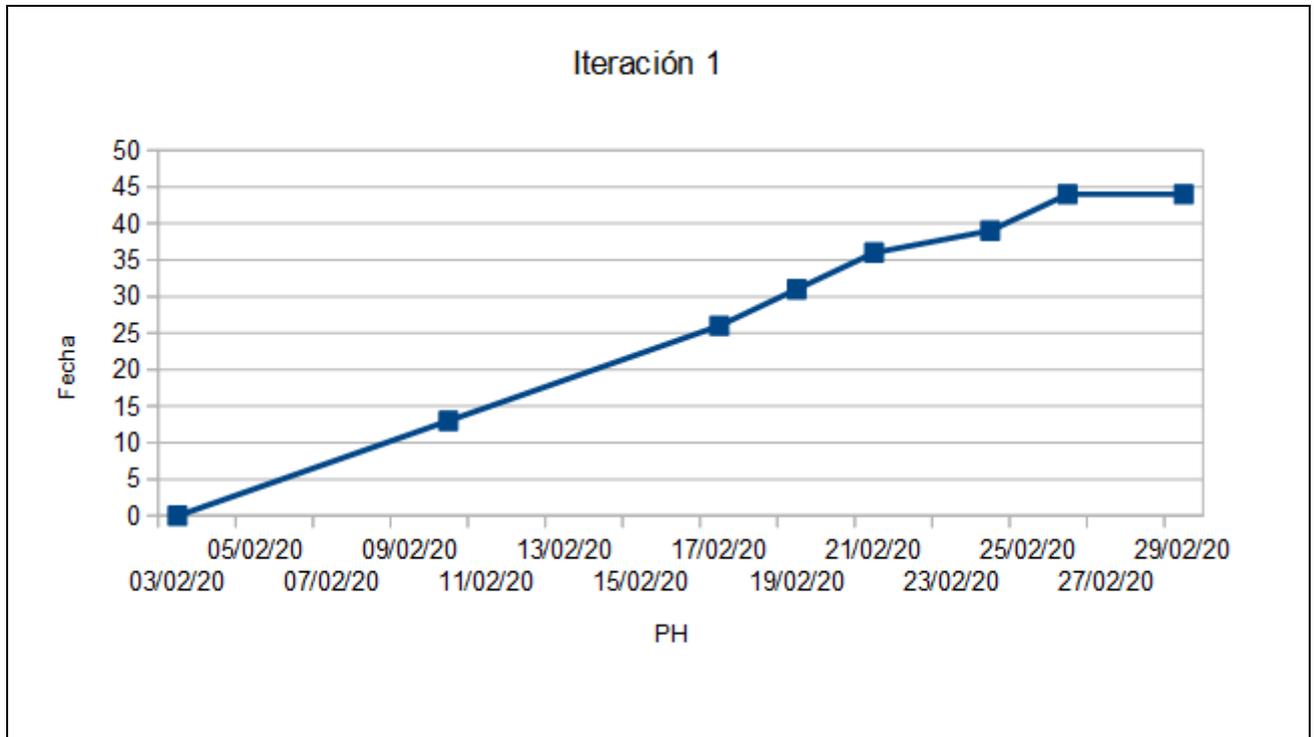


Figura 4.1.: Gráfico de la evolución de la iteración 1. [Elaboración propia]

Se comenzó a trabajar el día 3 ya que los días 1 y 2 eran festivos, al comenzar el mes en fin de semana. Los primeros 13 PH se acabaron el día 10, el mismo día en el que se comenzó a trabajar en los siguientes 13 PH, ya que sobraba algo de tiempo que tenía que ser aprovechado, acabándose con esta tarea el día 17. Las siguientes tareas, en las que había que trabajar sobre tareas que tenían 5 PH cada una, se acabaron los días 19 y 21 respectivamente, y la siguiente, que tenía 3 PH pudo acabarse en un sólo día. Después el trabajo transcurrió sin incidencias, terminando la última historia de usuario de 5 PH, según lo planeado e incluso se acabó de hacer todo lo planificado un día antes de tiempo, el día 27.

4.5.5.2. Gráfico de la evolución de la iteración 2

Para este mes y los siguientes se volvió a utilizar la equivalencia general

para la realización del proyecto, esto es de 2 PH por día. Se empezó a trabajar en esta iteración el día 2 de marzo, al ser festivo el día 1 y se terminó el día 31, como se muestra en el gráfico 4.2. Se terminaron historias de usuario los días 5, 10, 12, 17, 23, 27 y 31.

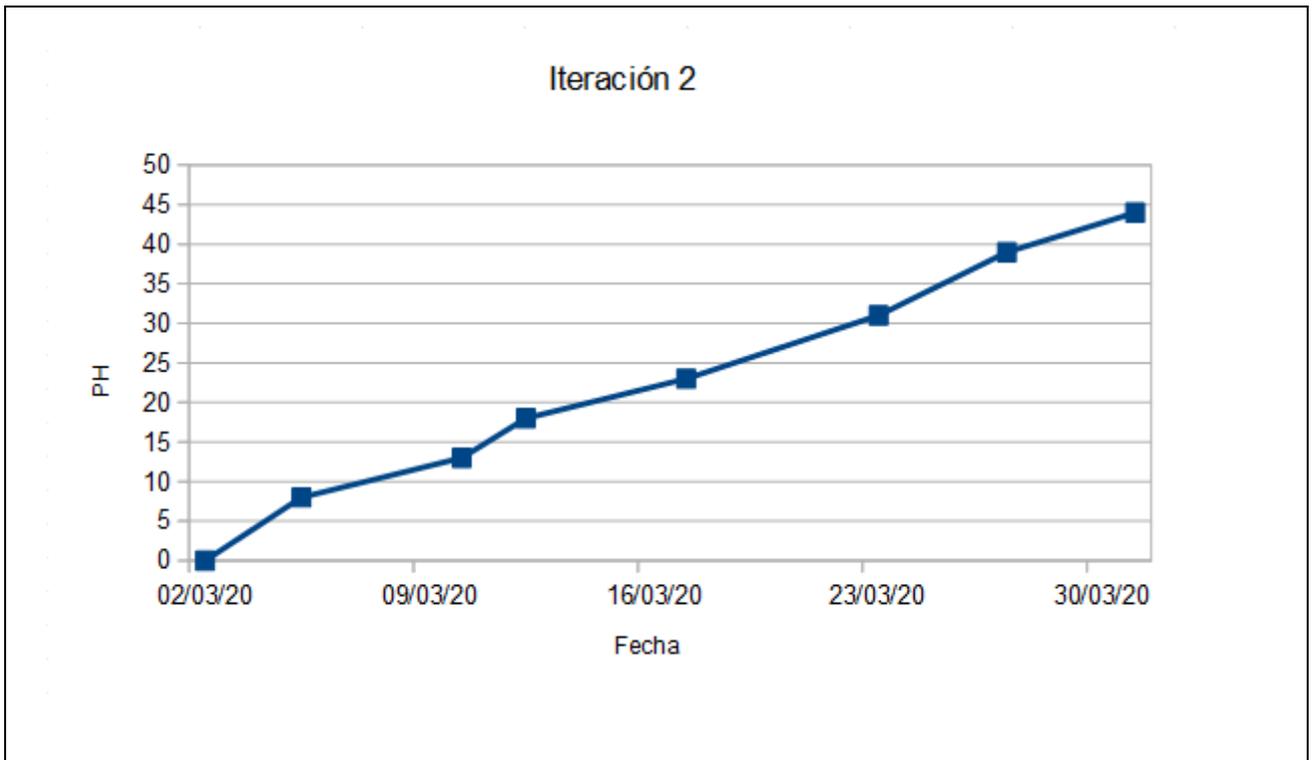


Figura 4.2.: Gráfico de la evolución de la iteración 2. [Elaboración propia]

4.5.5.3. Gráfico de la evolución de la iteración 3

El gráfico 4.3. muestra la evolución de la iteración 3, en la que empezó a trabajar el día 1 de abril y se terminaron historias de usuario los días 3, 7, 16, 22 y 24. Como ya se indicó, esta iteración era algo más corta, contando con tan sólo 34 PH, por lo que se pudo empezar a trabajar en la siguiente iteración.

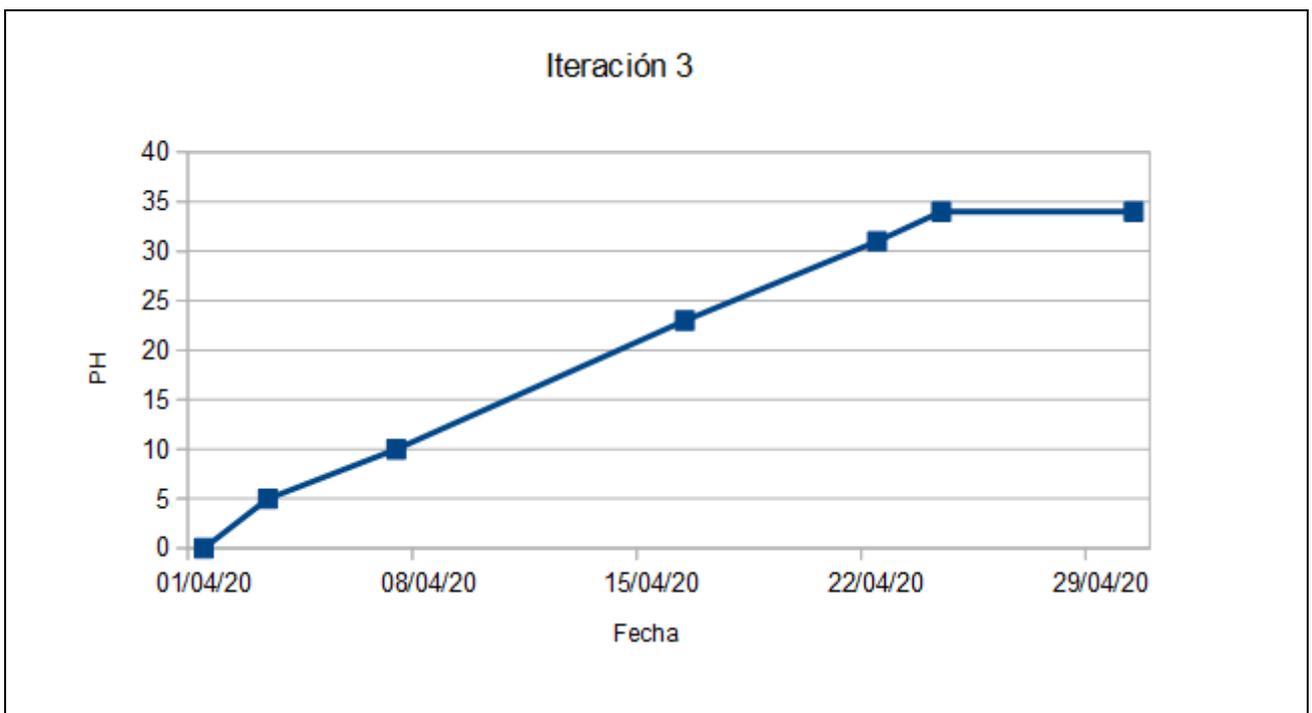


Figura 4.3.: Gráfico de la evolución de la iteración 3. [Elaboración propia]

4.5.5.4. Gráficos de la evolución de la iteración 4

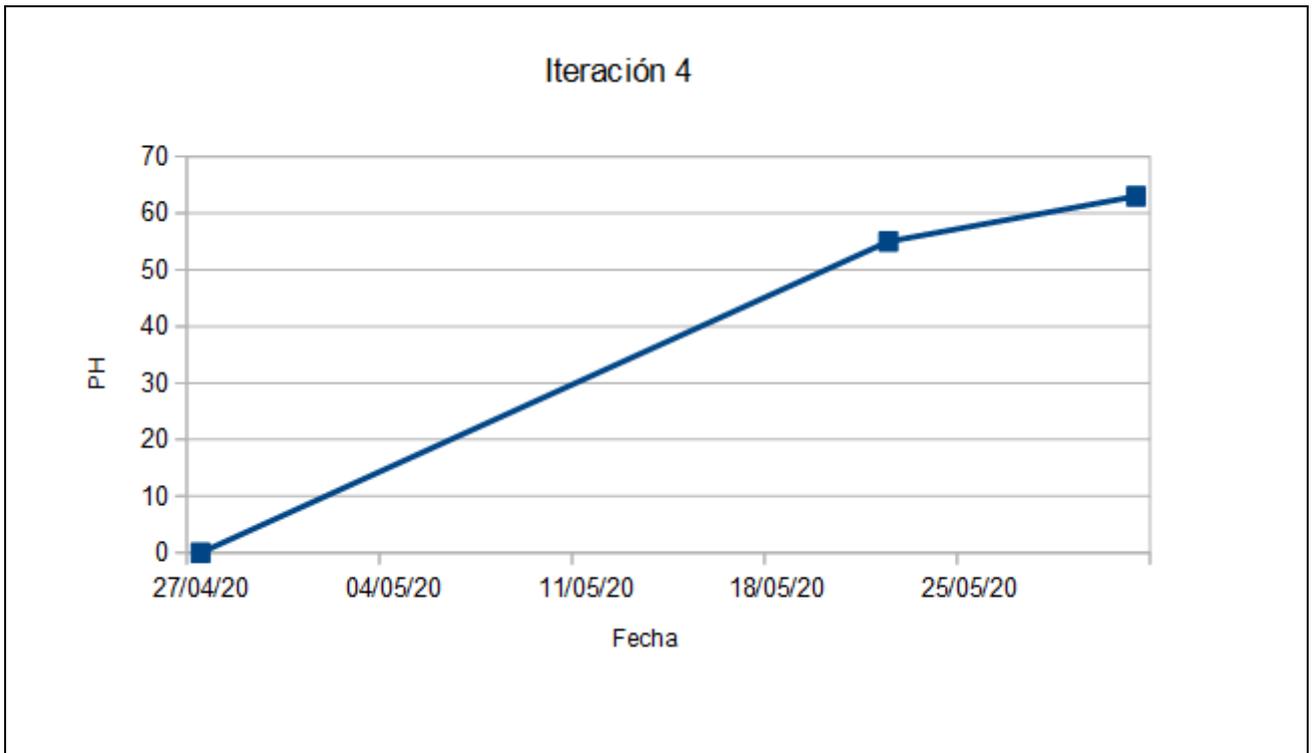


Figura 4.4.: Gráfico de la evolución de la iteración 4. [Elaboración propia]

Finalmente, se dedicó la última iteración a plasmar en la memoria todo el trabajo que se había ido haciendo para el proyecto, así como a modificarla según se iban sucediendo las diversas correcciones. Esto ocuparía gran parte de la iteración, reservándose tan sólo una pequeña parte a la elaboración de la presentación del trabajo realizado. La menor duración de la iteración anterior, hizo que se pudiera empezar a trabajar antes en la nueva iteración. El gráfico de la evolución de la iteración se muestra en la figura 4.4.

4.5.5.5. Gráfico de la evolución del proyecto

Como se puede observar en la figura 4.5., la evolución del proyecto fue avanzando sin mayores incidencias y de modo constante, debido a la igualdad de puntos de historia realizado en cada una de las iteraciones, hasta completar la totalidad del proyecto.

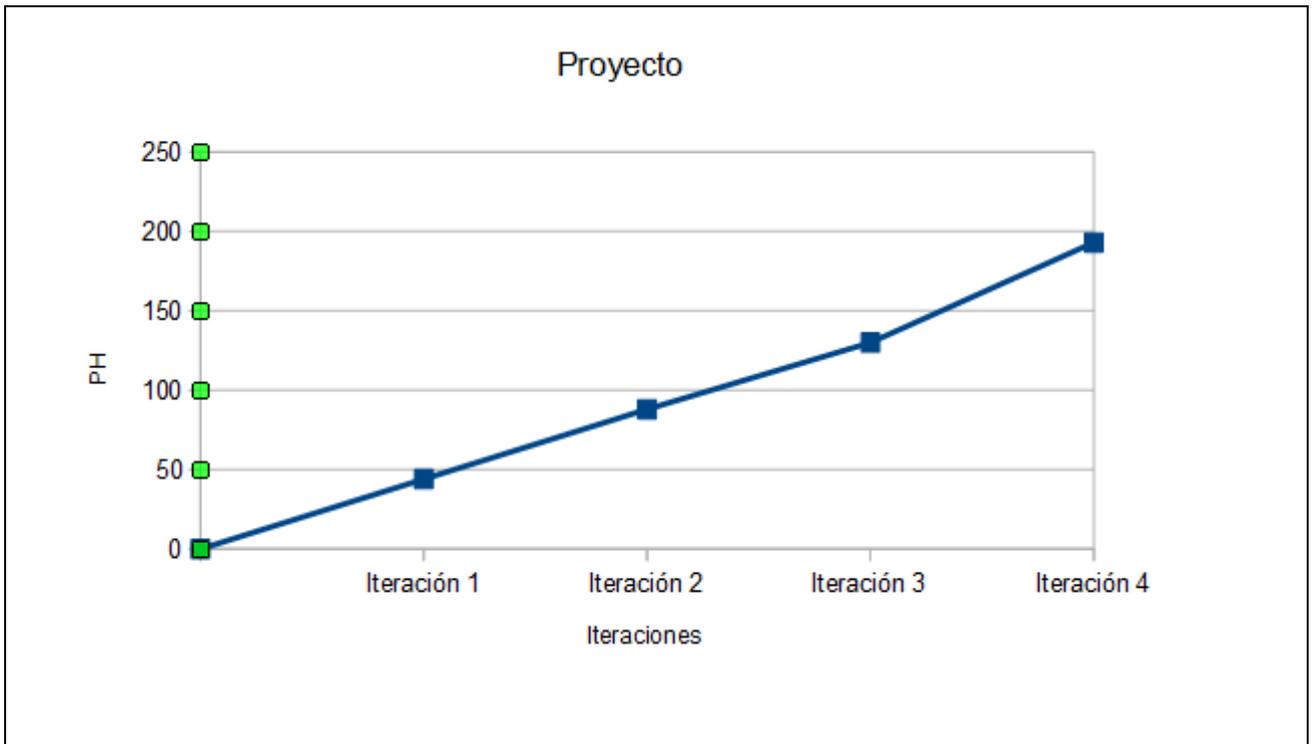


Figura 4.5.: Gráfico de la evolución del proyecto. [Elaboración propia]

5. IMPLEMENTACIÓN DE MYRECOMMENDER

A la hora de elaborar la aplicación, se ha buscado que sea sencilla, fácil de utilizar y que el usuario pueda comprender de un solo vistazo cómo se organiza y las diferentes opciones que tiene para interactuar con ella. En este capítulo, se recoge una pequeña guía de la aplicación que se apoya en imágenes para mostrar su funcionamiento.

