

UNIVERSIDAD CARLOS III DE MADRID

Escuela Politécnica Superior

**DESARROLLO DE UN
METABUSCADOR PARA LA
BÚSQUEDA DE APARTAMENTOS
DE ALQUILER HABITUAL**

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN



**Universidad
Carlos III de Madrid**
www.uc3m.es

Autor: Jaime Fernando Machasilla Sánchez

Tutor: Rubén Cuevas Rumín

Septiembre 2015

AGRADECIMIENTOS

En primer lugar me gustaría agradecer a mi familia por su apoyo y ánimo mostrados durante el transcurso de mi vida y, especialmente, durante el desarrollo de este trabajo.

A mi tutor Rubén por toda su ayuda prestada.

Índice de Contenido

Abstract	8
Resumen.....	14
Capítulo 1 INTRODUCTION / INTRODUCCIÓN.....	20
1.1 Introduction.....	20
1.1.1 Context	20
1.1.2 Motivation.....	20
1.1.3 Goals.....	21
1.1.4 Document structure	22
1.2 Introducción	23
1.2.1 Contexto	23
1.2.2 Motivación.....	23
1.2.3 Objetivos	24
1.2.4 Estructura del documento.....	25
Capítulo 2 ESTADO DEL ARTE	26
2.1 Origen y situación actual de la Web.....	26
2.2 Buscadores	28
2.2.1 Origen y situación actual de los buscadores.....	28
2.2.2 Cronología de los buscadores	31
2.2.3 Clases de los buscadores	34
2.2.4 Comparadores	37
2.2.5 Casos de éxito.....	37
2.3 Aplicaciones web.....	38
2.3.1 ¿Qué es una aplicación web?.....	39
2.3.2 Origen y situación actual de las aplicaciones web.	39
2.3.3 Tecnologías.....	40
Capítulo 3 DISEÑO E IMPLEMENTACIÓN.....	46
3.1 Visión general de la solución.....	46
3.2 Plataforma de desarrollo.....	47
3.3 Estructuración del código.....	48
3.4 Tecnología empleada	49
3.4.1 Client-side.....	49
3.4.2 Server-side.....	52
3.4.3 Apache Tomcat.....	57

3.4.4 Base de datos	58
3.4.5 Jsoup.....	61
Capítulo 4 RESULTADOS	62
4.1 Página principal.....	62
4.2 Casos de uso.....	63
4.2.1 Introducción correcta del municipio.....	63
4.2.2 Introducción correcta del municipio y precio máximo	64
4.2.3 Introducción correcta del municipio, precio máximo y número de habitaciones	64
4.2.4 Introducción correcta de todos los campos sin resultados	65
4.2.5 Introducción errónea de datos.....	66
Capítulo 5 PLANIFICACIÓN Y PRESUPUESTO	67
5.1 Ciclo de vida del trabajo	67
5.2 Fases de desarrollo.....	68
5.3 Presupuesto	68
5.3.1 Coste de personal.....	68
5.3.2 Coste de material	69
5.3.3 Resumen de costes.....	69
Capítulo 6	70
CONCLUSIONS AND FUTURE LINES / CONCLUSIONES Y LÍNEAS FUTURAS	70
6.1 Conclusions and future lines	70
6.1.1 Conclusions.....	70
6.1.2 Future lines.....	71
6.2 Conclusiones y líneas futuras	72
6.2.1 Conclusiones.....	72
6.2.2 Líneas futuras	73
Capítulo 7 REFERENCIAS.....	76

Índice de figuras

Figure 1. Access to information through the Internet.....	9
Figure 2. Flow of execution of the metasearch engine.....	9
Figura 3. Acceso a la información de través de Internet.....	13
Figura 4. Flujo de ejecución del metabuscador.....	15
Figura 5. Los cuatro nodos iniciales de ARPANET a finales de 1969.....	24
Figura 6. Número de servidores web en el mundo.....	25
Figura 7. Ejemplo de la interfaz gráfica del navegador Mosaic.....	26
Figura 8. Cuota de mercado de los principales buscadores entre enero y agosto de 2015.....	27
Figura 9. Tamaño de Google.....	28
Figura 10. Tamaño de Bing.....	29
Figura 11. Clasificación de los principales lenguajes de programación. Septiembre de 2015.....	40
Figura 12. Modelo de funcionamiento de la aplicación web.....	45
Figura 13. Logotipo del sistema operativo Ubuntu 14.04 Trusty Tahr.....	45
Figura 14. Estructuración del código.....	46
Figura 15. Fragmento del documento index.html.....	47
Figura 16. Modelo clásico de aplicaciones web vs. modelo Ajax de aplicaciones web.....	48
Figura 17. Fragmento de la función createList.....	49
Figura 18. Fragmento de la función select.....	49
Figura 19. Fragmento de la función _setup.....	49
Figura 20. Función autocompletar.....	49
Figura 21. Fragmentos de código del archivo basic.css.....	49
Figura 22. Fragmento de código del archivo resultado.jsp.....	50
Figura 23. Fragmento de código del archivo AutoComplete.java.....	51
Figura 24. Fragmento de código del archivo Buscar.java.....	52
Figura 25. Fragmento de código del archivo DownloadToMysql.java.....	52
Figura 26. Fragmentos de código del archivo LlamadaJS.java.....	53
Figura 27. Fragmento de código del archivo LlamadaJS_30first.java.....	54
Figura 28. Fragmento de código del archivo ObtenerImgs.java.....	54
Figura 29. Apache Tomcat.....	55
Figura 30. Configuración del pool de conexiones en el archivo web.xml.....	57
Figura 31. Configuración del pool de conexiones en el archivo context.xml.....	57
Figura 32. Configuración del pool de conexiones en el archivo Manager.java.....	57
Figura 33. Tablas contenidas en la base de datos relacional.....	58
Figura 34. Página de bienvenida de la aplicación web DreamHome.....	60
Figura 35. Alternativas en los campos precio máximo y número de habitaciones.....	61
Figura 36. Introducción del municipio. Muestra de resultados.....	61
Figura 37. Introducción del municipio y precio máximo. Muestra de resultados.....	62
Figura 38. Introducción del municipio, precio máximo y nº de habitaciones. Muestra de resultados.....	63
Figura 39. Introducción correcta de datos. No se encuentran resultados.....	64
Figura 40. Introducción errónea de datos (opción A parte superior izquierda, opción B parte superior derecha).....	64

Figura 41. Ciclo de vida en cascada del trabajo.....65
Figura 42. Diagrama de Gantt del metabuscador.....68

Abstract

Internet emerged in 1969 as a research project of the United States Department of Defense (DoD) in which three universities in California and one in Utah were joined with the goal of establishing two-way communication between them without stopping the system from working in the face of military attacks. In the beginning it bore the nickname of ARPANET (Advanced Research Project Agency Network), which followed a development and growth notable after the integration of different networks. Later it was renamed the *Internet*, as it is known today. The Internet is often referred as the network of networks, surpassing all forecasts and constituting a real revolution in society due to its essential and great utility. It is the cornerstone of communications, commerce and entertainment around the world.

In 1989 Tim Berners-Lee launched the *Enquire* system in order to store small portions of information linked together, allowing access of them through the Internet using HTTP (HyperText Transfer Protocol). It was the timid beginning of the World Wide Web (WWW or Web). A short time after appeared the HTML language which together with WWW are protocols and essential tools for remote queries at hypertext files.

Very few people can differentiate the concepts of Internet and the Web because these terms are often used interchangeably. The Web is the largest information network in the world but the service that provides access to it is the Internet. Really the Web is only one part because it coexists with other protocols such as SMTP, SSH, Telnet and FTP, among others.