5.1. Página de inicio

Es la página que se abre al iniciar la aplicación. Como se muestra en la figura 5.1., en esta página el usuario tiene la opción de pulsar sobre dos botones "Entrar" y "Registro." El primer botón le lleva hasta la página de login, mientras que el segundo abre la página de registro. Además, el usuario puede ver el nombre de la aplicación y su funcionalidad. Se ha añadido también la fecha y la hora de acceso a la aplicación.

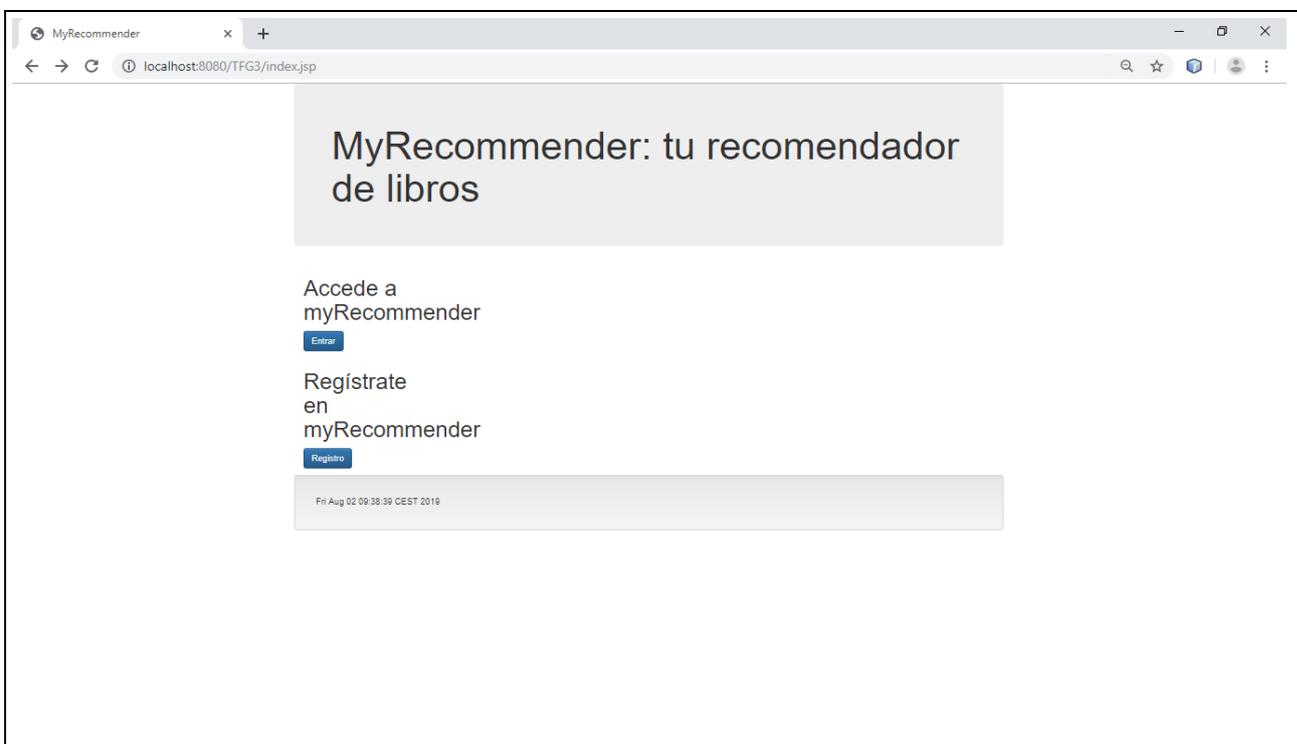


Figura 5.1.: Página de inicio. [Autora]

5.2. Registro de usuario

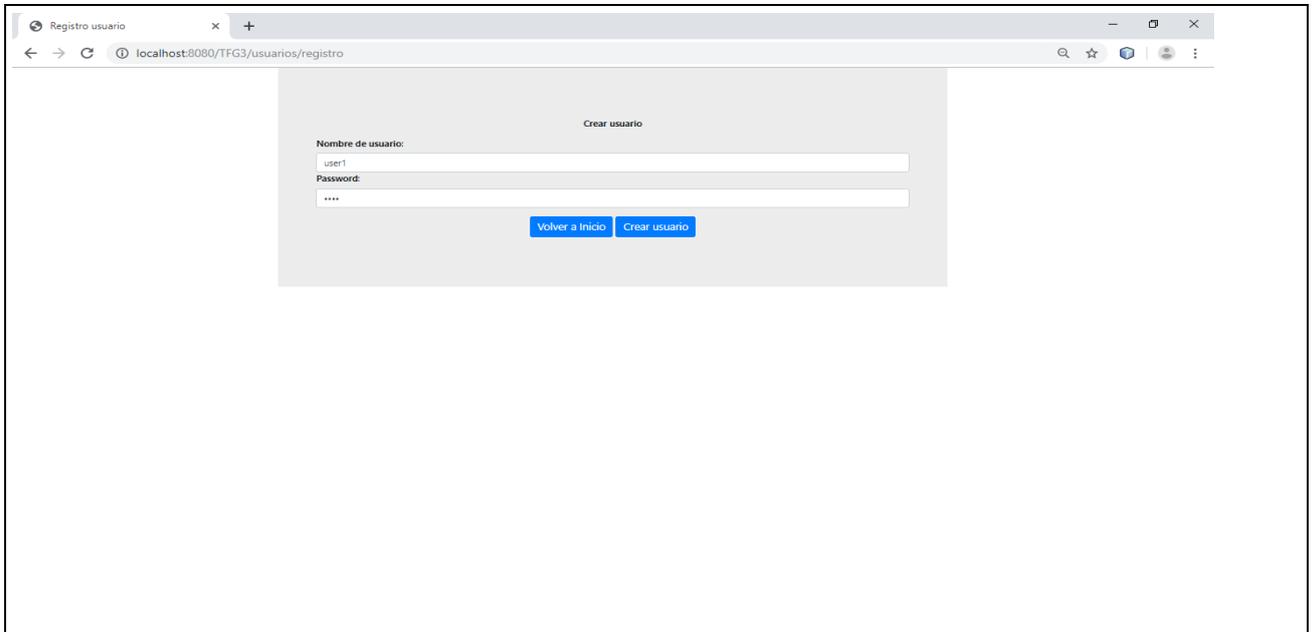
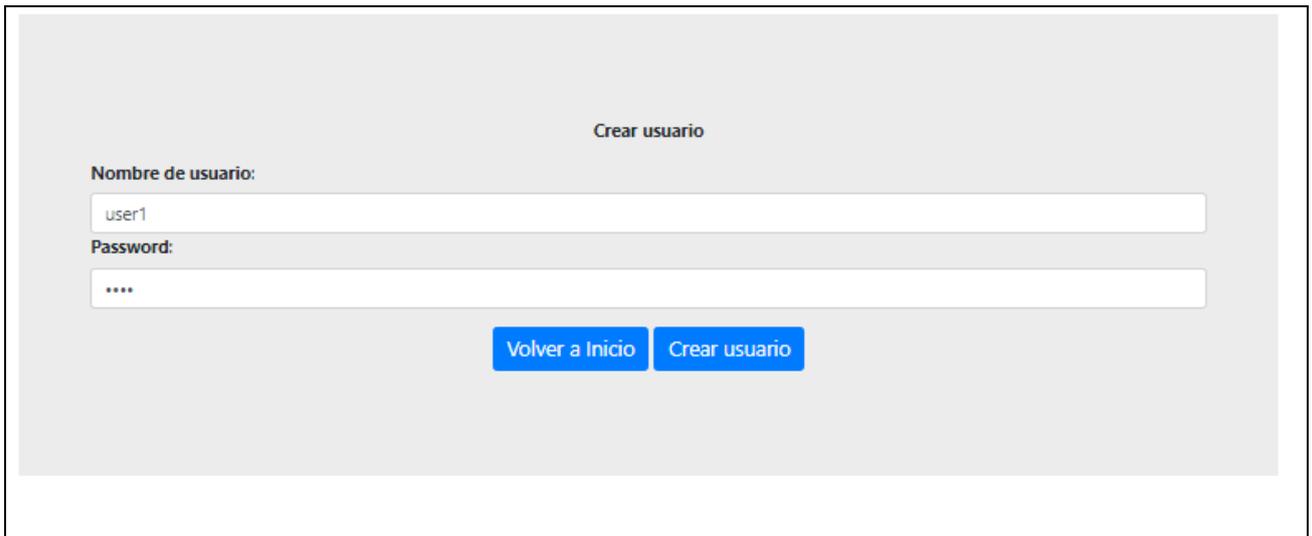


Figura 5.2.: Página de registro. [Autora]

En esta página, como muestran las figuras 5.2. y 5.3., hay un formulario con dos campos, nombre de usuario y password, y dos botones "Volver a inicio" y "Crear usuario". Para crear un nuevo usuario, el usuario tiene que proporcionar un nombre y contraseña. El nombre de usuario es exclusivo de cada usuario, por lo que no puede estar siendo utilizado por otro usuario. Cuando se abre la página, hay un nombre de usuario y una contraseña por defecto, que pueden ser sobrescritos con los que se quieran utilizar como credenciales en el programa.



Crear usuario

Nombre de usuario:
user1

Password:

Volver a Inicio Crear usuario

Figura 5.3.: Registro de usuario (detalle). [Autora]

Al pulsar sobre "Volver a inicio" se abrirá la página de inicio. Al pulsar sobre "Crear usuario" pueden suceder dos cosas. La primera, que el nombre de usuario introducido ya esté siendo utilizado, por lo que se informará al usuario de que el usuario ya existe, mostrando un mensaje en la página de resultado de la creación de usuario, como se muestra en la figura 5.4.



Datos del usuario

El usuario ya existe

Usuario:
[Empty input field]

Pass:
[Empty input field]

Volver a registro Login

Figura 5.4.: Registro de usuario. Mensaje de usuario ya existente (detalle). [Autora]

La segunda, es que el nombre de usuario sea válido, por lo que se

informará al usuario de que el usuario ha sido creado, como se puede ver en la figura 5.5.



Figura 5.5.: Registro de usuario. Mensaje de usuario creado (detalle). [Autora]

5.3. Página de portada

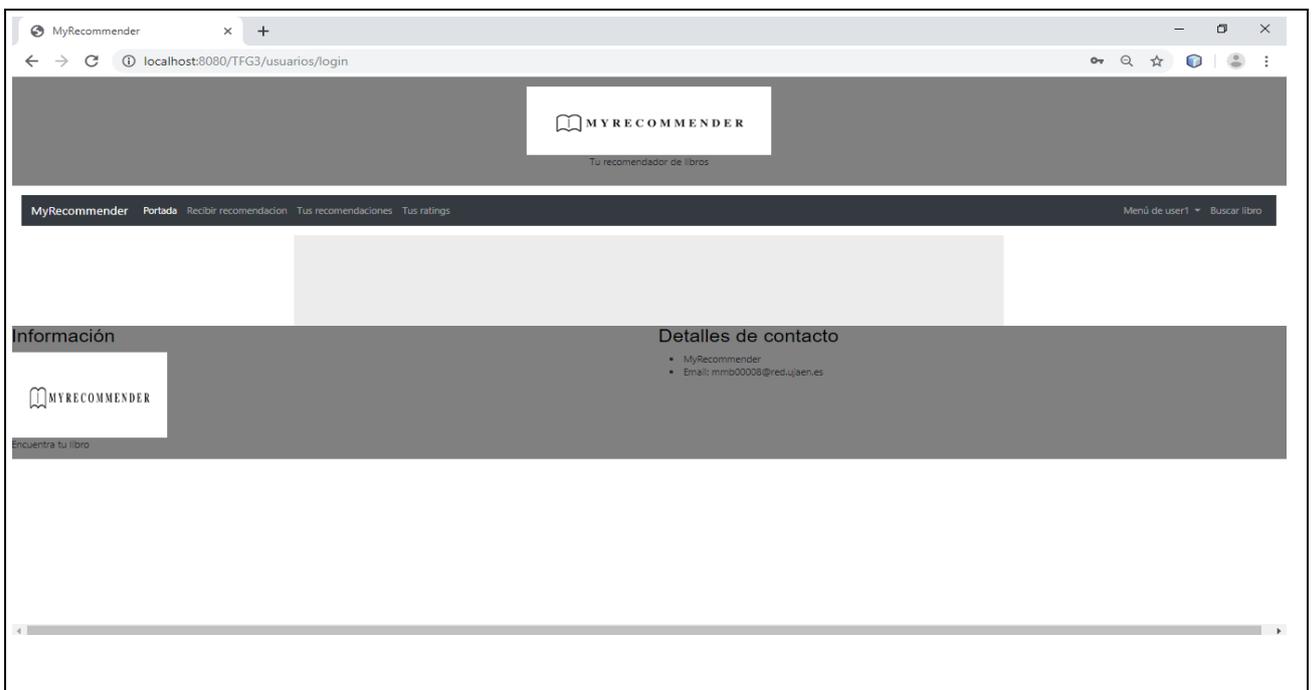


Figura 5.6.: Página de portada. [Autora]

Una vez que se ha introducido un nombre de usuario y una clave válidos en la página de login, se accede a la página de portada, que se muestra en la figura 5.6., en la que se puede ver el menú de navegación de la página. Además, en la parte central superior se sitúa el icono de la aplicación y uno de sus lemas, "Tu recomendador de libros", mostrado en la figura 5.7.



Figura 5.7.: Icono y lema 1 de la aplicación. [Autora]

En el pie de página y en la parte izquierda, podemos ver el mismo icono con el segundo lema "Encuentra tu libro", y en la otra, el nombre de la aplicación y los detalles de contacto, como se ve en la figura 5.8.



Figura 5.8.: Icono y lema 2 de la aplicación. [Autora]

5.4. Menú de navegación

En el menú de navegación, se tienen las diferentes opciones para moverse entre las páginas de la aplicación, las opciones del usuario, dentro de un menú, y la búsqueda de libros. La configuración de este menú se muestra en la figura

5.9.

Las opciones "MyRecommender" y "Portada" permiten situarse en la página de portada de la aplicación. "Recibir recomendación" lleva al usuario a una serie de páginas en las que tendrá que valorar los veinte libros más recomendados y a elegir las opciones de número y tipo de recomendaciones para su perfil, si es la primera vez que elige esta opción, o a recibir una recomendación, el resto de las veces que la seleccione.

Con las opciones "Tus recomendaciones" y "Tus ratings", el usuario puede ver, respectivamente, las recomendaciones que le ha hecho el programa en alguna de las rondas de recomendación y los libros que ha valorado.

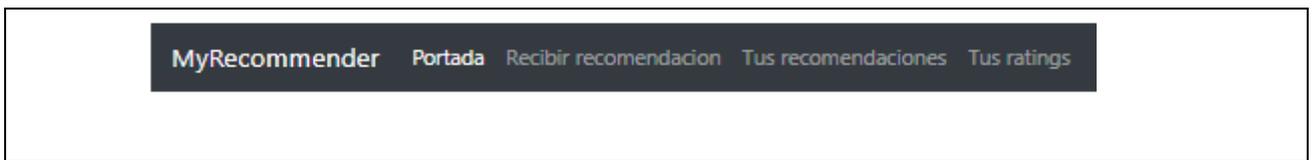


Figura 5.9.: Parte izquierda del menú (detalle). [Autora]

En la parte izquierda del menú, se sitúa el submenú con las opciones del usuario y la opción "Buscar libro". Este menú de usuario se puede ver en la figura 5.10.

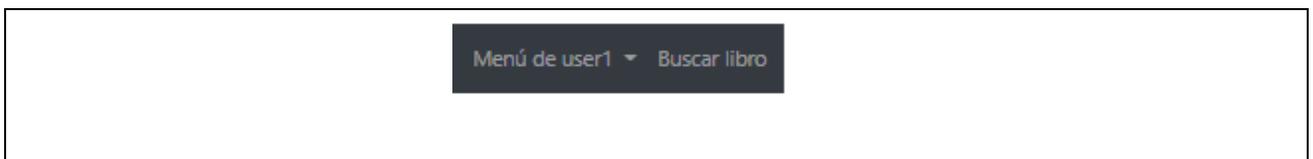


Figura 5.10.: Parte derecha del menú (detalle). [Autora]

5.5. Recibir recomendación

Si es la primera vez que se selecciona esta opción, el primer paso será valorar una serie de libros, los veinte más recomendados por otros usuarios, para que el programa pueda utilizar esa información para buscar libros para recomendar al usuario, a partir de los valorados por usuarios similares. Se presentará al usuario una lista de libros, como muestra la figura 5.11., en la que puede valorar mediante una de las opciones disponibles en una lista desplegable, que se corresponden con las etiqueta que utiliza el sistema para este fin, como se ve en la figura 5.12.