The amount of information stored on the Web grows exponentially every year due to the technology in which it is based is constantly developing thanks to the research and innovation of people that improve what is already established. As a consequence, the continued rise of users and connections are very noticeable on the Web. It is the law of life.

The idea of being able to access information through the Internet is very attractive but not long ago it was impossible to do because there were no tools that allowed. Therefore, the development of a mechanism that is able to connect to the Web is essential. With this objective the browsers and search engines were born.

In 1993 Marc Andreessen implemented the first commercial graphical browser called *Mosaic*. The integration of the mouse and images to the text made the starting point for such a remarkable career that has continued to the present day.

Once the browsers were implemented and with it the access to Internet became necessary to create a tool to search for information of any type on the Web, and through this search engines were born. Its operation is based on repeated consultation of their own database which store the information obtained from the Web and display results to the user. The first developments based their searches in URLs, later appeared directories and search engines that perform queries using keywords or titles. The first, Wandex and Aliweb, were developed in 1993. In 1997 Google came out, a year later Ask and finally Microsoft launched Bing in 2009. The most used worldwide are Google, Baidu (in China) and Bing.

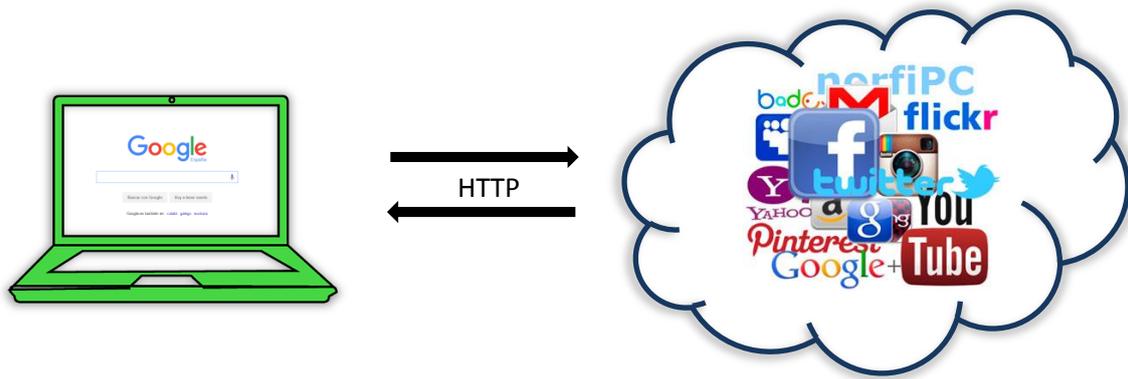


Figure 1. Access to information through the Internet.

As expected with the passage of time have emerged different types of engines have emerged that perform different functionalities because not all people search with the same way. The only thing all of them contemplate is the ability to track and obtain information from the Web. Among the most important are directories, search engines, mixed systems that combine the previous two, verticals and finally metasearch engines. A special case of metasearch engines are the comparators, web applications with high demand in sectors such as tourism, the automotive and housing.

As the name implies, the comparators are applications that allow collection of information from various search portals with certain requirements set by the user, compares and displays the best adverts. An example of this is Kayak. It is a metasearch engine that focuses on providing online travels and compares flights, hotels, rental cars and vacation packages on Booking.com, Atrápalo, Villas.com, Vueling, Iberia and eDreams, among others.

It is important to note that the housing sector is regaining strength after being one of those hardest hit by the economic crisis in recent years in Spain, although his timid recovery began from the second half of 2013. Business opportunity in this sector has been detected and we must seize it, so it was decided to generate a web application that fulfills the function of metasearch engine and dedicates itself to gather adverts about apartments for habitual rent. Most users who look for this type of stay do not want to invest in flats with a mortgage; therefore these applications whose main goal should be to facilitate the search are never too much.

After deciding to develop and deploy a web application it is necessary to mark certain objectives to make their performance as demanded by users. It has been established as the main objective to be able to perform certain queries in predetermined search portals. The final version will collect adverts from *Fotocasa* (fotocasa.es) and *Idealista* (idealista.com). It must be an application with a simple, intuitive and easy use user interface so that it can be used by any type of user. The rest of objectives that implicitly take the last two are not to fail in queries performed, to allow the introduction of location, maximum price and number of rooms, correct use and storage of information in the local database, finalize these queries with the

respective presentations of advertisements and finally be operable in Google Chrome and Firefox browsers.

These tools never cease to be applications and therefore are composed by a set of programs that can perform predetermined functionalities on the Web. Websites are used as a mechanism for communication between the user and the application. This performance is known as client-server architecture.

This type of web applications concept is not really new because before the Internet is completely accessible worldwide there was a programming language that permitted their development. This is the Perl language, invented in 1987. In 1995 the PHP programming language appeared. It is widely used today in important applications. Java and JavaScript were developed in the same year. JavaScript allows the dynamic change of the content of web pages making them more attractive to the user. Among the most successful applications on the web are Google which was released in 1998, Wikipedia in 2001, Facebook in 2004 and Twitter in 2006. The development of these applications is booming and has a great future ahead.

Another aspect of great importance is the way in which they are distributed over the web. They are stored on web servers that allow access to them through the Internet. Among the most used are Apache, Microsoft IIS and GlassFish. Each brings special features used for a common purpose.

As mentioned above, search engines collect any type of information accessible in the web. As you can imagine they manipulate huge amounts of data and require a tool to store them in an orderly manner in databases. Therefore there are management systems that store, modify and extract information from them. Among the most used are MySQL, Microsoft SQL Server and PostgreSQL. Each provides certain features that make them special.

The development of this work has been influenced by the previously discussed technologies. After assessing the advantages and disadvantages offered by each of them it has been decided to create a web application programmed in JavaScript and Java Servlets (Java programs), based on Apache tomcat (special distribution of Apache) as HTTP web server and composed by MySQL as a database manager system.

The performance of the application is itself a metasearch of rental apartments that does not have a database maintained by direct user intervention, in which the customer is who can access and perform a determined consultation based on the location where there are adverts and filters as the maximum price and the number of rooms.

First, the browser displays the user interface containing the logo of the application and a form to be filled. Once the user has entered all the data and proceeds to perform a search, the web server delegates certain functions to each of the Java Servlets programmed.

Specifically it proceeds to download information from the source pages previously commented. Once this download is complete the data obtained are treated, handled and

inserted into the local database. This is what differentiates with search engines due to that this local database is not supplied by users; in other words, they cannot have direct access to manipulate it. Finally, after having inserted the data the last step is to display the results in JSPs (JavaServer Pages).

The following figure shows the flow of execution that has the metasearch engine developed:

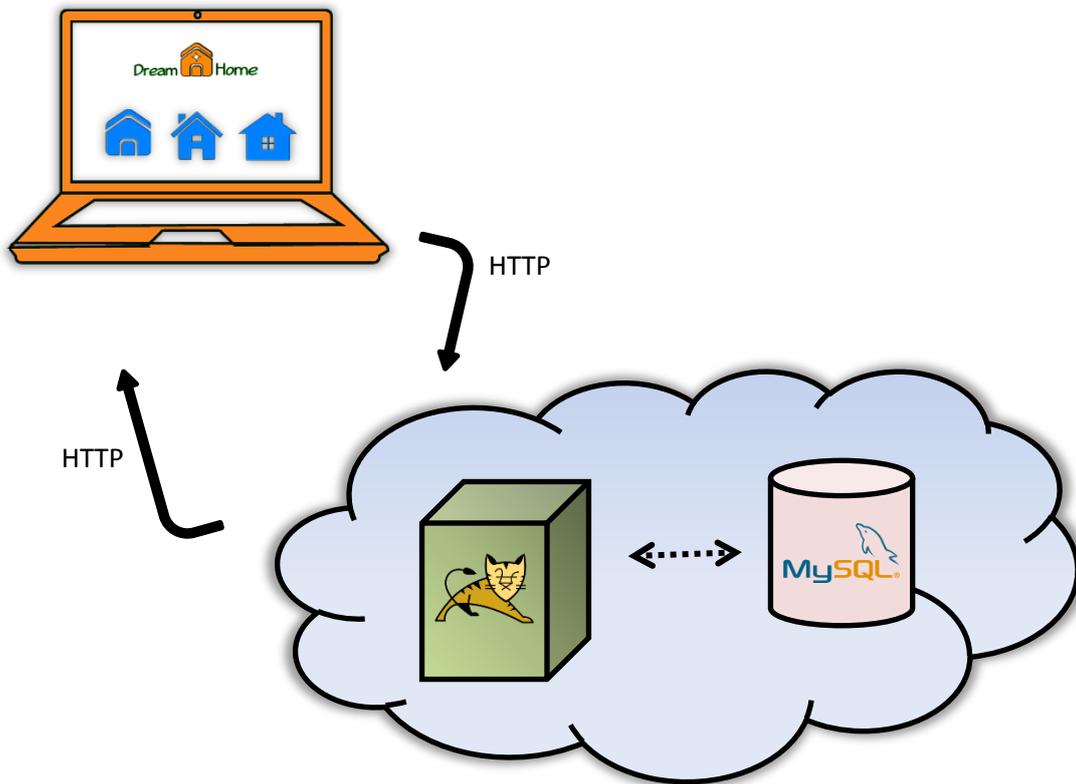


Figure 2. Flow of execution of the metasearch engine.

For the development and implementation of this metasearch engine called *DreamHome*, Ubuntu 14.04 LTS Trusty Tahr, a Linux distribution, has been used. The console of the operating system allows it to compile and run Java programs, taking previously installed the JDK (Java Development Kit). For this reason there is no need integrated development environments (IDEs) such as Eclipse, although it must be recognized that this software is highly used due to its potential, easy use and good performance that is capable of providing for developers.

To correctly obtain all the results it has been necessary to program various servlets that take in effect the determined function. In concrete it has been created the following servlets: `Autocomplete.java`, `Buscar.java`, `DownloadToMysql.java`, `LlamadaJS.java`, `LlamadaJS_30first.java` and `ObtenerImgs.java`. Also a package has been designed in which contains the necessary archives for the communication between Apache Tomcat and

MySQL. This communication contemplates the connection/disconnection with the database and the manipulation and insertion of data. All of these functionalities are gathered in `archiveManager.java` and should be included in the servlets if they need to for any motive access the database. Also the file `Vivienda.java` has been elaborated that performs the function of `JavaBean` and represents each downloaded advertisement. In this way the insertion of the database is organized.

Between the previously commented servlets highlight `LlamadaJS.java` and `LlamadaJS_30first.java`. The first of them is in charge of parse the HTML code of *fotocasa.es* and *idelist.com* to obtain the advertisements of the selected town for the user. This technique is commonly known as *Web Scraping*. The second file performs the same as the first but obtains a maximum of 30 first adverts of each webpage and their information is stored in files with the JSON (JavaScript Object Notation) format. Once these files are generated, these are read by this application to quickly show the adverts to the user while the complete download is performed in the background, totally transparent for the user.

Also various JSPs were implemented for deployment of adverts from a particular locality. These files are executed in the browser and to make them more attractive for the user have been programmed the files `busqueda_listas.js` and `basic.css`. The first of them allows dynamically adding titles, displaying the list with the images and their descriptions, displaying animations, handling events and making connections with other servlets. The second is responsible for improving the presentation by colors and predetermined forms to the elements contained in the documents.

Finally, as mentioned earlier the application needs a database to store some of the information that finds. MySQL has been used to define a database and within three tables. One of them contains all provinces of the Spanish territory, other has the towns and the last is specific to storing advertisements.

Once all the tools and programs necessary for the proper performance of this application are obtained, it proceeds to run it.

When this application is running and the user accesses it through his browser, he will find a webpage that should fill form elements. He must necessarily enter the town, the filters of the maximum price and number of rooms are optional.

With the previously mentioned data could be made different search types (called “use cases”) and therefore the application would end their execution and would return different results depending on what the user select.

Like any work or project the development of this metasearch engine it has passed through various phases called life cycle, from the study of the problem to the presentation of the final product. All work requires advance planning. Specifically, this application follows a cascade life cycle with feedback as always could return to an earlier phase to change or improve functionalities already established. *ProjectLibre*, a project management tool, is used to generate Gantt diagram to view estimates of length of each of the phases.

For the development of this metasearch engine are needed tools both software and hardware and requires a group of people (staff) that is capable of carrying out it. Only two people are needed, the tutor who plays the role of client and the student that makes the role of project manager, analyst, designer and programmer. It is estimated that the dedication in this work is 560 hours in 4,5 months. Therefore, staff costs are 14.035 € and material costs are 117,8 €. Total costs of this application are 14.152,8 €.

In the final part of this document it's performed a little reflection on the conclusions of the student after to have developed and implemented the metasearch engine, and possible improvements that could be added in the future.

Resumen

Internet surgió en 1969 como un proyecto de investigación del Departamento de Defensa Americano (United States Department of Defense, DoD) en el cual se unieron tres universidades en California y una en Utah con el objetivo de establecer una comunicación bidireccional entre ellas sin que el sistema deje de funcionar ante ataques militares. En sus inicios llevaba el sobrenombre de ARPANET (Advanced Research Project Agency Network), el cual siguió un desarrollo y un crecimiento muy notables tras la integración de diversas redes. Posteriormente fue renombrado *Internet*, tal como es conocida en la actualidad. Suele denominarse a *Internet* como la red de redes, superando cualquier previsión y constituyendo una verdadera revolución en la sociedad debido a su esencial y gran utilidad. Es el pilar de las comunicaciones, comercio y entretenimiento en todo el mundo.

En 1989 Tim Berners-Lee puso en marcha el sistema *Enquire* con el objetivo de almacenar pequeñas porciones de información conectadas entre sí, posibilitando el acceso a ellas a través de Internet haciendo uso de HTTP (HyperText Transfer Protocol). Fue el tímido comienzo de la World Wide Web (WWW o la Web). Poco después apareció también el lenguaje HTML, que junto a la WWW conforman los protocolos y herramientas imprescindibles para realizar consultas remotas de archivos de hipertexto.

Muy poca gente puede diferenciar los conceptos de Internet y de la Web, pues son términos que suelen usarse indistintamente. La Web es la mayor red de información en el mundo pero el servicio que posibilita el acceso a ella es Internet. Realmente la Web es sólo una parte de Internet pues coexiste con otros protocolos como el SMTP, SSH, Telnet y FTP, entre otros.

La cantidad de información almacenada en la Web crece exponencialmente cada año pues la tecnología en la que está basada se desarrolla continuamente gracias a la investigación e innovación de personas dedicadas a mejorar lo ya establecido. Como consecuencia es apreciable el aumento continuo de usuarios y conexiones presentes en la Web. Es la ley de la vida.

La idea de poder acceder a información a través de Internet es muy atractiva pero hasta hace no mucho era imposible hacerlo pues no había herramientas que lo permitan. Por ello se hizo imprescindible el desarrollo de un mecanismo que sea capaz de conectarse a la Web. Con este objetivo nacieron los navegadores y en concreto los buscadores.