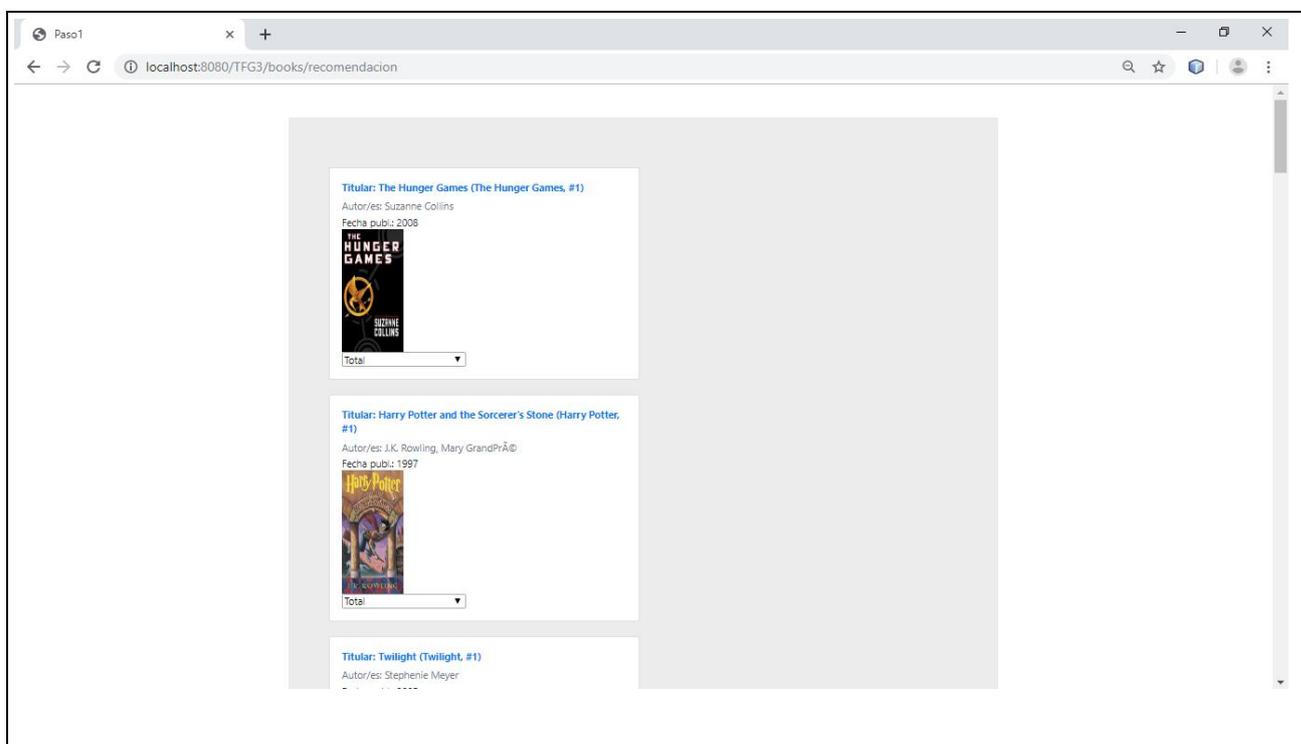


Figura 5.11.: Paso 1. Valorar los libros más valorados por el resto de usuarios. [Autora]

La opción seleccionada expresará el interés del usuario en ese libro, que puede ser "Nada", "Bajo", "Medio", "Alto" o "Total".



Figura 5.12.: Valoración del interés del usuario en un libro (detalle). [Autora]

A continuación, una vez que se seleccione "Siguiente", en el paso 2, se tendrá que seleccionar el número de recomendaciones que se quieren recibir, eligiendo una opción de la lista desplegable, del modo en el que muestran las figuras 5.13., 5.14 y 5.15.

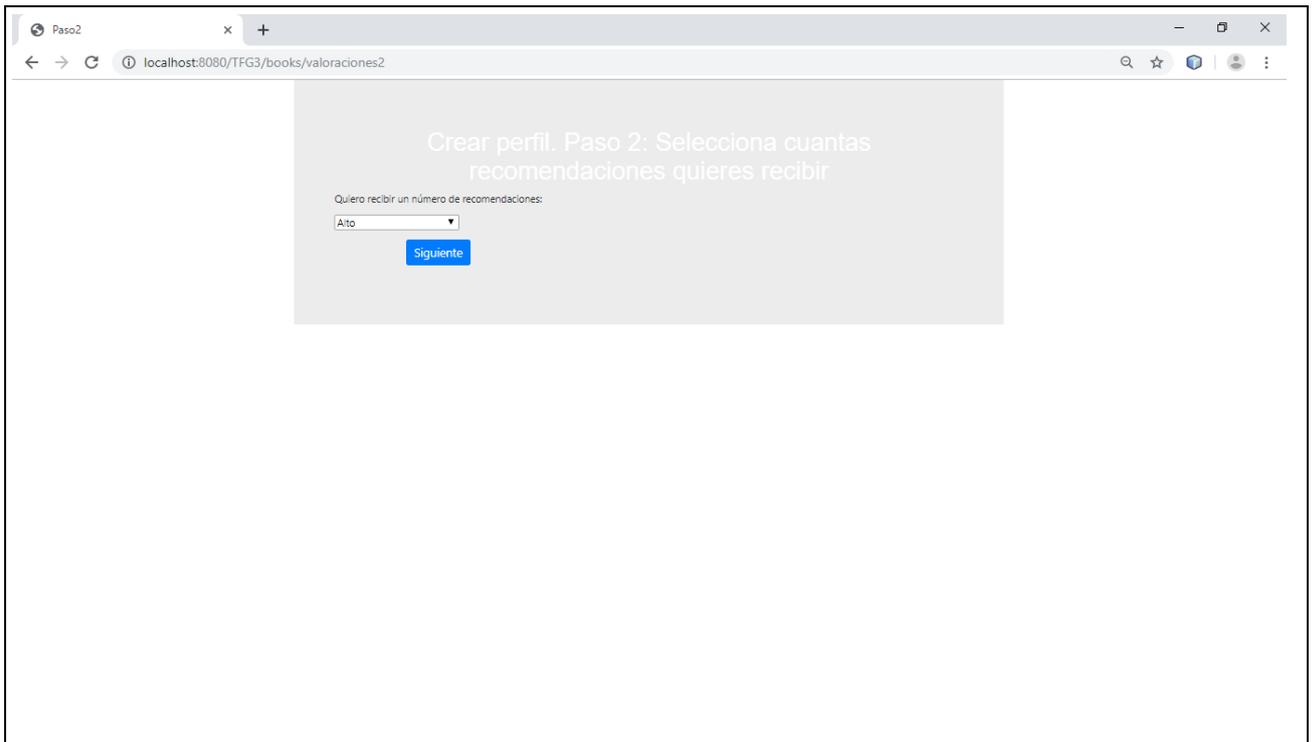


Figura 5.13.: Paso 2. Número de recomendaciones a recibir. [Autora]

Estas opciones son, "Bajo", "Medio" o "Alto". Esta información se guardará en el perfil del usuario y se puede modificar posteriormente, desde el menú de usuario, en la opción "Perfil del usuario".

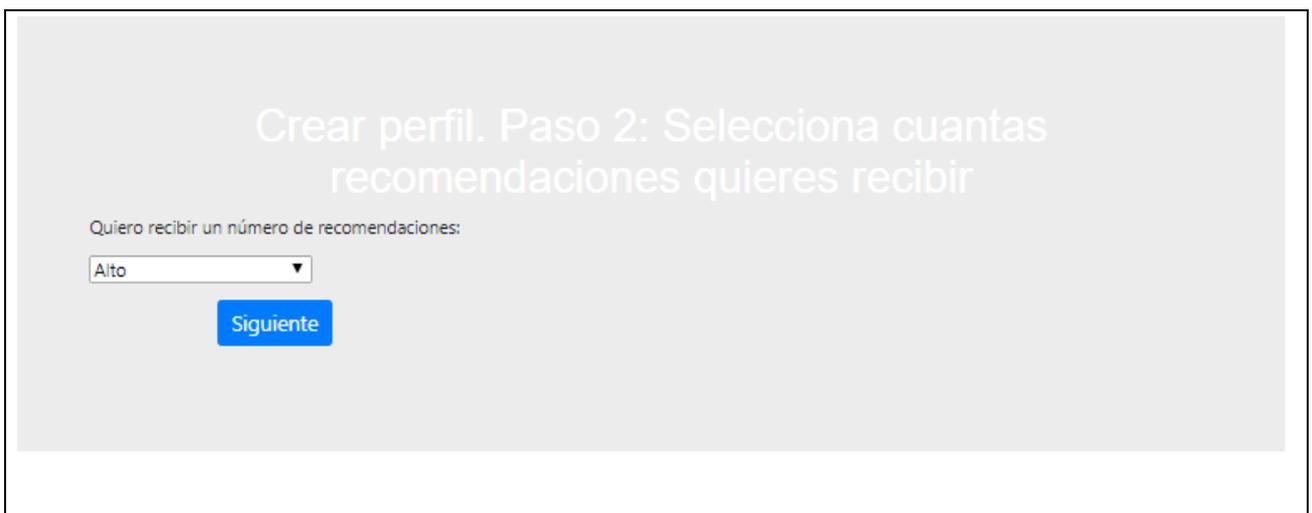


Figura 5.14.: Paso 2. Número de recomendaciones a recibir (detalle). [Autora]

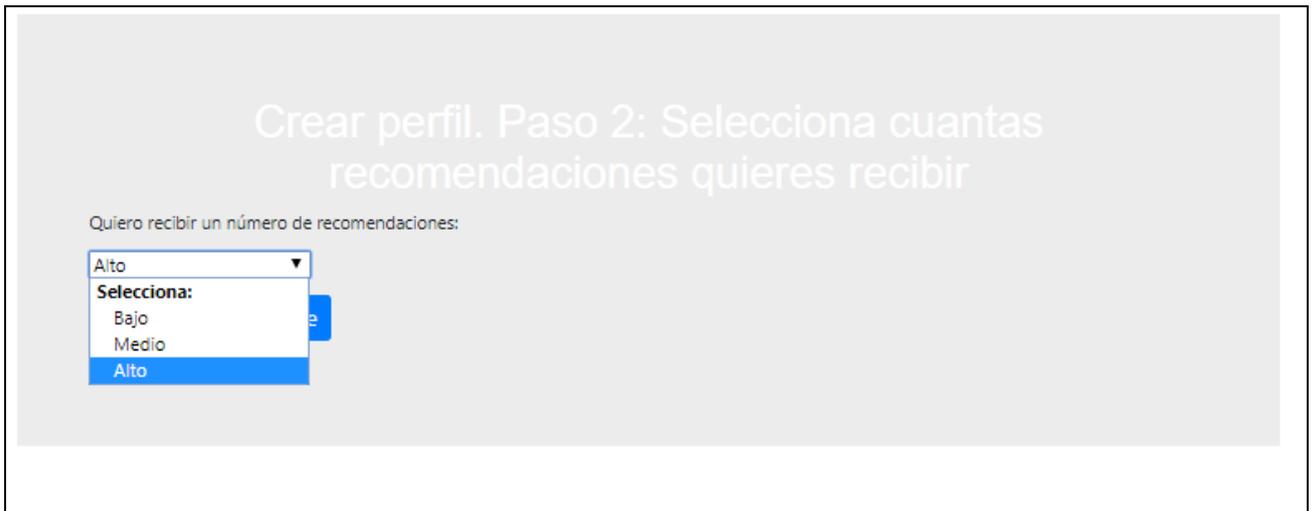


Figura 5.15.: Paso 2. Opciones sobre el número de recomendaciones a recibir (detalle).
[Autora]

Al seleccionar la opción, "Siguiente", se pasará al paso 3 de configuración del perfil, donde habrá que seleccionar el tipo de recomendaciones que se quieren recibir, como muestran las figuras 5.16. , 5.17. y 5.18.

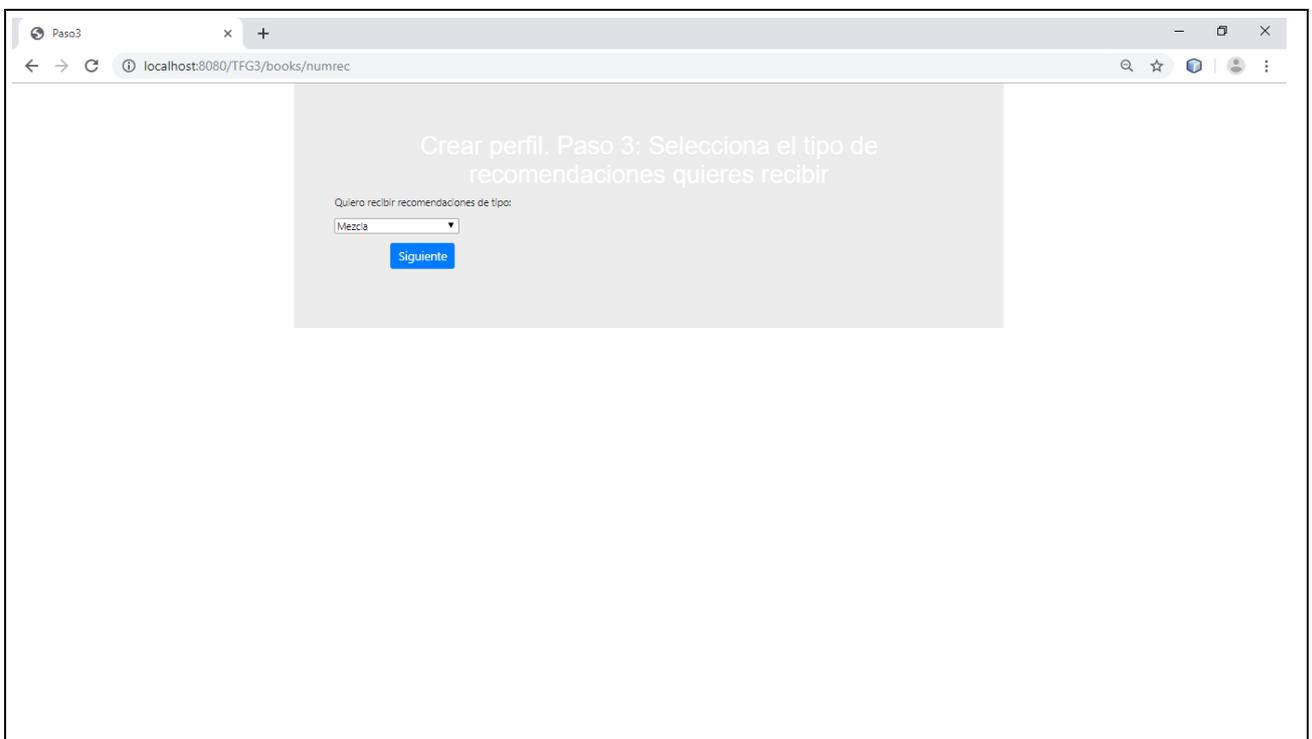


Figura 5.16.: Paso 3. Tipo de recomendaciones a recibir (detalle). [Autora]

Las opciones a seleccionar en una lista desplegable, son "Mezcla" o

"Nuevas". La opción mezcla, hará uso de la memoria del programa para utilizarlas como parte de las recomendaciones y la opción nuevas, buscará nuevas recomendaciones para el usuario.