En 1993 Marc Andreessen implementó el primer navegador gráfico comercial denominado *Mosaic*. La integración del ratón y las imágenes al texto hicieron de este navegador el punto de partida para la tan notable trayectoria que han seguido hasta nuestros días.

Una vez se implementaron los navegadores y con ella el acceso a Internet surgió la necesidad de crear una herramienta que permita buscar información de cualquier tipo en la Web, por ello nacieron los buscadores. Su funcionamiento se basa en consultar reiteradas

veces sus propias bases de datos en las cuales almacenan la información que obtienen de la Web y mostrar los resultados al usuario. Los primeros desarrollados basaban sus búsquedas en direcciones URL, posteriormente aparecieron los directorios y los motores de búsqueda que realizan consultas mediante títulos o palabras clave. Entre los primeros buscadores se encuentran Wandex y Aliweb desarrollados en 1993. En 1997 salió a la luz Google, un año más tarde Ask y finalmente en 2009 Microsoft lanzó Bing. Los más usados en todo el mundo son Google, Baidu (en China) y Bing.

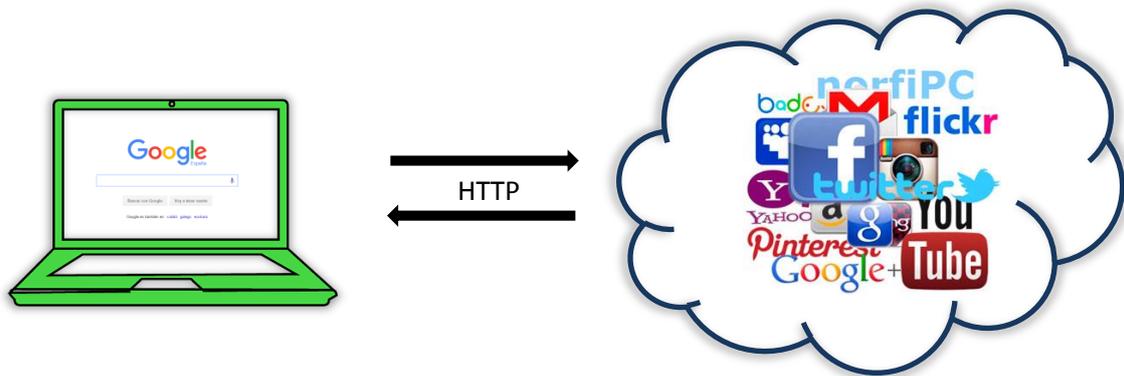


Figura 3. Acceso a la información a través de Internet.

Como era de esperar con el paso del tiempo han ido surgiendo distintos tipos de buscadores que realizan acometidos diferentes pues no todas las personas realizan búsquedas de la misma manera. Lo único en lo que sí coinciden absolutamente todos los buscadores es la capacidad de rastrear y obtener información de la Web. Entre los más importantes se pueden destacar a los directorios, motores de búsqueda, sistemas mixtos que combinan los dos anteriores, los verticales y finalmente los metabuscadores. Un caso concreto de éstos últimos son los comparadores, aplicaciones web con mucha demanda en sectores como el turístico, el de automoción y el inmobiliario.

Como el propio nombre lo dice, los comparadores son aplicaciones que permiten recopilar información de distintos portales búsqueda con ciertos requisitos establecidos por el usuario, los compara y muestra las mejores ofertas, como es el caso de Kayak. Se trata de un metabuscador que se centra en ofrecer viajes online, además de comparar vuelos, hoteles, coches de alquiler y paquetes vacacionales presentes en Booking.com, Atrápalo, Villas.com, Vueling, Iberia o eDreams, entre otros.

Es importante destacar que el sector inmobiliario está recobrando fuerzas tras haber sido uno de los que más ha sufrido la crisis económica durante los últimos años en España, aunque su tímida recuperación empezó a partir de la segunda mitad de 2013. Se ha detectado una oportunidad de negocio en este sector y hay que aprovecharla, por ello se ha decidido generar una aplicación web que cumpla la función de metabuscador y se dedique a recopilar ofertas sobre apartamentos de alquiler habitual. La mayoría de usuarios que buscan este tipo de estancia ya no desean invertir en pisos con una hipoteca y por ello estas aplicaciones, cuyo mayor objetivo debe ser el de facilitar la búsqueda, nunca están demás.

Tras decidir desarrollar e implementar una aplicación web es necesario marcar ciertos objetivos para que su funcionamiento sea el demandado por los usuarios. Se ha establecido como objetivo principal que sea capaz de realizar consultas predeterminadas en ciertos portales de búsqueda. La versión final tomará como fuente los anuncios de *Fotocasa* (fotocasa.es) e *Idealista* (idealista.com). Debe ser una aplicación con una interfaz de usuario sencilla, intuitiva y de fácil uso para que pueda ser utilizada por cualquier tipo de usuarios. El resto de los objetivos que llevan implícitamente los dos anteriores son: no fallar al realizar consultas, permitir la introducción de la localidad, precio máximo y número de habitaciones, correcto uso y almacenamiento de información en la base de datos local, finalizar las consultas con las respectivas presentaciones de anuncios y, por último, ser operable en los navegadores Google Chrome y Firefox.

Estas herramientas no dejan de ser aplicaciones y por tanto están compuestas por un conjunto de programas capaces de llevar a cabo un acometido predeterminado en la Web. Utilizan páginas web como mecanismo para establecer la comunicación entre el usuario y la aplicación. Esta funcionalidad es conocida como arquitectura cliente-servidor.

El concepto de este tipo de aplicaciones web no es realmente nuevo pues antes de que Internet sea completamente accesible en todo el mundo existía un lenguaje de programación que ya permitía desarrollarlas. Se trata del lenguaje Perl, inventado en 1987. En 1995 se dio un gran paso más allá con la aparición del lenguaje de programación PHP, muy utilizado en la actualidad en grandes e importantes aplicaciones. Además del desarrollo de Java en este mismo año apareció JavaScript que permite cambiar dinámicamente el contenido de las páginas web haciéndolas más atractivas al usuario. Entre las aplicaciones con mayor éxito en la Web son Google, que vio la luz en 1998, Wikipedia en 2001, Facebook en 2004 y Twitter en 2006. El desarrollo de este tipo de aplicaciones está en auge y cuentan con un gran futuro por delante.

Otro aspecto de gran relevancia es la forma en la que están distribuidas en la Web. Se encuentran almacenadas en servidores web que permiten el acceso a ellas a través de Internet. Entre los más usados se encuentran Apache, Microsoft IIS y GlassFish. Cada uno de ellos reúne características especiales que se usan para un fin común.

Como se ha comentado anteriormente, los buscadores recopilan todo tipo de información accesible en la Web. Como es de imaginar éstos manipulan enormes cantidades de datos y precisan de una herramienta que permita guardarlos de una manera ordenada en bases de datos. Por ello existen los sistemas gestores que permiten almacenar, modificar y extraer información de dichas bases. Entre los más usados se encuentra MySQL, Microsoft SQL Server y PostgreSQL. Cada uno proporciona ciertas funcionalidades que les hace particulares.

El desarrollo de este trabajo se ha visto influido por las tecnologías anteriormente comentadas. Tras valorar todas las ventajas e inconvenientes que ofrecen cada una de ellas se ha apostado por generar una aplicación web programada en JavaScript y Java Servlets (programas escritos en Java), basada en Apache Tomcat (distribución especial de Apache) como servidor web HTTP y compuesta por MySQL como sistema gestor de bases de datos.