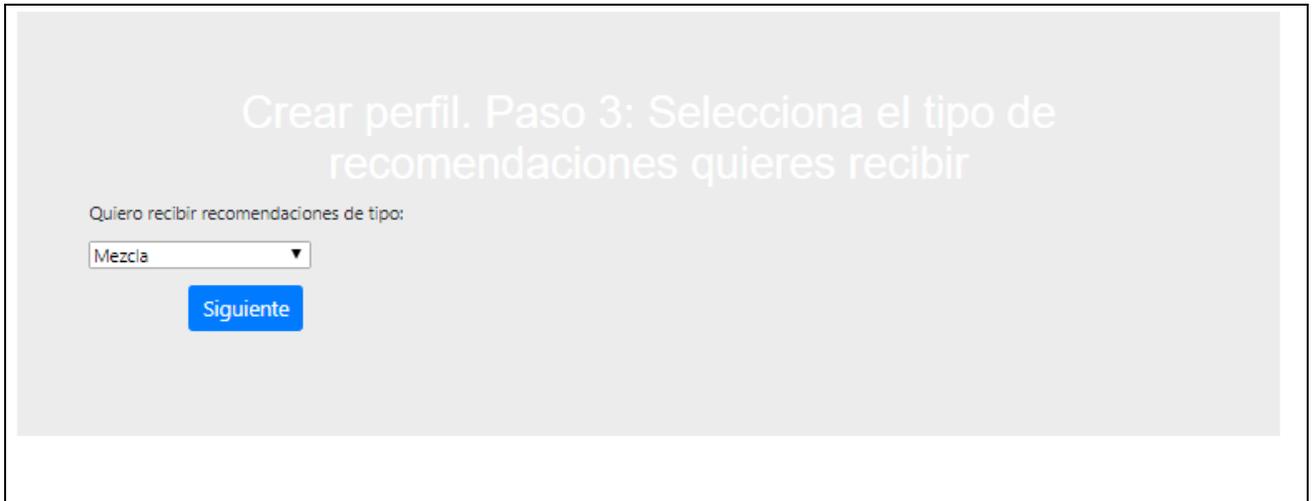


Figura 5.17.: Paso 3. Tipo de recomendaciones a recibir (detalle). [Autora]

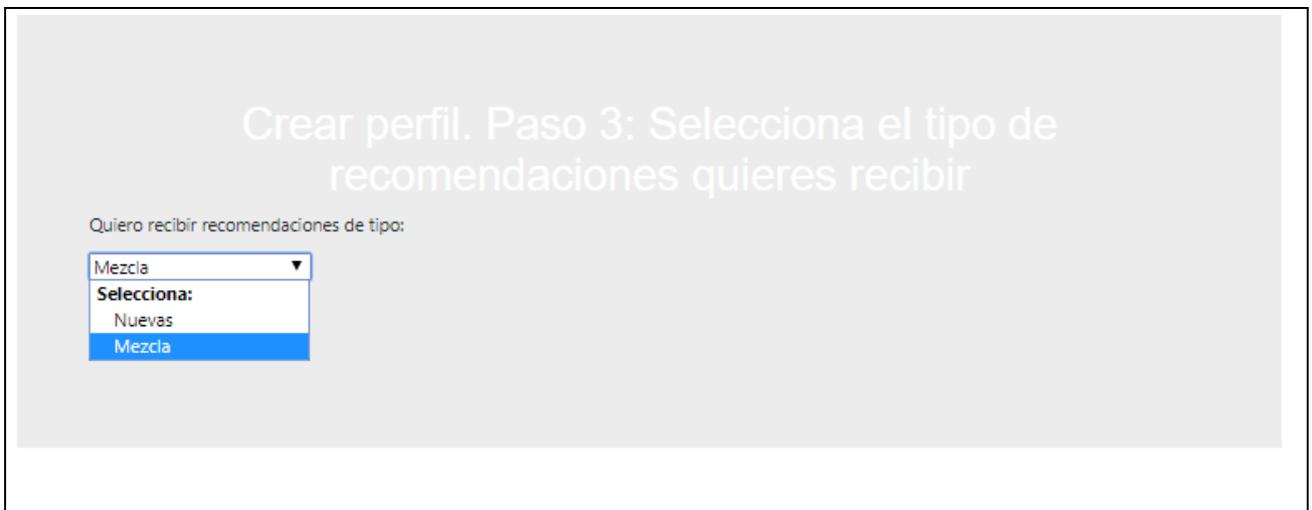


Figura 5.18.: Paso 3. Lista desplegable del tipo de recomendaciones a recibir (detalle). [Autora]

La siguiente vez que el usuario seleccione recibir recomendación, el sistema le ofrecerá una lista con el tipo y número de recomendaciones que están indicados en el perfil, como muestran las figuras 5.19. y 5.20.

Desarrollar un sistema de recomendaciones de libros basado en memoria

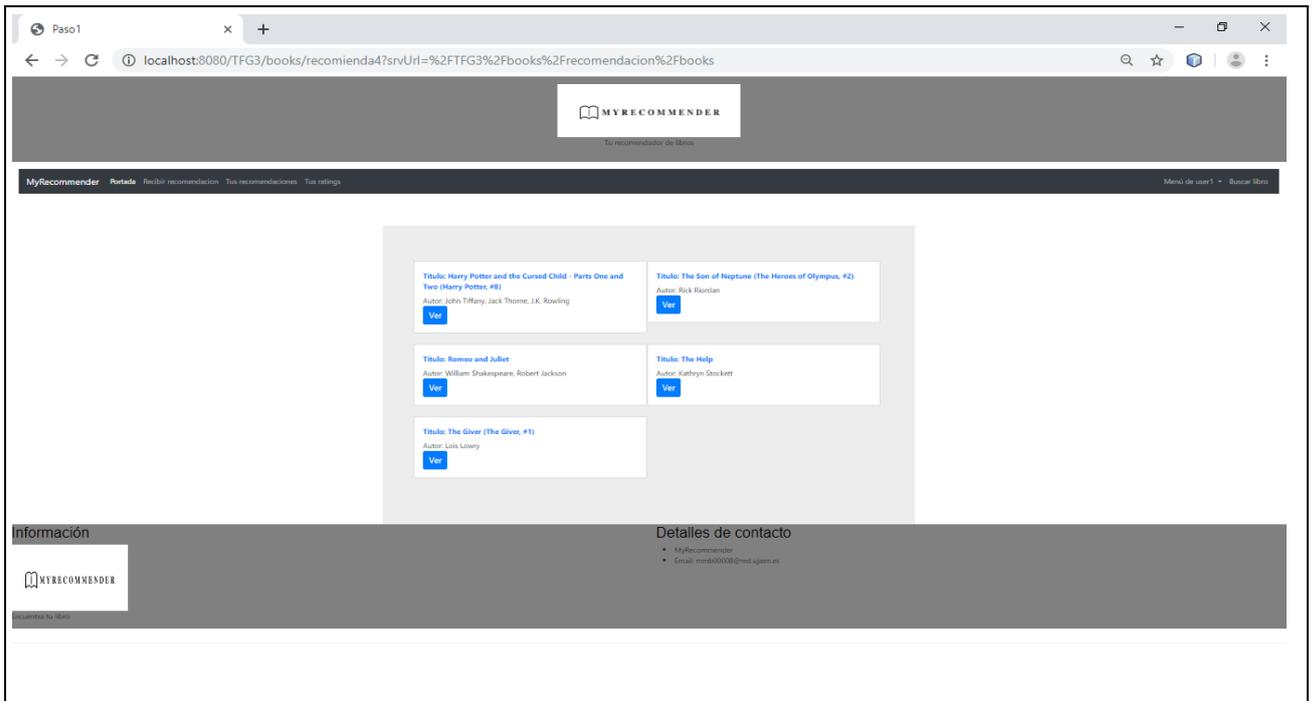


Figura 5.19.: Lista de recomendaciones para el usuario. [Autora]

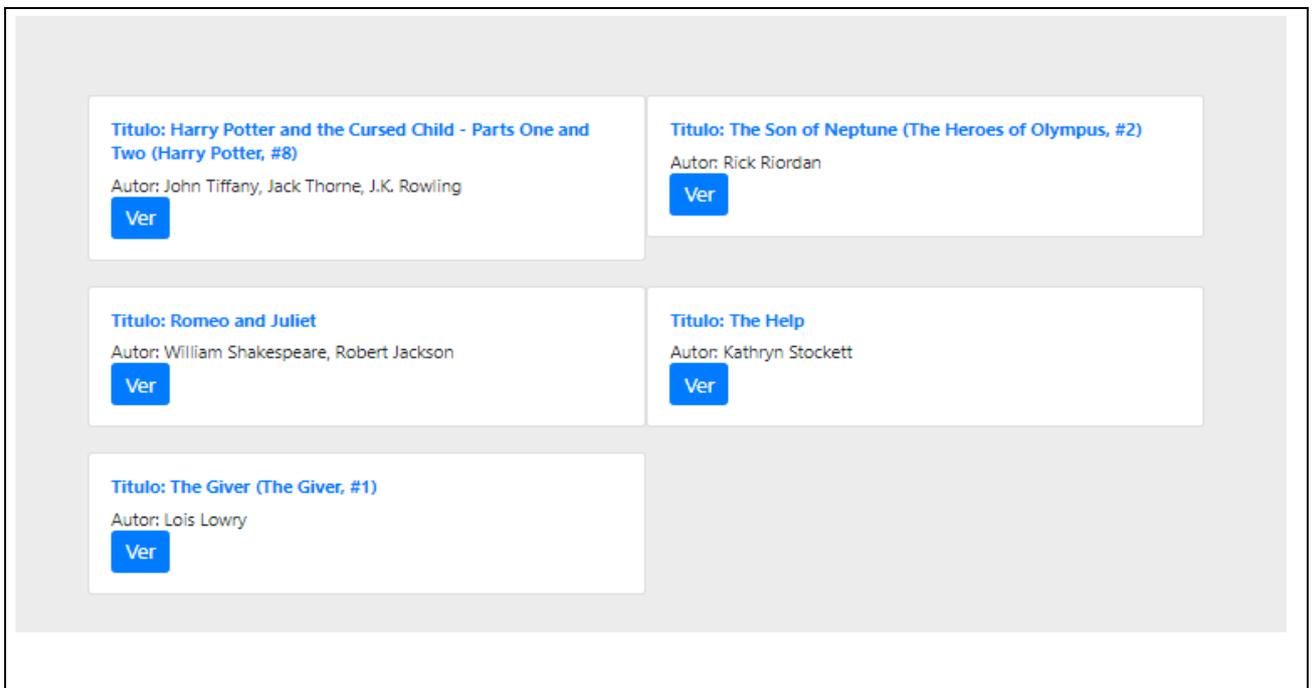


Figura 5.20.: Lista de recomendaciones para el usuario (detalle). [Autora]

5.6. Tus recomendaciones

Al pulsar sobre la opción del menú "Tus recomendaciones", se muestran

las recomendaciones que el sistema le ha hecho al usuario hasta el momento, como muestra la figura 5.21.

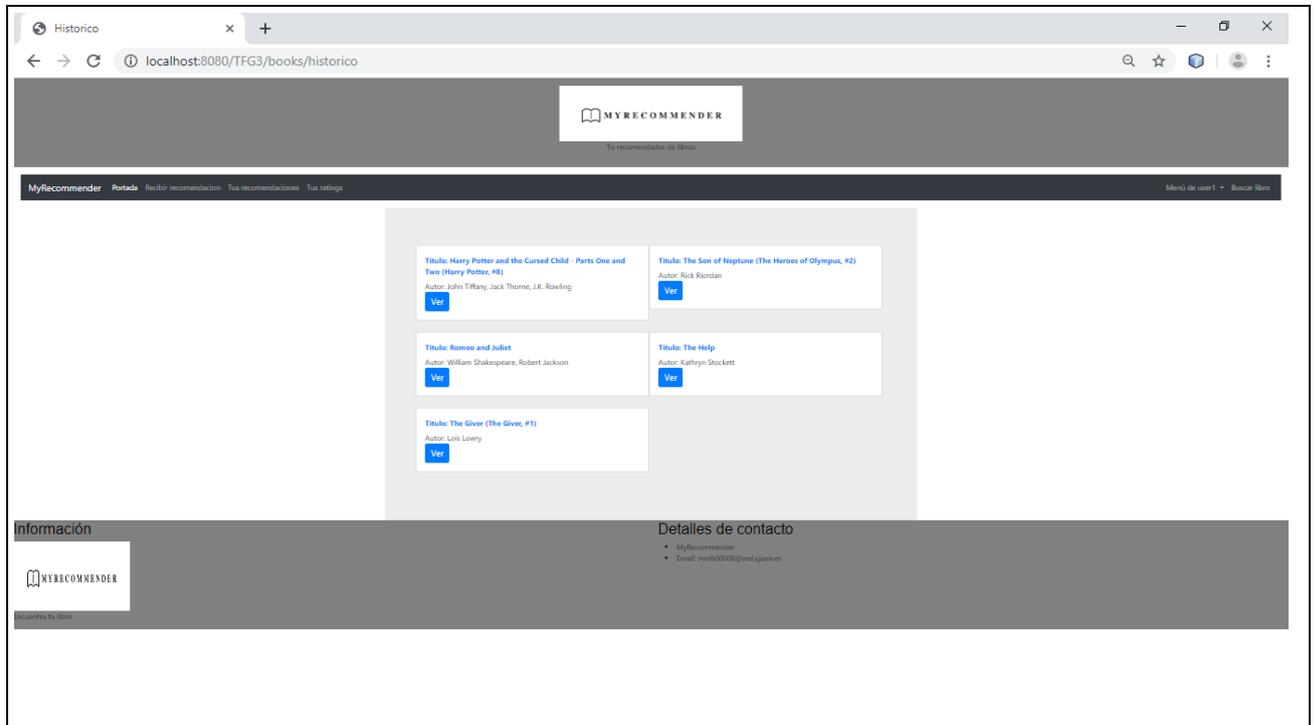


Figura 5.21.: Lista de recomendaciones del usuario. [Autora]

5.7. Tus valoraciones

Esta opción permite visualizar los libros que ha valorado un usuario. Sólo se visualizan los datos básicos del libro, título, autor y la valoración que el usuario ha emitido para el libro, como se puede ver en la figura 5.22. Mediante la opción "Ver", se pueden visualizar los detalles del libro y cambiar la valoración, en su caso.

Desarrollar un sistema de recomendaciones de libros basado en memoria

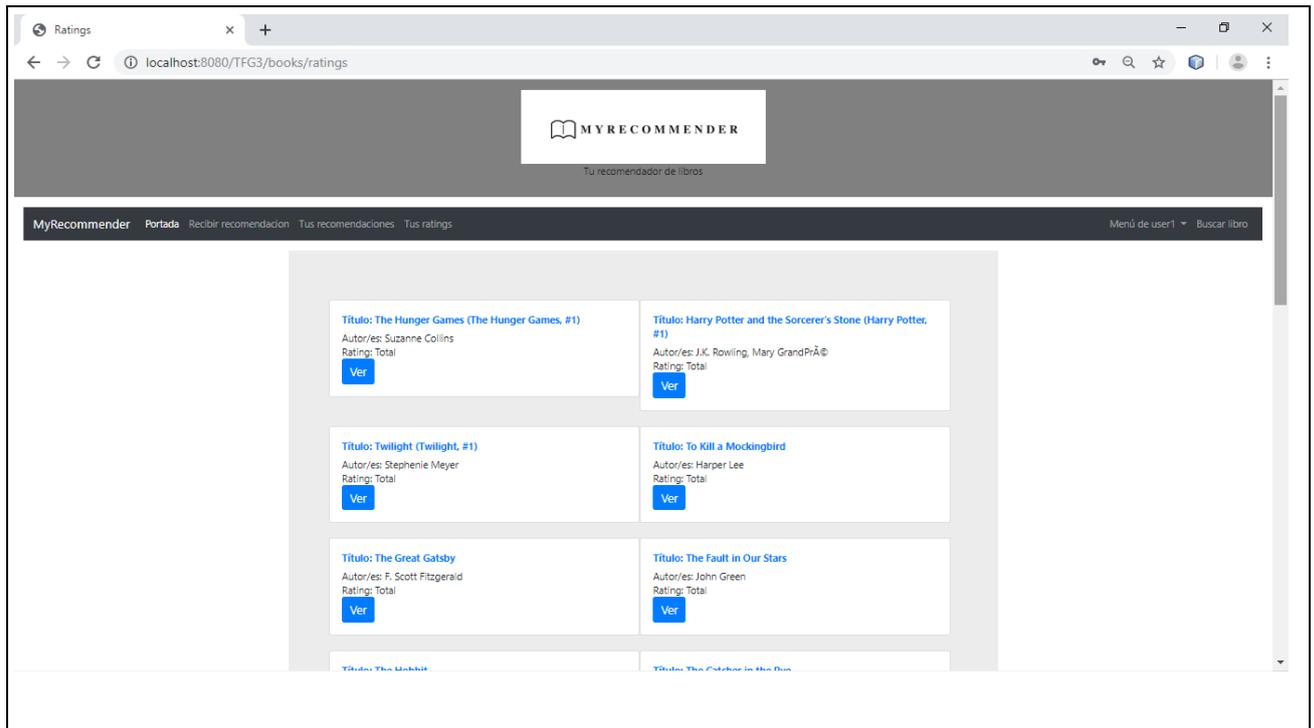


Figura 5.22.: Lista de libros valorados por el usuario. [Autora]

Al pulsar en la opción "Ver", se puede acceder a la información detallada de un libro, como muestra la figura 5.23., y modificar la valoración que se le ha dado, si se quiere cambiar.

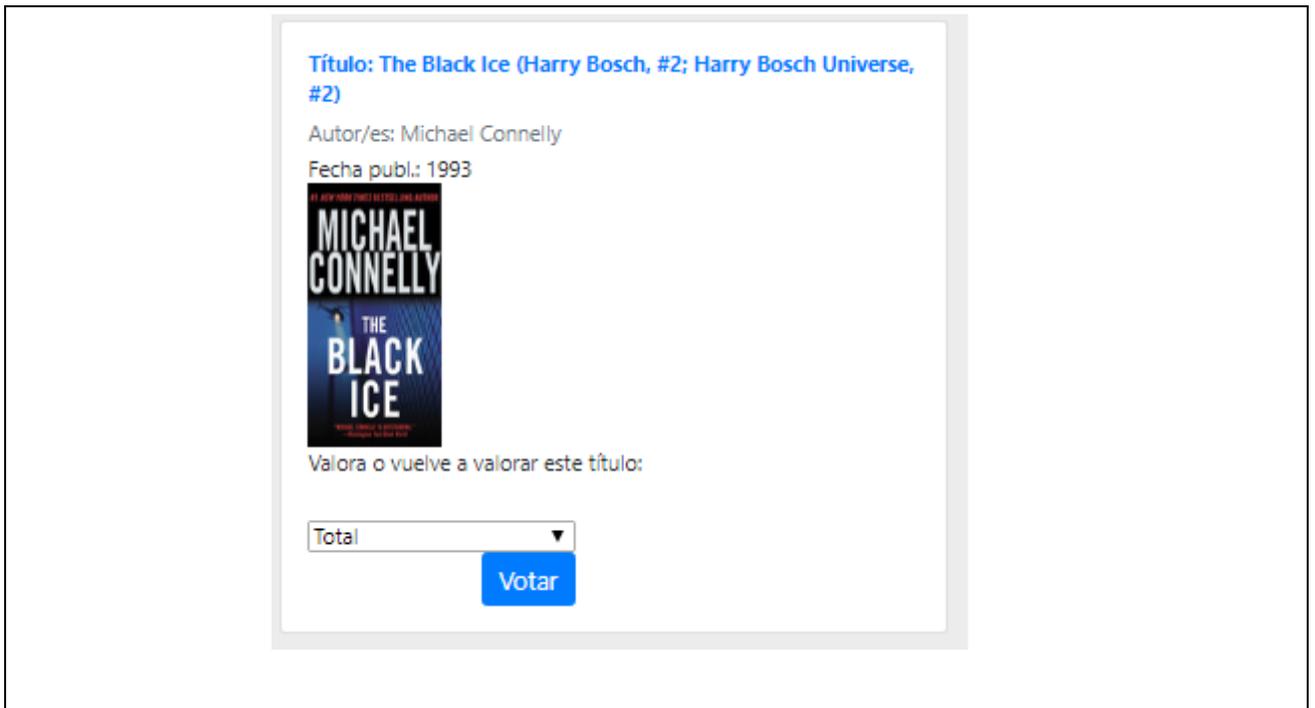


Figura 5.23.: Información detallada de uno de los libros (detalle). [Autora]

5.8. Menú del usuario

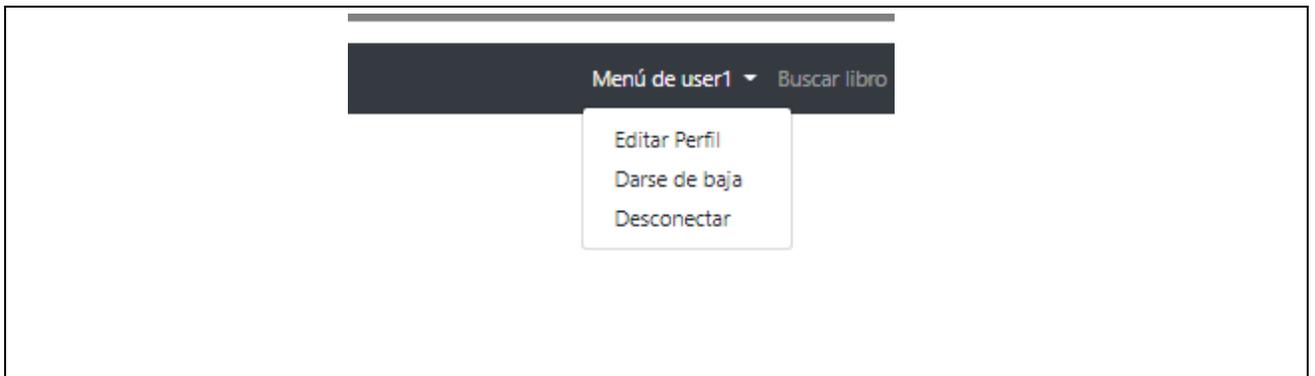
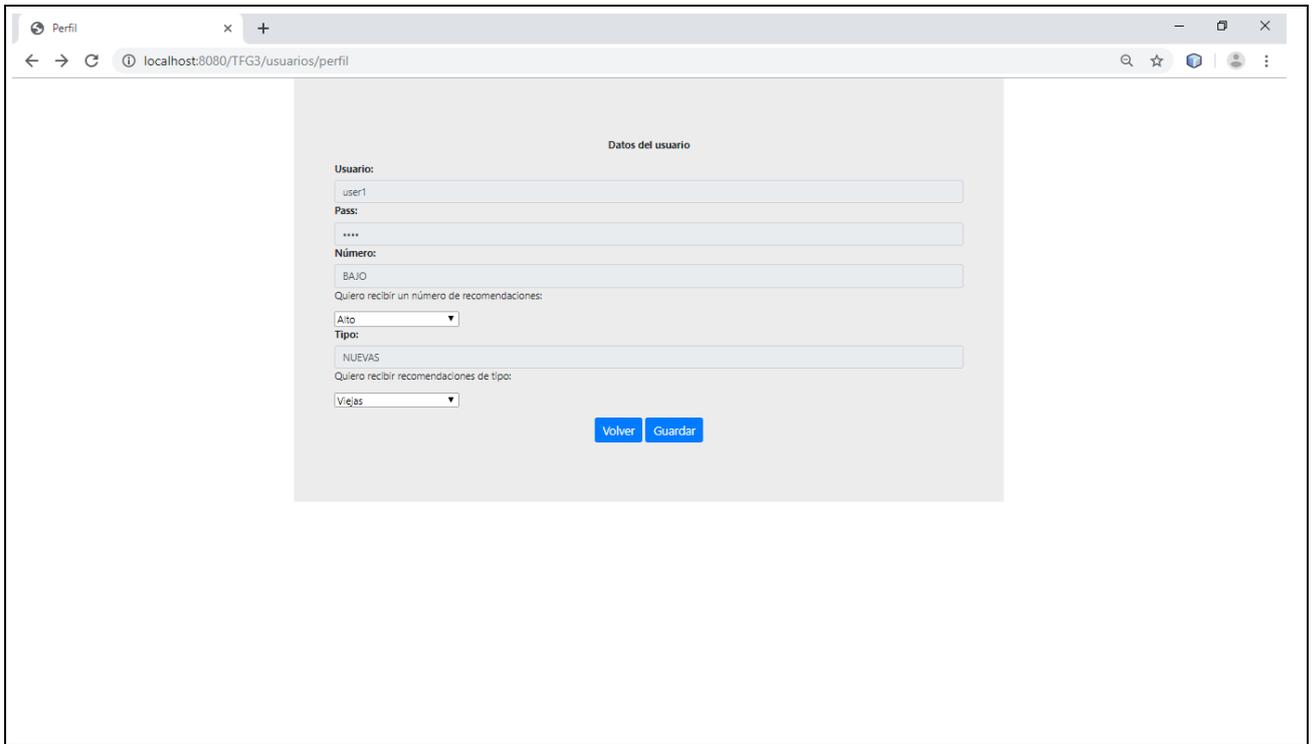


Figura 5.24.: Menú del usuario (detalle). [Autora]

Las operaciones relativas a los usuarios se han situado en un menú desplegable, bajo el título "Menú de" más el nombre del usuario activo, como muestra la figura 5.24. Estas operaciones son la edición del perfil, la baja y la desconexión del sistema.

5.8.1. Editar perfil

Al seleccionar la opción de "Editar perfil" en el menú de usuario, se abre el perfil, como muestra la figura 5.25. y con más detalle la 5.26.



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/TFG3/usuarios/perfil'. The page content is a form titled 'Datos del usuario'. The form contains the following elements:

- Usuario:** A text input field containing 'user1'.
- Pass:** A password input field with masked characters '****'.
- Número:** A text input field containing 'BAJO'.
- Quiero recibir un número de recomendaciones:** A dropdown menu with 'Alto' selected.
- Tipo:** A text input field containing 'NUEVAS'.
- Quiero recibir recomendaciones de tipo:** A dropdown menu with 'Viejas' selected.
- Buttons:** Two blue buttons labeled 'Volver' and 'Guardar'.

Figura 5.25.: Perfil del usuario. [Autora]

Datos del usuario

Usuario:
user1

Pass:

Número:
Alto

Quiero recibir un número de recomendaciones:
Alto

Tipo:
Mezcla

Quiero recibir recomendaciones de tipo:
Mezcla

Volver Guardar

Figura 5.26: Perfil del usuario (detalle). [Autora]

Se pueden modificar desde aquí el número de recomendaciones a recibir, y el tipo, como muestran las figuras 5.27. y 5.28. respectivamente.

Quiero recibir un número de recomendaciones:

Alto

Selección:

- Bajo
- Medio
- Alto

nes de tipo:

Mezcla

Figura 5.27.: Cambio del número de recomendaciones en el perfil (detalle). [Autora]

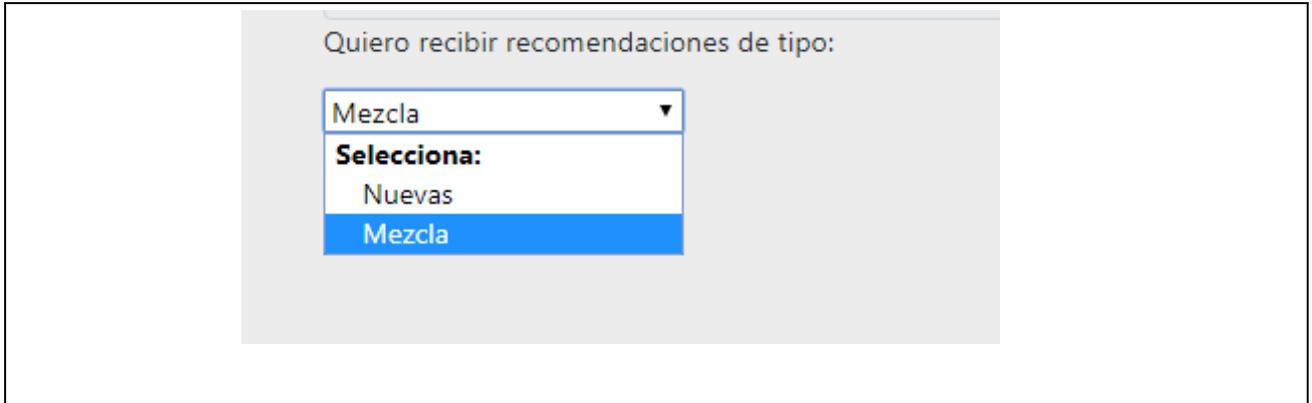


Figura 5.28.: Cambio del tipo de recomendaciones en el perfil (detalle). [Autora]

5.8.2. Darse de baja

Desde el menú del usuario, se puede seleccionar la opción de darse de baja de la aplicación. Tras realizar esta opción, el programa solicitará la confirmación de la baja, como muestran las figuras 5.29. y 5.30.

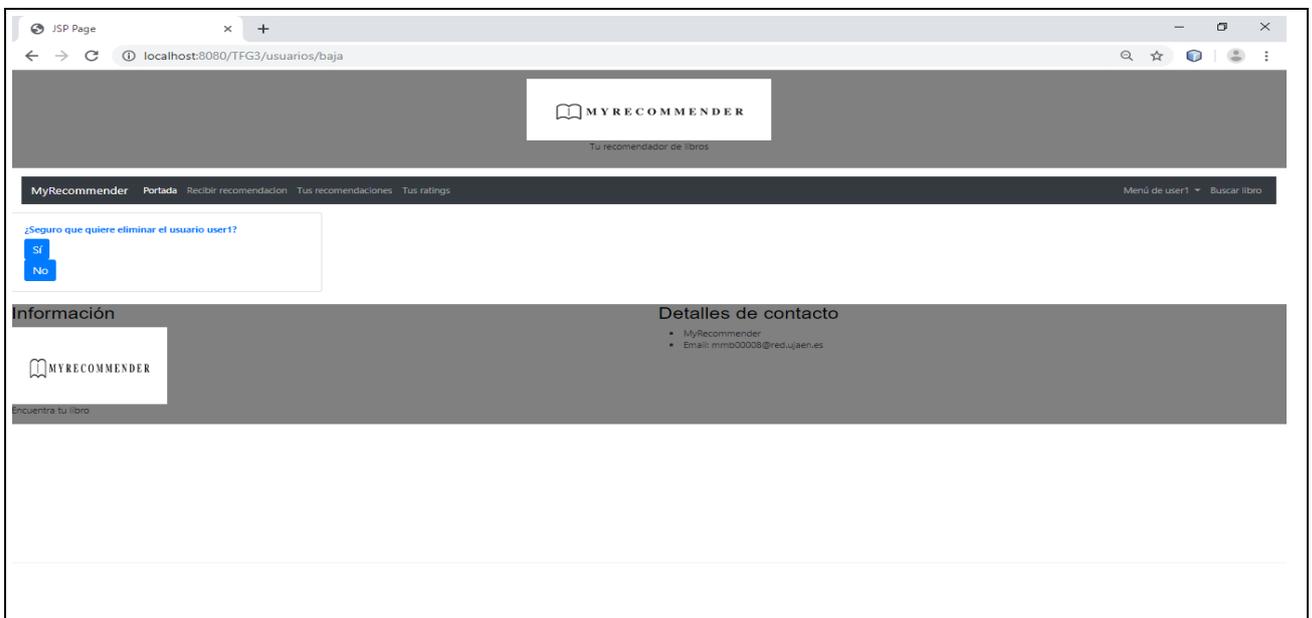


Figura 5.29.: Baja del sistema. [Autora]

Si se selecciona la opción de continuar con la operación, el usuario será dado de baja del sistema, además de eliminar sus valoraciones y sus

recomendaciones.



Figura 5.30.: Baja del sistema (detalle). [Autora]

Si se selecciona la opción de no continuar, la baja no tendrá efecto y se volverá a la página de portada.

5.8.3. Desconectar

Cuando se selecciona la opción de desconectarse del sistema desde el menú de usuario, el usuario deja de ser el usuario activo y se abrirá la página de inicio.

5.9. Buscar un libro

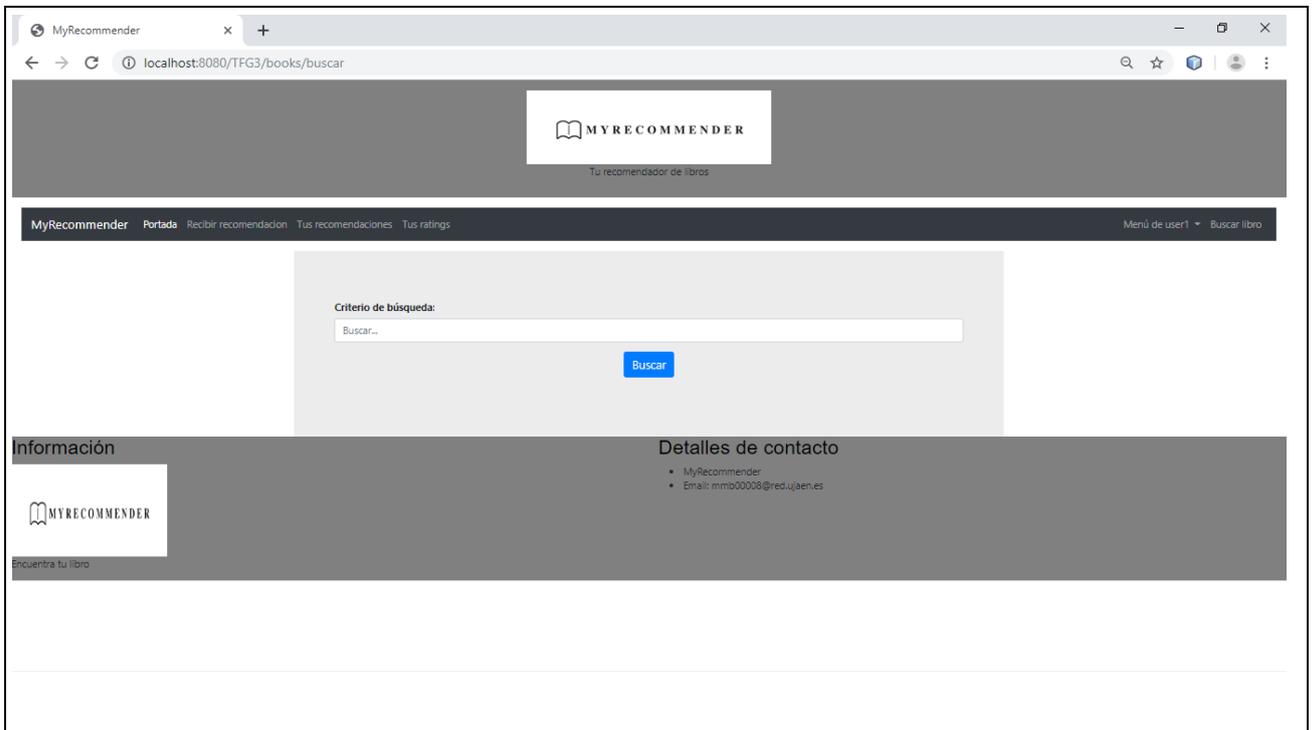


Figura 5.31.: Búsqueda de un libro. [Autora]

La página cuenta con un buscador que permite buscar libros, utilizando un sólo formulario, por múltiples criterios, como muestra la figura 5.31.y más detalladamente la 5.32.