El funcionamiento de la aplicación es propio de un metabuscador de apartamentos de alquiler que carece de una base de datos alimentada por intervención directa de los usuarios, en el cual el cliente es el que accede y realiza una consulta determinada basada en la localidad que desea encontrar anuncios y filtros como el precio máximo y el número de habitaciones.

En primer lugar el navegador muestra al cliente la interfaz de usuario que contiene el logotipo de la aplicación y un formulario a ser rellenado. Una vez que el usuario ha introducido todos los datos y procede a realizar la búsqueda el servidor web delega ciertas funcionalidades a cada uno de los Java Servlet programados.

En concreto se procede a descargar información de las páginas fuente anteriormente comentadas. Una vez se completa dicha descarga los datos que se obtienen son tratados, manipulados e insertados en la base de datos local. Esto último es lo que le diferencia de un buscador pues aunque cuenta con una base de datos local no es alimentada por los usuarios, es decir, éstos no pueden acceder directamente a ella para manipularla. Finalmente, tras haberse insertado los datos se procede al despliegue de los resultados en páginas JSP (JavaServer Pages).

En la siguiente figura se puede apreciar el flujo de ejecución que tiene el metabuscador desarrollado:

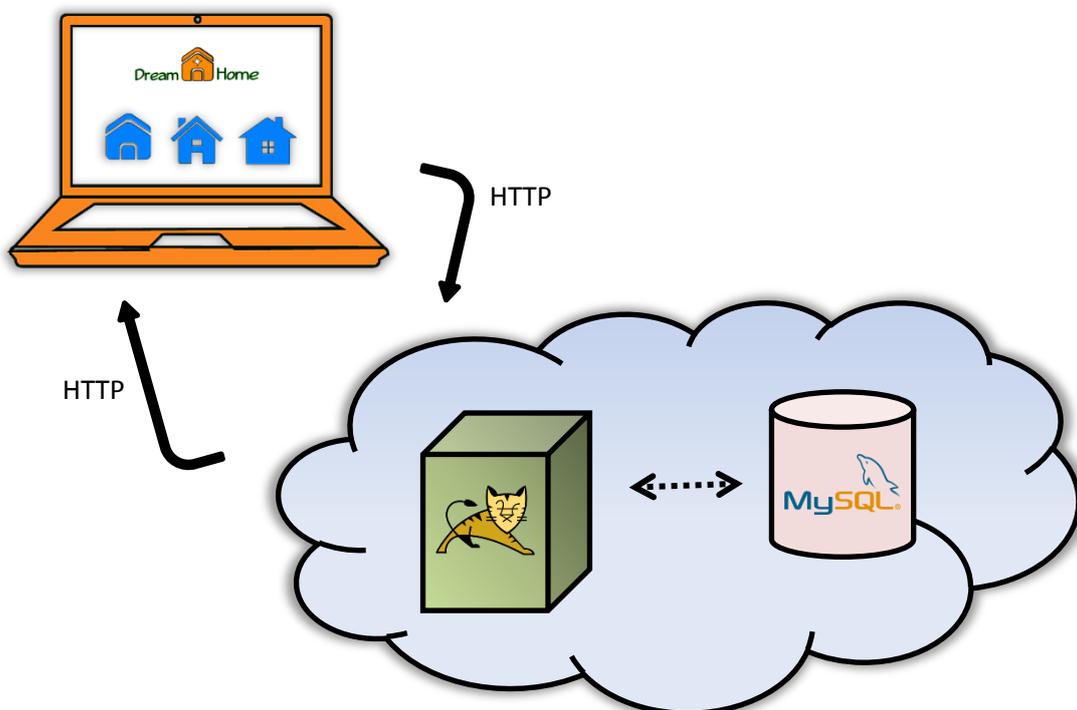


Figura 4. Flujo de ejecución del metabuscador.

Para el desarrollo e implementación de este metabuscador denominado DreamHome se ha utilizado Ubuntu 14.04 LTS Trusty Tahr, una distribución de Linux. La consola de este

sistema operativo permite compilar y ejecutar programas Java, teniendo previamente instalado el JDK. Por este motivo no se ha necesitado de entornos de desarrollo integrados (Integrated Development Environment, IDE) como Eclipse, aunque se debe reconocer que este software es altamente utilizado debido a su potencial, facilidad de uso y a las buenas prestaciones que es capaz de brindar a desarrolladores.

Para obtener correctamente todos los resultados ha sido necesario programar varios servlets que lleven a cabo una función determinada. En concreto se ha precisado los siguientes servlets: `Autocomplete.java`, `Buscar.java`, `DownloadToMysql.java`, `LlamadaJS.java`, `LlamadaJS_30first.java` y `ObtenerImgs.java`. También se ha diseñado un paquete (package) el cual contiene archivos necesarios para la comunicación entre Apache Tomcat y MySQL. Esta comunicación contempla la conexión/desconexión con la base de datos y la manipulación e inserción de datos. Todas estas funcionalidades están reunidas en un único archivo denominado `Manager.java` y debe ser incluido en dichos servlets si necesitan acceder a la base de datos por cualquier motivo. También se ha elaborado el archivo `Vivienda.java` que desempeña la función de JavaBean y representa a cada anuncio descargado. De esta manera la inserción en la base de datos es ordenada.

Entre los servlets anteriormente comentados hay que destacar a `LlamadaJS.java` y `LlamadaJS_30first.java`. El primero de ellos es el encargado de parsear el código HTML de fotocasa.es e idealista.com para obtener los anuncios del municipio seleccionado por el usuario. Esta técnica es conocida comúnmente como *Web Scraping*. El segundo fichero realiza lo mismo que el primero pero obtiene como máximo los 30 primeros anuncios de cada portal y su información es almacenada en archivos con formato JSON (JavaScript Object Notation). Una vez generados estos archivos son leídos por la aplicación para mostrar rápidamente los anuncios al usuario mientras la descarga completa se realiza en segundo plano, totalmente transparente al mismo.

También se han implementado varios JSPs para el despliegue de los anuncios de una localidad concreta. Estos documentos son ejecutados en el navegador y para hacerlos más atractivos de cara al usuario se han programado el script `busqueda_listas.js` y el archivo `basic.css`. El primero de ellos permite de forma dinámica añadir títulos, mostrar la lista de anuncios con las imágenes y sus respectivas descripciones, desplegar animaciones, manejar eventos y realizar conexiones con los distintos servlets. El segundo se encarga de mejorar la presentación dando colores y formas predeterminadas a los elementos contenidos en los documentos.

Finalmente, como se ha comentado con anterioridad la aplicación necesita una base de datos para almacenar parte de la información que encuentra. Se ha utilizado MySQL con el cual se ha podido definir una base de datos y dentro de ella tres tablas. Una de ellas contiene todas las provincias del territorio español, otra los municipios y la última es específica para el almacenamiento de los anuncios.

Una vez se obtienen todas las herramientas y los programas necesarios para el correcto funcionamiento de esta aplicación se procede a desplegarla.

Cuando la aplicación está ejecutándose y el usuario accede a la misma mediante su navegador se encontrará con una página web en la que debe rellenar elementos de un formulario. Obligatoria debe introducir la localidad, los filtros del precio máximo y número de habitaciones son opcionales.

Con los datos anteriormente comentados se podrían realizar distintos tipos de búsqueda (denominados “casos de uso”) y, por ende, la aplicación finalizaría su ejecución devolviendo al usuario distintos resultados en función de lo que seleccione.