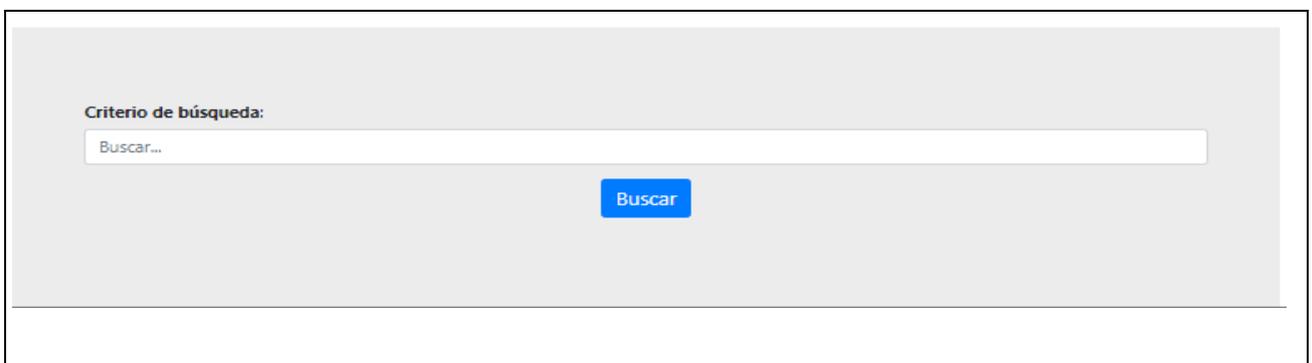
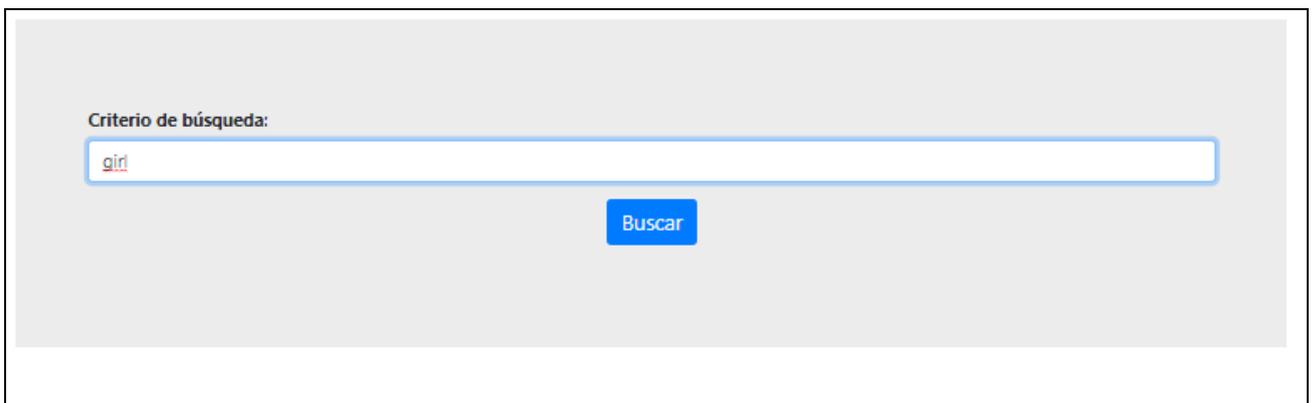


Figura 5.32.: Búsqueda de un libro (detalle). [Autora]

Una vez introducido un criterio, como muestra la figura 5.33., y se selecciona "Buscar", el programa seleccionará aquellos libros que coincidan con ese criterio en alguno de sus campos: título, autor, etc., y los devolverá en forma de lista que puede ser visualizada por el usuario, como muestra la figura 5.34.

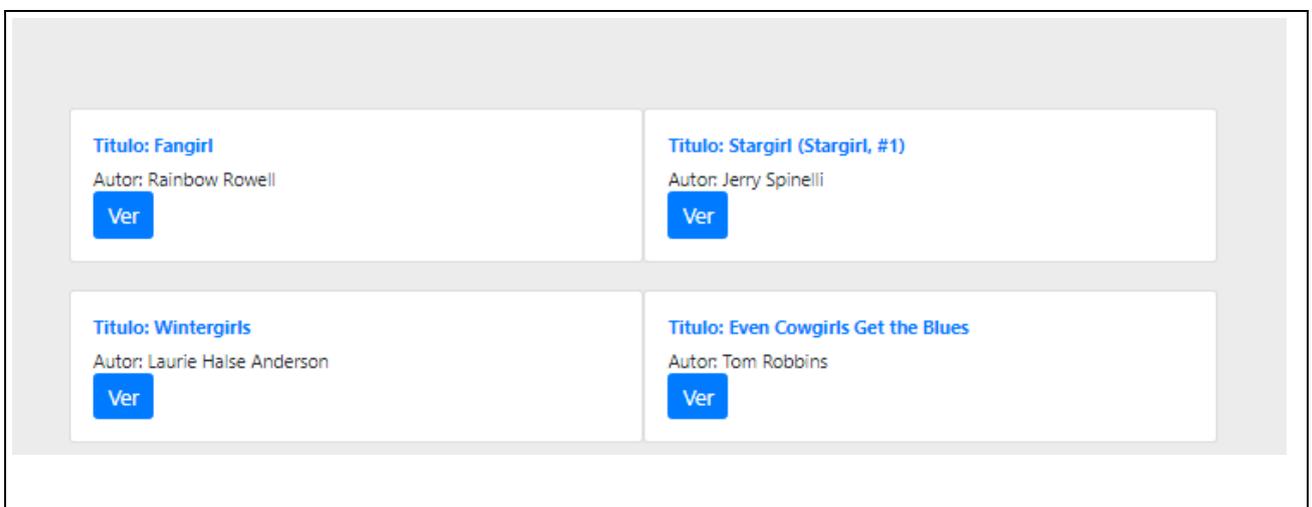


Criterio de búsqueda:

Buscar

Figura 5.33.: Búsqueda con el criterio "girl" (detalle). [Autora]

Desde la página que muestra el resultado de la búsqueda, se pueden ver los detalles de un libro y valorarlo o volver a valorarlo, para que esa información pase a formar parte del perfil del usuario, y se tenga en cuenta para hacerle futuras recomendaciones.



<p>Título: Fangirl Autor: Rainbow Rowell Ver</p>	<p>Título: Stargirl (Stargirl, #1) Autor: Jerry Spinelli Ver</p>
<p>Título: Wintergirls Autor: Laurie Halse Anderson Ver</p>	<p>Título: Even Cowgirls Get the Blues Autor: Tom Robbins Ver</p>

Desarrollar un sistema de recomendaciones de libros basado en memoria

Figura 5.34.: Resultado de la búsqueda con el criterio "girl" (detalle). [Autora]

6. PRUEBAS

Para probar el correcto funcionamiento de la aplicación, se han utilizado datos que intentan hacerla fallar. La idea de la realización de las pruebas es descubrir errores que no se habían detectado durante su programación. Por ejemplo, si se lograra crear un nuevo usuario utilizando un nombre de usuario ya asignado, se estaría descubriendo un error en la aplicación y se habría realizado una prueba con éxito.

Para hacer pruebas sobre las aplicaciones se suelen utilizar dos estrategias diferentes: las pruebas de caja negra y las pruebas de caja blanca.

Las pruebas de caja blanca se ocupan de la codificación de las funciones, son pruebas estructurales. Mientras que las de caja negra se interesan por la salida de las funciones en base a la entrada que se les proporciona.

Para este proyecto se ha utilizado la segunda aproximación, las pruebas de caja negra, como se explica en el siguiente capítulo.

6.1. Pruebas de caja negra

El objetivo de las pruebas de caja negra es probar la entrada y salida de una función, son pruebas funcionales. Buscan probar si se han tenido en cuenta todas las posibles entradas y si cada entrada produce una salida esperada.

6.1.1. Pruebas de caja negra realizadas

Las pruebas de caja negra que se han realizado tenían por objeto descubrir errores de salida de las funciones o errores de acceso a la base de datos. Los problemas presentados se resolvieron inmediatamente.

Para la selección de las pruebas a realizar, se ha partido desde los casos de uso identificados durante la fase de análisis y se han dividido por iteraciones. Ya que el resultado de cada iteración es un producto que funciona, se han hecho pruebas para cada sprint (iteraciones 1, 2 y 3), para comprobar que las funcionalidades que forman parte de cada uno de ellos producen la respuesta esperada, incluyendo la navegación por la página web.

6.1.1.1. Pruebas de caja negra de la iteración 1

En primer lugar, para la página principal se decidió hacer las pruebas que se indican en la tabla 6.1., probando el correcto funcionamiento de los dos botones, el de entrar y el de registro. El resultado esperado es que, al pulsar el primero, se ha de abrir la página de login y el de registro la página de creación de un nuevo usuario o registro.

Página principal	Resultado
El usuario selecciona "Entrar"	Se abre la página de login
El usuario selecciona "Registro"	Se abre la página de registro

Tabla 6.1.: Pruebas de caja negra de la página principal [Elaboración propia]

En la página de registro se hicieron las pruebas de los botones para volver a la página principal, pulsando el botón con el texto volver a inicio, y el que abre la página de creación de usuario, como se muestra en la tabla 6.2.

Página de registro	Resultado
El usuario selecciona "Volver a inicio"	Se abre la página principal
El usuario selecciona "Crear usuario"	Se muestra la página de información sobre la creación del usuario

Tabla 6.2.: Pruebas de caja negra de la página de registro [Elaboración propia]

Respecto al menú de navegación de la página, se realizaron las pruebas que muestra la tabla 6.3., comprobando que cada opción funcionara como se esperaba.

Menú de navegación	Resultado
Un usuario pulsa sobre "MyRecommender"	Se abre la página de portada

Un usuario pulsa sobre "Portada"	Se abre la página de portada
Un usuario pulsa sobre "Recibir recomendación"	Se abre la primera página de introducción de datos sobre el perfil de recomendaciones si es la primera vez que se selecciona o se abre la página de recomendaciones
Un usuario pulsa sobre "Tus recomendaciones"	Se abre la página del histórico de recomendaciones
Un usuario pulsa sobre "Tus ratings"	Se abre la página de valoraciones
Un usuario pulsa sobre "Buscar un libro"	Se abre la página de búsqueda
Un usuario pulsa sobre "Menú de (nombre de usuario)"	Se despliega el menú del usuario
Un usuario pulsa sobre "Editar perfil"	Se muestra el perfil del usuario
Un usuario pulsa sobre "Darse de baja"	Se abre la página de confirmación
Un usuario pulsa sobre "Desconectar"	Se abre la página principal

Tabla 6.3.: Pruebas de caja negra del menú de navegación [Elaboración propia]

6.1.1.2. Pruebas de caja negra de la iteración 2

A la hora de probar la página de registro de usuario, éste sólo se crea si el nombre introducido no está ocupado por otro usuario del sistema, por lo que se avisará al usuario, mediante un mensaje, sobre si el usuario se ha creado o sobre la preexistencia de un usuario con ese mismo nombre. Por tanto, se hicieron las pruebas que se muestran en la tabla 6.4., intentando crear un usuario con el nombre de uno que ya existía y otro con un nombre que no estaba incluido en el sistema.

Registro de usuario (página de registro)	Resultado
Un usuario quiere crear un usuario	Se crea el usuario

Un usuario quiere crear un usuario con un nombre de usuario ya utilizado por otro usuario	No se crea el usuario
---	-----------------------

Tabla 6.4.: Pruebas de registro de usuario. [Elaboración propia]

En la página de acceso, como se muestra en la tabla 6.5., se hicieron pruebas intentando acceder con credenciales válidas e inválidas y el sistema respondió correctamente en cada uno de los casos.

Acceso a la aplicación (página de login)	Resultado
Un usuario introduce un usuario y contraseña válidos	Se accede a la página principal
Un usuario introduce un usuario válido y una contraseña no válida	Se informa al usuario de que la contraseña es inválida
Un usuario introduce un usuario que no existe	Se informa al usuario de que el usuario no existe

Tabla 6.5.: Pruebas de acceso a la aplicación. [Elaboración propia]

Respecto a la baja de usuario, como muestra la tabla 6.6., al pulsar sobre esta opción se tenía que abrir la página de confirmación de baja.

Darse de baja (menú de usuario)	Resultado
El usuario selecciona darse de baja en el menú	Se abre la página de confirmación de baja

Tabla 6.6.: Pruebas de baja de usuario. [Elaboración propia]

En la página de confirmación de baja, un mensaje le indica al usuario si está seguro de que quiere darse de baja del sistema. Se probaron los dos

botones que corresponden a las respuestas afirmativa y negativa. Si el usuario pulsa sí, el usuario es dado de baja del sistema y se abre la página principal. Si el usuario responde que no, se abriría la portada de la aplicación y el usuario la puede seguir utilizando con normalidad. Las pruebas tuvieron el resultado esperado, como se puede ver en la tabla 6.7.

Página de confirmación de baja	Resultado
El usuario pulsa "Sí"	Se elimina el usuario, sus recomendaciones y sus valoraciones de la base de datos
El usuario pulsa "No"	Se abre la página de la portada

Tabla 6.7.: Pruebas de caja negra de confirmación de baja. [Elaboración propia]

Para el perfil del usuario, se hicieron las pruebas que muestra la tabla 6.8., para que comprobar que se actualiza correctamente el perfil del usuario cuando desea cambiar el tipo o el número de recomendaciones, o ambas, que desea recibir.

Editar perfil (página de perfil)	Resultado
Un usuario selecciona cambiar el número de recomendaciones que quiere recibir desde su perfil	Al seleccionar la opción de "guardar" se actualiza el número de recomendaciones a recibir en el perfil del usuario
Un usuario selecciona cambiar el tipo de recomendaciones que quiere recibir desde su perfil	Al seleccionar la opción de "guardar" se actualiza el tipo de recomendaciones a recibir en el perfil del usuario

Tabla 6.8.: Pruebas de edición de perfil. [Elaboración propia]

6.2.1.3. Pruebas de caja negra de la iteración 3

Para esta iteración, se probó en primer lugar, la visualización de los libros valorados por el usuario en el sistema, como se puede ver en la tabla 6.9.

El sistema tiene que localizarlos en la base de datos y mostrarlos en pantalla.

Ver lista de libros valorados (menú de navegación)	Resultado
Un usuario selecciona ver la lista de los libros que ha valorado	Puede visualizar la lista de los libros que ha valorado

Tabla 6.9.: Pruebas de visualización de libros valorados. [Elaboración propia]

En segundo lugar, como se ve en la tabla 6.10., se probó la visualización de las recomendaciones que le ha ofrecido el sistema hasta el momento al usuario.

Ver lista de recomendaciones (menú de navegación)	Resultado
Un usuario selecciona ver la lista de recomendaciones que ha recibido	Puede visualizar la lista de los libros que se le han recomendado

Tabla 6.10.: Pruebas de visualización de recomendaciones. [Elaboración propia]

En tercer lugar, se probó la opción de recibir una recomendación según los criterios de número y tipo incluidos en el perfil de usuario, como muestra la tabla 6.11.

Recibir recomendación (menú de navegación)	Resultado
Un usuario selecciona recibir recomendación	Se visualiza el número de libros recomendados que el usuario ha de terminado en su perfil que quiere recibir

Tabla 6.11.: Pruebas de recepción de recomendaciones. [Elaboración propia]

En cuarto lugar, como refleja la tabla 6.12., se probaron las opciones que se pueden realizar sobre los libros, ya sea los valorados o los recomendados, ya que desde una de estas listas el usuario puede darle a la opción ver de uno de los libros, y cambiar una valoración dada anteriormente, en el caso de libros valorados, o dar una valoración a un libro no votado antes, en el caso del listado de libros recomendados.

Cambiar valoración de un libro (desde opción "ver" un libro)	Resultado
Un usuario selecciona la opción "ver" un libro.	El usuario puede visualizar los datos de ese libro
Un usuario valora un libro con una etiqueta	Se actualiza la valoración para ese libro en la base de datos
Un usuario selecciona buscar un libro por un criterio	Se recuperan los libros que cumplan con el criterio de búsqueda

Tabla 6.12.: Pruebas de cambio de valoración de un libro. [Elaboración propia]

Por último, como se ve en la tabla 6.13., se probó la desconexión de la aplicación, al seleccionar esta opción en el menú del usuario, tras lo que el usuario deja de ser el usuario activo y se abriría la página de inicio de la aplicación.

Desconectarse de la aplicación (menú de usuario)	Resultado
Un usuario selecciona desconectarse de la aplicación desde el menú de usuario	El usuario deja de ser el usuario activo y se abre la página de acceso a la aplicación

Tabla 6.13.: Pruebas de desconexión de la aplicación. [Elaboración propia]

7. CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS

Este capítulo, dividido en dos secciones, está dedicado a exponer las conclusiones extraídas de la realización de este proyecto y a proponer líneas de trabajo para el futuro.

7.1. Conclusiones

Los sistemas de recomendación están cada vez más presentes en todas las áreas en las que se necesita filtrar una gran cantidad de productos disponibles, ya que resultaría inabarcable para el usuario revisar cada uno de ellos de forma individual, para ver si encajan o no con lo que buscan. Por consiguiente, son imprescindibles para que los resultados de su búsqueda contengan una cantidad menor de ruido (resultados que se ofrecen al usuario pero que no le interesan).

Dentro de los enfoques en los que se ha trabajado hasta ahora en la elaboración de sistemas de recomendación se ha optado por utilizar un enfoque híbrido, que combina técnicas usadas por los sistemas colaborativos y los basados en contenidos, y que se apoya en dos herramientas adicionales:

1. El uso de una memoria, que permite recordar libros que se podrían haber recomendado al usuario, si no se hubiesen quedado fuera por las limitaciones acerca del número de recomendaciones que se querían recibir.
2. La utilización de un sistema de valoraciones que tiene como base etiquetas, lo que facilita al usuario expresar sus preferencias de una manera más cercana al lenguaje natural que suele utilizar.

El uso de la memoria es especialmente interesante para el mundo editorial, como se indicó en un capítulo anterior, ya que cada año se publican millones de libros nuevos que se van incorporando al mercado. Pero eso no hace menos válidos los publicados anteriormente y que ya formaban parte del sistema.