Como cualquier otro trabajo o proyecto el desarrollo de este metabuscador ha pasado por diversas fases denominado *ciclo de vida*, desde el estudio del problema hasta la presentación del producto final. Todo trabajo requiere una planificación previa. Concretamente, esta aplicación sigue un ciclo de vida en cascada con retroalimentación ya que siempre se podría volver a una fase anterior para cambiar o mejorar funcionalidades ya establecidas. Se utiliza *ProjectLibre*, una herramienta de gestión de proyectos, para generar un diagrama de Gantt y poder visualizar estimaciones de la duración de cada una de las fases.

Para el desarrollo del metabuscador se necesitan herramientas tanto software como hardware y requiere de un grupo de personas (personal) que sea capaz de llevarlo a cabo. Se necesitan sólo dos personas, el tutor que desempeña la función de cliente, y el alumno que hace de jefe de proyecto, analista, diseñador y programador. Se estima que la dedicación a este trabajo es de 560 horas en cuatro meses y medio. Por ello, los costes totales del personal son de 14.035 € y los del material 117,8 €, haciendo un coste total del metabuscador de 14.152,8 €.

En la parte final del documento se realiza una pequeña reflexión sobre las conclusiones del alumno después de haber desarrollado e implementado el metabuscador y las posibles mejoras que podrían añadirse en un futuro.

Capítulo 1 INTRODUCTION / INTRODUCCIÓN

1.1 Introduction

The main objective of this chapter is to give at the reader a small general overview of this work, explaining their context, the motivation to tackle it and the marked goals. It also has a section where the structure of the rest of the document is defined.

1.1.1 Context

This work is immersed within the area of the World Wide Web (WWW or Web), the largest information network in the world and it is accessible through Internet. It will focus on the broad domain of search engines and, in particular, within the metasearch engines.

In recent years, the frequency and the reasons of search information in the web have increased. Any type of information is accessible through web applications or web services such as search engines, metasearch engines or comparators. If it is not present may be because it does not exist or several restrictions have been imposed that prevent find it.

Specifically, a metasearch engine is a web application whose system focuses on gather information based on titles or keywords of several searchers and it differs from them because it does not have database externally powered (there is not intervention by any user). It uses the information from databases of other search engines, processed it and finally is displayed it to the user. It can work in two ways: real-time searches when the user makes a query, or carry out massive searches in the background in order to be stored all information in local databases and to be available when the user accedes to the service.

As can be seen, due to the performance of a metasearch engine and the ability to show results from diverse origins on a single page have led to the increased use of these applications in the digital world.

1.1.2 Motivation

According to the inquiry about Equipment and Use of Information and Communication Technologies in Households in 2014 in Spain, there are more Internet users than computer users over the previous year. [7] About 76% of Spanish citizens use this medium to access to any type of information. This data is relevant because it makes foresee that the number of users will increase due to the advancement of technology and how to access it.

These users will become more demanding when making very specific searches and therefore they will use web services that provide them quickly and easily what they want. To this we must add that the searches are more profitable if these applications are able to show results from different places because this can generate greater confidence and security in the user, allowing himself to contrast and to decide which data to use.

One consequence of the increase of Internet users is the significant increase of search engines and metasearch engines in the web.

Currently, there are a lot of web applications focused on search and book rooms, travels or cars, but there are not a lot of search engines of usual rental apartments. This is one of the reasons why this work is aimed at trying to fill a hole in this area, where there is a marked increase in demand due to the situation in Spain in this moment.

On the other hand, it is known that the housing sector has been one of the most has suffered the economic crisis, noting a timid recovery since the second half of 2013 to so far this year. [10] According to the economic daily *Expansion*, during the first months of 2015 it has exceeded the investment with respect at year 2007, at the height of this sector. [5] Therefore an increased building constructions are expected and thus a higher demand for rental apartments.

Note that another consequence of the crisis, not least important, is the job instability and an increase in unemployment that prevents to obtain mortgages. This is the reason why the trend to look for rental apartments before purchasing them is gaining strength.

The development of this work comes from the idea of providing a service through software that allows searching for this type of residence, taking advantage of the opportunities that might emerge by the data previously discussed.

1.1.3 Goals

The main objective of this work is the development of a metasearch engine, based in Apache Tomcat with the version 8.0.24 and programmed in Java Servlets and JavaScript, which to be able to perform default queries among the following rental apartment webpages: Fotocasa and Idealista.

The secondary objective is to provide quick access to the web application and their results, with a simple, easy operation and intuitive user interface because it is destined to any type of users.

The rest of marked goals are break down below:

- The application must be able to search without failures in the default web pages.

- The application must allow at user to use certain filters such as the location, the maximum price and the number of rooms for concrete queries.
- The application must store the results in their local database for increased availability in later searches.
- The application must show at user the results and to be able to access at the detail of the apartment in their respective websites.
- Finally, it must be compatible in the web browsers with major marked share as Google Chrome and Firefox.

1.1.4 Document eststructure

This document is divided into various chapters which collect all the information concerning the development of this work. The content of each is summarized below:

- In this chapter has been performed a brief introduction to the application developed, presenting the current context of search engines and metasearch engines, the motivations and the objectives.
- The Chapter 2: “State of the art” contains an overview of the technologies related to this work.
- The Chapter 3: “Materials and methods” explains all the necessary tools to carry out this work.
- The Chapter 4: “Results” performs a balance of performance that is capable of to provide this application and shows the efforts for such action.
- In Chapter 5: “Planning and future lines” is developed a description of the stages to overcome by the application, and an estimate of the budget.
- In Chapter 6: “Conclusions and future lines” are collected the conclusions during the development of this work. Besides are described the knowledge learned and future lines of expansion and improvement.
- Finally, are listed the diverse references to books, documents and webpages used in this work.

1.2 Introducción

El objetivo principal de este capítulo es dar al lector una pequeña visión general sobre este trabajo, explicando su contexto, la motivación necesaria para abordarlo y los objetivos marcados. Además cuenta con un apartado donde se define la estructura del resto del documento.

1.2.1 Contexto

Este trabajo se encuentra inmerso dentro del área de la World Wide Web (WWW o la Web), la mayor red de información en el mundo y accesible a ella a través de Internet. Se centrará en el amplio dominio de los buscadores (“search engines”) y, en concreto, dentro de los metabuscadores (“metasearch engines”).

Durante los últimos años ha ido aumentando la frecuencia y los motivos por los que se busca información en la Web. Cualquier tipo de información está accesible gracias a aplicaciones o servicios web como los buscadores, metabuscadores o comparadores, y en caso de no estar presente es porque no existe o se han impuesto restricciones de diversa índole que impiden encontrarla.

Concretamente, un metabuscador es una aplicación web moderna cuyo sistema se enfoca en recopilar información, en función de títulos o palabras claves, de varios buscadores (motores de búsqueda o directorios) y se diferencia de ellos porque carece de base de datos alimentada externamente, es decir, no hay intervención por parte de ningún usuario de Internet. Utiliza la información presente en bases de datos de otros buscadores, la procesa y finalmente la muestra al usuario. Puede funcionar de dos maneras: realizar búsquedas en tiempo real en el momento que el usuario hace una consulta, o bien llevar a cabo búsquedas masivas en segundo plano (procesos en background) con el objetivo de almacenar en bases de datos locales toda la información, pasando a estar disponible con mayor rapidez cuando un usuario acceda a tal servicio.

Como se puede apreciar, debido al funcionamiento en sí de un metabuscador y la capacidad de mostrar resultados de diversos orígenes en una misma página han provocado el progresivo uso de este tipo de aplicaciones dentro del mundo digital.