Para el caso de los libros, parece que resulta lógico que la primera técnica para recomendar libros sea la colaborativa. Desde que se lee masivamente, las

personas han ido recomendándose libros entre sí. Pero, a la vez, esto sólo permitía tener acceso a un reducido número de títulos, los que conociera tu círculo cercano o de los que el lector hubiese oído hablar por algún medio. Con la incorporación de los sistemas de recomendación a editoriales, comercios o a páginas especializadas en lectura, se amplía el número de títulos a los que se tienen acceso y también el de los nuevos títulos que se pueden descubrir.

Pero, como se ha indicado anteriormente, esta clase de sistemas de recomendación tienen un problema: el problema de arranque en frío (cold-start problem). Esto es, un usuario no puede recibir recomendaciones desde el inicio, ya que el sistema no tiene información para compararla con otros usuarios y elaborar una recomendación para él.

Este problema se ha intentado minimizar obligando al usuario a votar una serie de libros antes de recibir su primera recomendación, concretamente se ha optado por seleccionar los veinte libros más votados por el total de usuarios en el sistema. Con esta información y con los filtros acerca del número y tipo de recomendaciones que se quieren recibir, el sistema puede empezar a ofrecerlas.

El sistema localiza los vecinos más cercanos disponibles en el sistema y las votaciones para ver si un libro es recomendado o no al usuario. Este grupo de usuarios irá cambiando con el tiempo, ya que a medida que un usuario realice nuevas votaciones el cálculo de la similitud entre usuarios irá aumentando o disminuyendo.

Otra de las cuestiones que se ha tenido en cuenta es que el gusto del usuario puede cambiar con el tiempo. De modo que si decide cambiar su valoración acerca de un libro votado anteriormente, esta información se actualizará en la base de datos.

Por lo tanto, este trabajo propone una solución al problema de crear un sistema de recomendación de libros que pueda ofrecer recomendaciones útiles al usuario y que se adapte a sus necesidades presentes y futuras.

7.2. Líneas de trabajo futuras

En esta última sección, se han recogido algunas posibles mejoras para el

sistema.

En primer lugar, si se tratase de un sistema real y no de un prototipo realizado con el fin de probar el sistema de recomendación, habría que separar el sistema en dos aplicaciones distintas, el front-end y el back-end.

Además, habría que mejorar el diseño de la página web, que se ha hecho con el único objetivo de que resultase sencilla de manejar y que fuese funcional, para darle un aspecto más profesional.

Por último, el sistema funciona adecuadamente cuando se manejan pocas votaciones, pero habría que mejorar los algoritmos para hacerlos mucho más rápidos cuando su número aumenta considerablemente.

8. BIBLIOGRAFÍA

- [1] Ali, Z., Khusro, S. y Ullah, I. (2016, May). A hybrid book recommender system based on table of contents (toc) and association rule mining. En *Proceedings of the 10th International Conference on Informatics and Systems*(pp. 68-74).
- [2] Barranco, M., Pérez, L. G. y Martínez, L. (2006). Un sistema de recomendación basado en conocimiento con información lingüística multigranular. En *Proceedings of the SIGEF XIII: Optimization techniques: Fuzziness and non linearity for management and economy*.
- [3] Burke, R. (2000). Knowledge-based recommender systems. *Encyclopedia of library and information systems*, 6 (Supplement 32), 175-186.
- [4] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4), 331-370.
- [5] Burke, R. (2007). Hybrid web recommender systems. En *The adaptive web* (pp. 377-408). Springer, Berlin, Heidelberg.
- [6] Burke, R., Felfernig, A. y Göker, M. H. (2011). Recommender systems: An overview. *Ai Magazine*, 32(3), 13-18.
- [7] Cadavid, A. N., Martínez, J. D. F. y Vélez, J. M. (2013). Revisión de metodologías ágiles para el desarrollo de software. *Prospectiva*, 11(2), 30-39.
- [8] Canós, J. H., Letelier, P. y Penadés, M. C. (2003). Metodologías ágiles en el desarrollo de software. *Universidad Politécnica de Valencia, Valencia*, 1-8.
- [9] Campos, L. M. de, Fernández-Luna, J. M., Huete, J. F. y Rueda-Morales, M. A. (2011). Uso de conocimiento estructurado en un sistema de recomendación basado en contenido. *Departamento de Ciencias de la computación e Inteligencia Artificial, Universidad de Granada, Granada, España*.
- [10] Castellano, E. J., Martínez, L., Barranco, M. y Pérez, L. (2007). Recomendación de perfiles académicos mediante algoritmos colaborativos basados en el expediente. En *Conferencia IADIS Ibero-Americana*

WWW/Internet 2007.

- [11] Chandak, M., Girase, S. y Mukhopadhyay, D. (2015). Introducing hybrid technique for optimization of book recommender system. *Procedia Computer Science*, 45, 23-31.
- [12] Chen, L., Chen, G. y Wang, F. (2015). Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction*, 25(2), 99-154.
- [13] Hanani, U., Shapira, B. y Shoval, P. (2001). Information filtering: Overview of issues, research and systems. *User modeling and user-adapted interaction*, 11(3), 203-259.
- [14] Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). *Recommender systems: an introduction*. Cambridge University Press.
- [15] Linden, G., Smith, B. y York, J. (2003). Amazon. Com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1), 76-80.
- [16] Meghabghab, G. y Kandel, A. (2008). *Search engines, link analysis, and user's web behavior: a unifying web mining approach* (Vol. 99). Springer Science& Business Media.
- [17] Melville, P. y Sindhvani, V. (2010). Recommender systems. *Encyclopedia of machine learning*, 1, 829-838.
- [18] Nelson, M. R. (1994). We have the information you want, but getting it will cost you! held hostage by information overload. *XRDS: Crossroads, The ACM Magazine for Students*, 1(1), 11-15.
- [19] Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artificial intelligence review*, 13(5-6), 393-408.
- [20] Porcel, C., Morales del Castillo, J. M., Cobo, M. J., Ruiz, A. A. y Herrera-Viedma, E. (2010). An improved recommender system to avoid the persistent information overload in a university digital library. *Control and Cybernetics*, 39(4), 899-923.
- [21] Porcel, C., Moreno, J. M. y Herrera-Viedma, E. (2009). A multi-disciplinar

recommender system to advice research resources in university digital libraries.

Expert Systems with Applications, 36(10), 12520-12528.

[22] Schwaber, K. y Sutherland, J. (2017). La guía de Scrum. La guía definitiva de Scrum: Las reglas del juego [en línea, noviembre].

[23] Xiao, B. y Benbasat, I. (2007). E-commerce product recommendation agents: use, characteristics, and impact. *MIS quarterly*, 31(1), 137-209.

Webgrafía

Sobre sistemas de recomendación de libros:

[24] <https://www.amazon.es>

[25] <https://www.goodreads.com>

[26] <https://www.thefussylibrarian.com/>

[27] <https://www.whatshouldireadnext.com/>

[28] <https://www.whichbook.net/>

Sobre sistemas de recomendación colaborativos:

[29] <https://www.filmaffinity.com>

Sobre datasets de libros:

[30] <https://www.kaggle.com/zygmunt/goodbooks-10k> goodbooks

[31] <http://www2.informatik.uni-freiburg.de/~ctiegl/BX/> bookCrossing

Sobre las herramientas utilizadas:

[32] <https://getbootstrap.com>

[33] <https://netbeans.org>

[34] <https://spring.io>

[35] <https://www.oracle.com/es/tools/technologies/netbeans-ide.html>

Índice de figuras:

Figura 2.1.: XML descriptivo del personaje de un videojuego.

Figura 2.2.: Logo de amazon.

Figura 2.3.: Información sobre libros comprados o vistos por los que ven un libro en amazon.com.

Figura 2.4.: Logo de goodreads.

Figura 2.5.: Recomendaciones por género en goodreads.

Figura 2.6.: Logo de TheFussyLibrarian.

Figura 2.7.: Logo de Whatshould I readnext?.

Figura 2.8.: Recomendaciones en What I shouldreadnext?.

Figura 2.9.: Logo de Whichbook.

Figura 2.10.: Controles deslizantes y recomendaciones en Wichbook.

Figura 3.1.: Diagrama de casos de uso del sistema.

Figura 3.2.: Modelo de dominio del sistema.

Figura 3.3.: Logo de Spring.

Figura 3.4.: Logo de HTML5.

Figura 3.5.: Logo de Bootstrap.

Figura 3.6.: Logo de Netbeans.

Figura 3.7.: Proceso de Scrum.

Figura 4.1.: Gráfico de la evolución de la iteración 1.

Figura 4.2.: Gráfico de la evolución de la iteración 2.

Figura 4.3.: Gráfico de la evolución de la iteración 3.

Figura 4.4.: Gráfico de la evolución de la iteración 4.

Figura 4.5.: Gráfico de la evolución del proyecto.

Figura 5.1.: Página de inicio.

Figura 5.2.: Página de registro.

Figura 5.3.: Registro de usuario (detalle).

Figura 5.4.: Registro de usuario. Mensaje de usuario ya existente (detalle).

Figura 5.5.: Registro de usuario. Mensaje de usuario creado (detalle).

Figura 5.6.: Página de portada.

Figura 5.7.: Icono y lema 1 de la aplicación.

Figura 5.8.: Icono y lema 2 de la aplicación.

Figura 5.9.: Parte izquierda del menú (detalle).

Figura 5.10.: Parte derecha del menú (detalle).

Figura 5.11.: Paso 1. Valorar los libros más valorados por el resto de usuarios.

Figura 5.12.: Valoración del interés del usuario en un libro (detalle).

Figura 5.13.: Paso 2. Número de recomendaciones a recibir.

Figura 5.14.: Paso 2. Número de recomendaciones a recibir (detalle).

Figura 5.15.: Paso 2. Opciones sobre el número de recomendaciones a recibir (detalle).

Figura 5.16.: Paso 3. Tipo de recomendaciones a recibir (detalle).

Figura 5.17.: Paso 3. Tipo de recomendaciones a recibir (detalle).

Figura 5.18.: Paso 3. Lista desplegable del tipo de recomendaciones a recibir (detalle).

Figura 5.19.: Lista de recomendaciones para el usuario.

Figura 5.20.: Lista de recomendaciones para el usuario (detalle).

Figura 5.21.: Lista de recomendaciones del usuario.

Figura 5.22.: Lista de libros valorados por el usuario.

Figura 5.23.: Información detallada de uno de los libros (detalle).

Figura 5.24.: Menú del usuario (detalle).

Figura 5.25.: Perfil del usuario.

Figura 5.26.: Perfil del usuario (detalle).

Figura 5.27.: Cambio del número de recomendaciones en el perfil (detalle).

Figura 5.28.: Cambio del tipo de recomendaciones en el perfil (detalle).

Figura 5.29.: Baja del sistema.

Figura 5.30.: Baja del sistema (detalle).

Figura 5.31.: Búsqueda de un libro

Figura 5.32.: Búsqueda de un libro (detalle)

Figura 5.33.: Búsqueda con el criterio "girl" (detalle).

Figura 5.34.: Resultado de la búsqueda con el criterio "girl" (detalle).

Figura 1.: Proyecto abierto con Netbeans.

Figura 2.: Creación de la base de datos.

Figura 3.: Credenciales de la base de datos.

Figura 4.: Archivo de creación de tablas de la base de datos.

Figura 5.: Ejecución del archivo Esquema.sql.

Figura 6.: Selección de la base de datos sobre la que ejecutar el archivo .sql.

Índice de fórmulas

Fórmula 2.1.: Similitud entre objetos basada en las palabras clave.

Fórmula 2.2.: TF frecuencia del término i .

Fórmula 2.3.: IDF inversa de la frecuencia del término i .

Fórmula 2.4.: TF-IDF.

Fórmula 2.5.: Coeficiente de Correlación de Pearson.

Índice de tablas:

Tabla 2.1.: Representación de un ítem mediante un vector de características.

Tabla 2.2.: Ejemplo de cálculo de TF, IDF y TF-IDF.

Tabla 2.3.: Frecuencia de los términos en los documentos.

Tabla 2.4.: Cálculo del TF en los documentos.

Tabla 2.5.: Cálculo del IDF en los documentos.

Tabla 2.6.: Cálculo del TF-IDF en los documentos.

Tabla 2.7.: Ejemplo de valoraciones de usuarios en un sistema de recomendación de películas.

Tabla 2.8.: Similitud entre usuarios en un sistema de recomendación de películas.

Tabla 3.1.: Caso de uso principal 1.

Tabla 3.2.: Caso de uso principal 2.

Tabla 3.3.: Caso de uso principal 3.

Tabla 3.4.: Caso de uso principal 4.

Tabla 3.5.: Caso de uso principal 5.

Tabla 3.6.: Caso de uso principal 6.

Tabla 3.7.: Caso de uso principal 7.

Tabla 3.8.: Caso de uso principal 8.

Tabla 3.9.: Caso de uso principal 9.

Tabla 3.10.: Caso de uso principal 10.

Tabla 4.1.: Historias de usuario relacionadas con la web.

Tabla 4.2.: Historias de usuario relacionadas con la documentación del proyecto.

Tabla 4.3.: Planificación de inicio-fin de los sprints del proyecto.

Tabla 4.4.: Resumen de sueldos de un analista trabajador.

Tabla 4.5.: Características del equipo informático.

Tabla 4.6.: Cálculo de la amortización del equipo informático.

Tabla 4.7.: Costes del proyecto.

Tabla 4.8.: Presupuesto total del proyecto.

Tabla 4.9.: Sprint backlog de la Iteración 1.

Tabla 4.10.: Sprint backlog de la Iteración 2.

Tabla 4.11.: Sprint backlog de la Iteración 3.

Tabla 4.12.: Sprint backlog de la Iteración 4.

Tabla 6.1.: Pruebas de caja negra de la página principal.

Tabla 6.2.: Pruebas de caja negra de la página de registro.

Tabla 6.3.: Pruebas de caja negra del menú de navegación.

Tabla 6.4.: Pruebas de registro de usuario.

Tabla 6.5.: Pruebas de acceso a la aplicación.

Tabla 6.6.: Pruebas de baja de usuario.

Tabla 6.7.: Pruebas de caja negra de confirmación de baja.

Tabla 6.8.: Pruebas de edición de perfil.

Tabla 6.9.: Pruebas de visualización de libros valorados.

Tabla 6.10.: Pruebas de visualización de recomendaciones.

Tabla 6.11.: Pruebas de recepción de recomendaciones.

Tabla 6.12.: Pruebas de cambio de valoración de un libro.

Tabla 6.13.: Pruebas de desconexión de la aplicación.

ANEXO. MANUAL DE INSTALACIÓN

Para poder ejecutar la aplicación, primero hay que crear la base de datos que utiliza la aplicación en Netbeans. Para ello hay que seguir una serie de pasos.

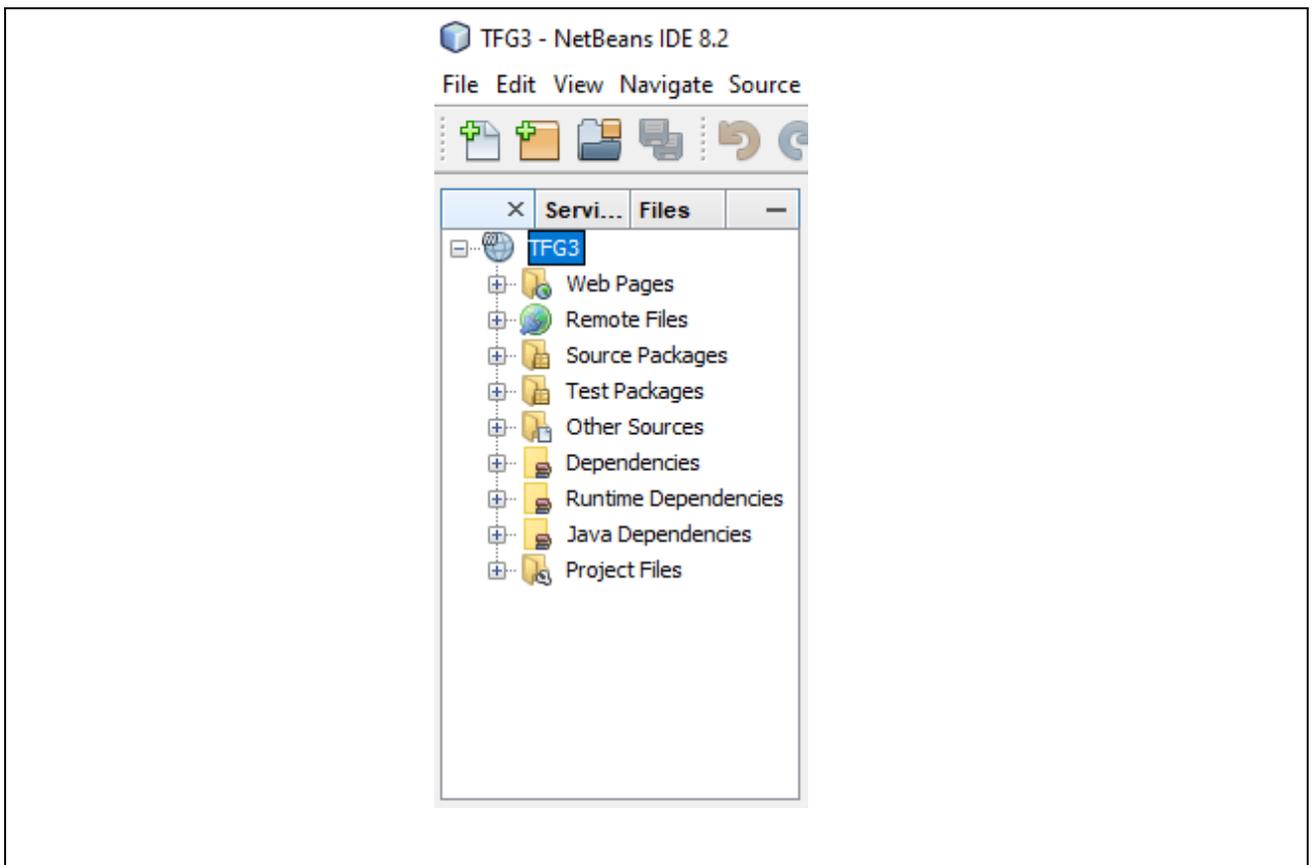


Figura 1.: Proyecto abierto con Netbeans. [Autora]

Tras descomprimir el proyecto, se abre el proyecto desde Netbeans. Una vez que ya se tiene el proyecto abierto, como se muestra en la figura 1., se selecciona la pestaña Services y hay que desplegar el menú Databases para poder visualizar la opción Java DB, sobre la que hay que pulsar el botón derecho del ratón y seleccionar Create Database, como se ve en la figura 2.