1.2.2 Motivación

Según la encuesta sobre Equipamiento y Uso de Tecnologías de Información y Comunicación en los Hogares en el año 2014 en España, hay más usuarios de Internet que de ordenador con respecto al año anterior. [7] Alrededor del 76% de los ciudadanos españoles

usan este medio para acceder a cualquier tipo de información. Este dato resulta relevante pues hace prever que el número de usuarios aumentará debido al avance de la tecnología y a la forma de acceder a ella.

Estos usuarios cada vez serán más exigentes al momento de realizar búsquedas muy concretas y, por ende, utilizarán servicios web que les brinde con rapidez y facilidad lo que desean encontrar. A esto hay que añadir que las búsquedas son mucho más provechosas si estas aplicaciones son capaces de mostrar resultados de diversos lugares, pues ayudará a generar mayor confianza y seguridad en el usuario, permitiendo que él mismo la contraste y decida qué datos usar.

Una de las consecuencias del aumento de internautas es el notable incremento de buscadores presentes en la Web y, cómo no, el de los metabuscadores.

Actualmente, hay una gran cantidad de aplicaciones web centradas en buscar y reservar habitaciones en hoteles, viajes o coches, y dentro de las cuales no abundan buscadores de apartamentos de alquiler habitual. Es por ello que este trabajo va encaminado en intentar cubrir un hueco en esta área, donde hay un notable aumento de demanda debido a la situación que atraviesa en este momento España.

Por otro lado, se sabe que el sector inmobiliario ha sido uno de los que más ha sufrido la temida crisis económica, notándose una tímida recuperación desde la segunda mitad del año 2013 hasta lo que va del presente año. [10] Según el diario económico *Expansión*, durante los primeros meses del 2015 se ha superado la inversión inmobiliaria en España que se realizó en 2007, en pleno apogeo de este sector. [5] Se prevé mayor construcción de edificios y, por ende, mayor demanda de alquiler de viviendas.

Cabe destacar que otra consecuencia de la crisis, no menos importante, es la desencadenante inestabilidad laboral y, con ella, el aumento de desempleados incapaces de hacerse cargo de hipotecas. Por ello está cobrando fuerza la tendencia de buscar apartamentos de alquiler habitual antes que adquirirlos.

El desarrollo de este trabajo surge de la idea de prestar un servicio a través de un software que permita la búsqueda de este tipo de estancia, aprovechando las oportunidades que podrían conllevar los datos anteriormente comentados.

1.2.3 Objetivos

El objetivo principal de este trabajo es el desarrollo de un metabuscador, basado en Apache Tomcat versión 8.0.24 y programado en Java Servlets y JavaScript, que sea capaz de realizar consultas predeterminadas entre los portales de búsqueda de pisos de alquiler: *Fotocasa* e *Idealista*.

El objetivo secundario es brindar un acceso rápido a la aplicación web y a los resultados encontrados, con una interfaz de usuario sencilla, intuitiva y de fácil manejo pues está destinada a todo tipo de usuarios.

A continuación se procede a desglosar el resto de los objetivos marcados:

- La aplicación debe ser capaz de realizar la búsqueda sin fallos en los portales predeterminados.
- La aplicación debe permitir al usuario usar ciertos filtros, como la localidad, el precio máximo y el número de habitaciones para concretar las consultas.
- La aplicación debe almacenar en su base de datos local los resultados obtenidos para mayor disponibilidad en posteriores búsquedas.
- La aplicación debe mostrar al usuario los resultados y poder acceder al detalle de las viviendas en sus respectivas páginas webs.
- Por último, debe ser compatible en los navegadores web con mayor cuota de mercado como Google Chrome y Firefox.

1.2.4 Estructura del documento

El presente documento está dividido en diversos capítulos que reúnen toda la información concerniente al desarrollo de este trabajo. Acto seguido se procede a resumir el contenido de cada uno de ellos.

- En el presente Capítulo se ha realizado una pequeña introducción a la aplicación desarrollada, presentando el contexto actual de los buscadores y metabuscadores, las motivaciones y los objetivos marcados.
- El Capítulo 2: “Estado del arte” contiene una visión global de las tecnologías estrechamente relacionadas con este trabajo.
- En el Capítulo 3: “Materiales y metodología” se mencionan y relatan todas las herramientas necesarias para abordar este trabajo.
- En el Capítulo 4: “Resultados” se realiza un balance del rendimiento que es capaz de brindar esta aplicación, además de mostrar los esfuerzos realizados para tal acción. Se recogen los resultados obtenidos de su implementación.
- En el Capítulo 5: “Planificación y presupuesto” se desarrolla una descripción de las fases por las que ha pasado la aplicación, y como añadido una estimación del presupuesto.
- En el Capítulo 6: “Conclusiones y líneas futuras” se recogen las conclusiones extraídas a lo largo del desarrollo de este trabajo. Además se describen los conocimientos aprendidos y las líneas futuras de ampliación y mejora.
- Por último se enumeran las diversas referencias a libros, documentos y webs utilizadas.

Capítulo 2 ESTADO DEL ARTE

El principal objetivo de este capítulo es realizar un pequeño análisis sobre las tecnologías y herramientas que guardan cierta relación con el presente trabajo. Se busca poder facilitar al lector la comprensión de ciertos conceptos y del resto de los capítulos.

Aunque este trabajo está orientado al desarrollo de una herramienta software capaz de realizar determinadas consultas sobre apartamentos y mostrar sus resultados, es necesario hacer una pequeña introducción sobre las tecnologías en las que está implementada. Es importante abordar el estudio de las técnicas y metodologías antecedentes que han permitido el avance de dichas tecnologías, desde los orígenes de Internet hasta los actuales buscadores y metabuscadores. Por ello este capítulo se divide en tres temas principales: la Web, los buscadores y las aplicaciones web.

2.1 Origen y situación actual de la Web.

El origen de Internet se remonta hacia 1969. Surgió como un proyecto de investigación dentro de un ámbito militar. En dicho año el Departamento de Defensa Americano (DoD) consideró vulnerable a su sistema de comunicaciones pues estaba basada en la comunicación telefónica bajo una tecnología de conmutación de circuitos, es decir, con establecimientos de conexiones únicas entre los puntos finales. Como es de imaginar, en aquel entonces estos enlaces eran limitados y corrían el riesgo de quedar aisladas partes del país en caso de sufrir ataques militares.

Por ello, este Departamento a través de su Agencia de Proyectos de Investigación Avanzados (Advanced Research Projects Agency, ARPA) llevó a cabo un proyecto denominado ARPANET en la cual se desarrolló una red experimental de cuatro nodos desplegados en diciembre de 1969, con el objetivo de conseguir la correcta recepción de información en su destino aunque la red estuviera destruida entre tres universidades en California y una en Utah, Estados Unidos.

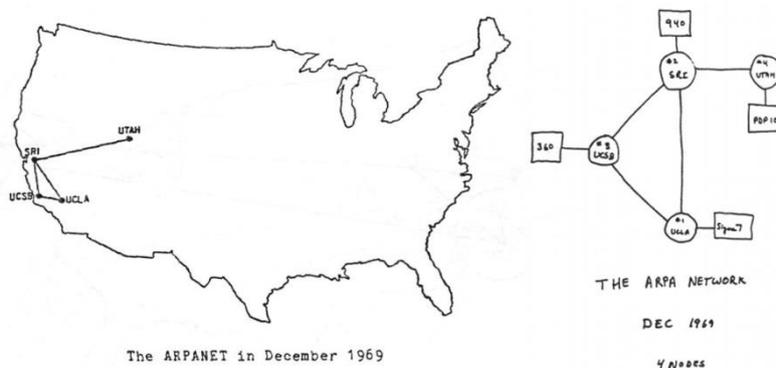


Figura 5. Los cuatro nodos iniciales de ARPANET a finales de 1969.