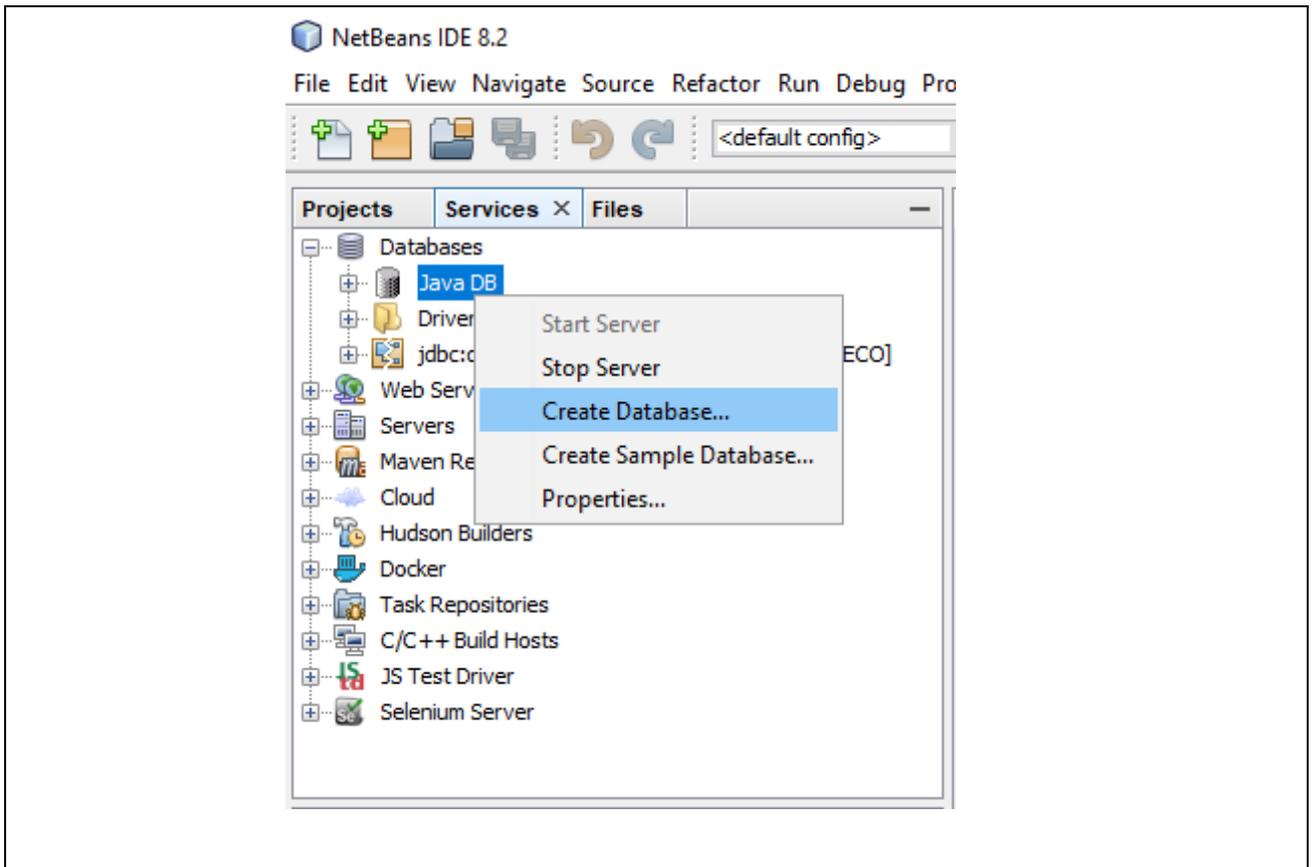


Figura 2.: Creación de la base de datos. [Autora]

A continuación, en la ventana que se abre, introduciremos la palabra "reco" en los campos nombre de la base de datos, usuario, password y su confirmación, como se muestra en la figura 3.

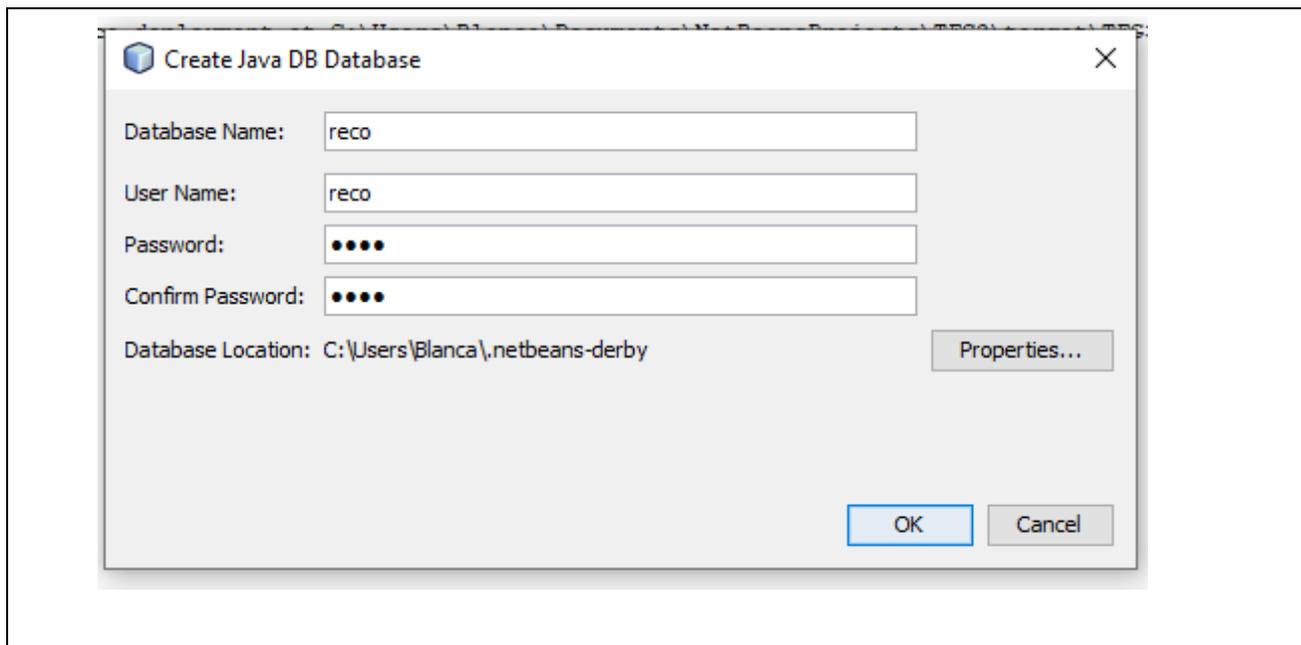


Figura 3.: Credenciales de la base de datos. [Autora]

Una vez creada la base de datos, hay que ejecutar el archivo esquema.sql que se encuentra dentro de la carpeta WEB-INF del proyecto, como se ve en la figura 4.

Desarrollar un sistema de recomendaciones de libros basado en memoria

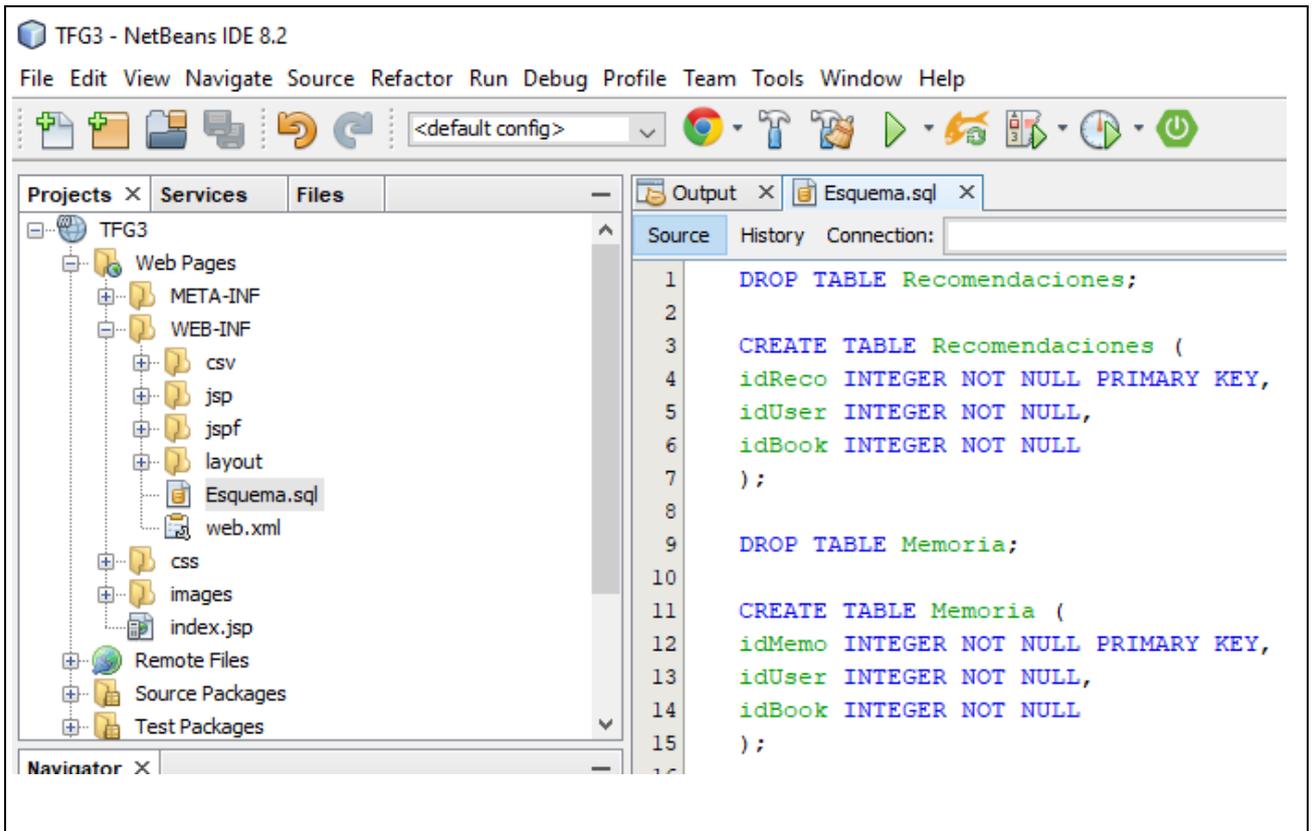


Figura 4.: Archivo de creación de tablas de la base de datos. [Autora]

Ahora se tendrá que ejecutar el archivo, pulsando en el botón Execute SQL, como muestra la figura 5.

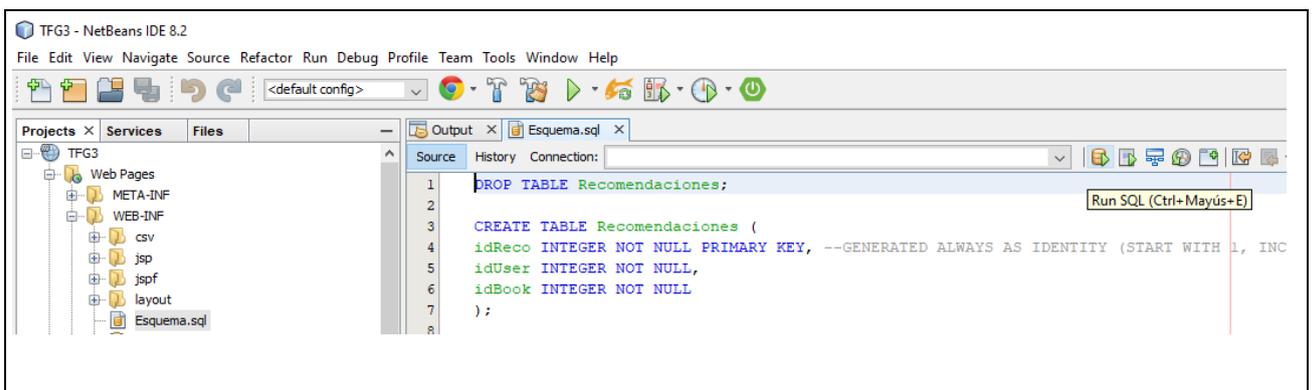


Figura 5.: Ejecución del archivo Esquema .sql. [Autora]

Se nos pedirá que seleccionemos una base de datos de la lista sobre la que ejecutar el archivo y seleccionaremos la base creada anteriormente, como

se muestra en la figura 6.

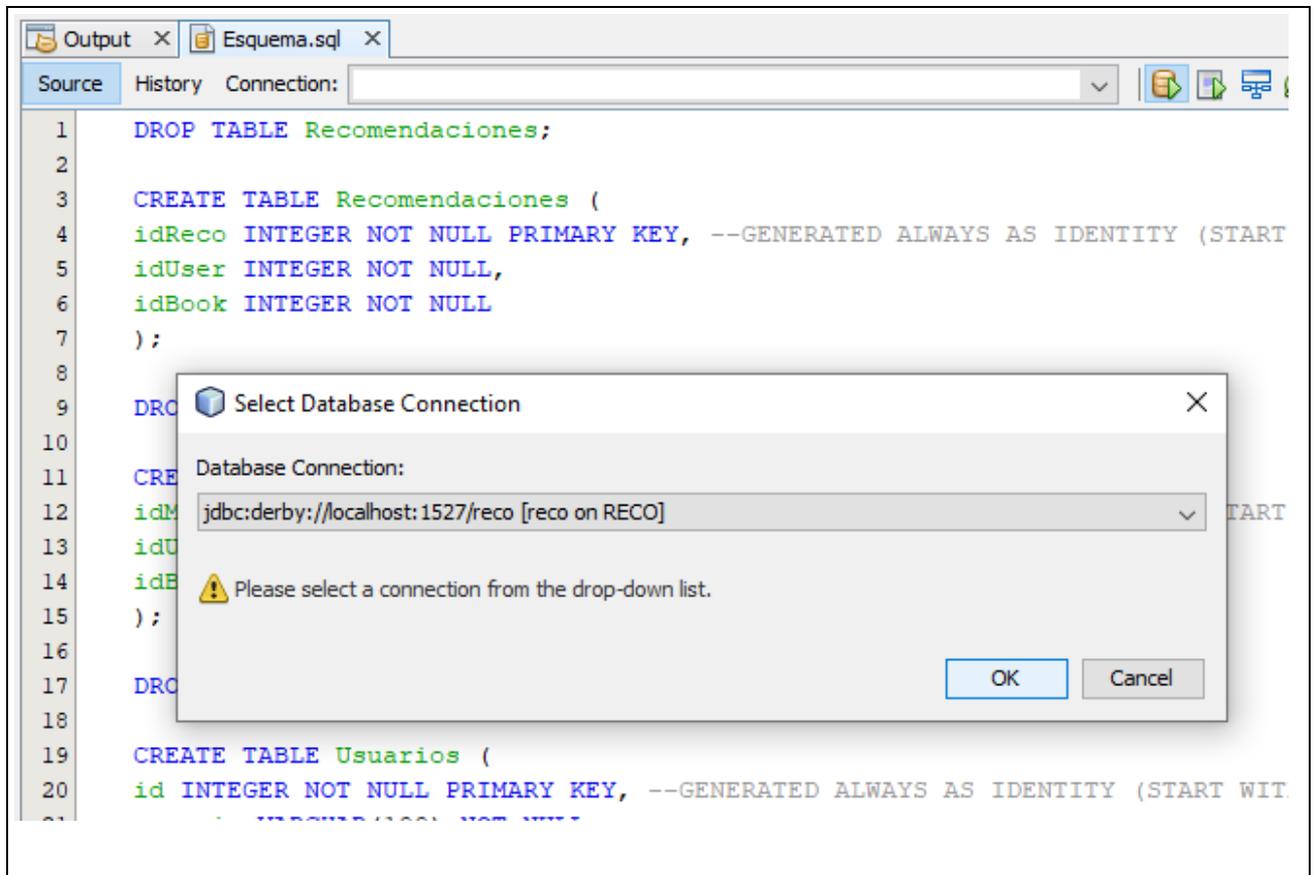


Figura 6.: Selección de la base de datos sobre la que ejecutar el archivo .sql. [Autora]

Y una vez hecho esto ya podremos utilizar la aplicación con la conexión a esta base de datos.