Fuentes: <http://learn.icann.org/mod/book/tool/print/index.php?id=1088> (modo invitado)
http://www.hipertexto.info/documentos/h_www.htm

Esta nueva red de ordenadores causó tal impacto que en 1972 se introdujo un sistema de correo electrónico y supuso un sorprendente aumento en el tráfico generado. Dos años más tarde se presentó el protocolo “Transmission Control Protocol/Internet Protocol” (TCP/IP), el cual describía los pasos a seguir para establecer una comunicación, además de proporcionar sistemas independientes de intercambio de datos entre ordenadores y redes de distinto destino. Gracias a esto aumentaron las redes de área local conectadas a ARPANET, la cual sufrió diversos cambios de nombre hasta que finalmente pasó a denominarse Internet. La Internet que conocemos en la actualidad. [30]

Alrededor de 1989 Tim Berners-Lee, ingeniero británico e investigador especializado en telecomunicaciones, propuso y puso en marcha el sistema de hipertexto llamado *Enquire* que permitía almacenar pequeñas porciones de información con la capacidad de conectarlas entre sí. Berners-Lee utilizó un NextCube como primer servidor web del mundo, llegando a elaborar en 1991 el primer programa visualizador (primer navegador web) para un servidor y cliente, la World Wide Web (WWW o Web). En dicho año apareció el lenguaje HTML (HyperText Markup Language), y junto a la WWW forman un conjunto de protocolos que permiten realizar fácilmente consultas remotas de archivos de hipertexto y utilizan Internet como medio de transmisión. [17]

Desde sus inicios hasta la actualidad la Web no ha cesado de desarrollarse y evolucionar con nuevas tecnologías y metodologías que completan y mejoran este sistema digital, hipertextual y en red. Su crecimiento ha sido exponencial con el paso de los años, pasando de 50 servidores web en 1993 a existir varios cientos de millones hoy en día (la mayoría concentrados en Estados Unidos), visibles en la siguiente figura:



Figura 6. Número de servidores web en el mundo.

Fuente: <http://www.indexmundi.com/map/?v=140&l=es>

Por otro lado, podemos ver que el desarrollador web Maurice de Kunder, autor de la página *worldwidewebsite.com*, ofrece diariamente datos estadísticos sobre la cantidad de

páginas web indexadas (rastreadas) presentes en la Web. A día 11 de septiembre de 2015 se obtiene como resultado una cantidad de al menos 4.74 billones páginas. Debido a esto es prácticamente imposible hacerse una idea sobre la dimensión de la cantidad de información accesible en la Web. La tecnología avanza y no sería extraño que este dato vaya incrementándose con el paso del tiempo.

Está claro que uno de los servicios con más éxito que se ha producido en Internet es la WWW, a tal punto que frecuentemente se usan los términos “Internet” y “Web” indistintamente. La Web es tan sólo una parte de Internet pues actualmente coexiste con otros protocolos y servicios como el envío de correo electrónico (SMTP), el acceso remoto a otros dispositivos (SSH y Telnet) y la transmisión de archivos (FTP, P2P), entre otros.

2.2 Buscadores

2.2.1 Origen y situación actual de los buscadores.

Debido al gran impacto que causó el nacimiento y desarrollo de la World Wide Web surgió la necesidad de generar un navegador (“browser”) que sea capaz de rastrear la red con el objetivo de encontrar información de interés accesible para los usuarios de Internet, pues se podía acceder a las distintas páginas web si se conocían previamente sus direcciones URL, también mediante un enlace dentro de alguna de ellas (o correo electrónico), o bien se accedían a directorios (sitios web en los que cualquier usuario puede acceder a páginas que aborden temas concretos) de las mismas.

En febrero de 1993 Marc Andreessen (fundador del navegador Netscape) genera la primera versión del navegador gráfico “Mosaic” que permitió acceder con mayor fluidez a la Web, llegando a ser el primer navegador comercial con una interfaz gráfica de fácil manejo a través del uso del ratón y la incorporación de imágenes al texto [17], visible en la siguiente ilustración



Figura 7. Ejemplo de la interfaz gráfica del navegador Mosaic.

Fuente: <http://www.notialdia.net/mosaic-el-primero-navegador-referente-de-la-world-wide-web>

Con posterioridad surgieron los buscadores. Un buscador es una aplicación web cuyo funcionamiento se basa en consultas a una base de datos que almacena datos capaces de relacionar direcciones de páginas web con su contenido, facilitando la muestra de un listado de páginas estrechamente relacionadas con el tema que el usuario ha decidido buscar. Existen diversos tipos de buscadores, en función del acceso a la base de datos y a la forma en la que están desarrollados, pero todos ellos cumplen con la funcionalidad de realizar consultas y mostrar resultados afines a un tema en concreto.

En primer lugar apareció el buscador de índices basado en direcciones URL y dio lugar finalmente al desarrollo de motores de búsqueda basados en palabras claves. Hay que destacar que antes de la llegada de la WWW, ya existían motores de búsqueda para otros protocolos o usos, como por ejemplo *Archie* para FTP y *Veronica* para el protocolo Gopher.

En la actualidad hay un número reducido de buscadores destacados que son realmente utilizados por los usuarios. Los principales son: Google, Bing (Microsoft), Yahoo! (motor de Bing), Ask (motor de Google) y Baidu (en China). Se puede ver en la siguiente figura la cuota de mercado (porcentaje de uso) de dichos buscadores a nivel mundial:

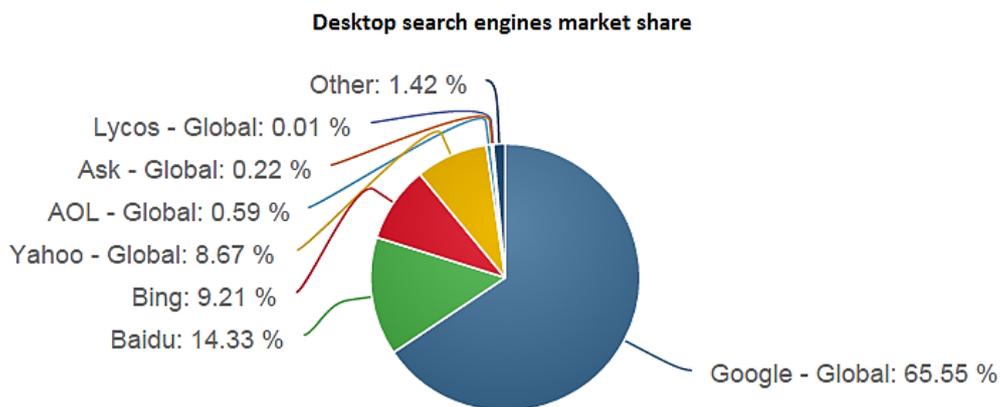


Figura 8. Cuota de mercado de los principales buscadores entre enero y agosto de 2015.

Fuentes: <https://www.netmarketshare.com/search-engine-market-share.aspx?qprid=4&qptimeframe=Y&qpmr=1000&qpcustomd=0&qpcd=1300>

Si bien es fácil apreciar que Google es el buscador más utilizado por los usuarios de Internet alrededor del mundo, aunque un estudio de hace un par de años realizado por *Greenlight*, una agencia de crecimiento digital, emitió algunas excepciones al concretar que en China lidera Baidu, en Rusia lo hace Yandex y en Japón Yahoo!. [28]

En España, según *Posicionamiento Internacional*, la cuota de mercado de este gran buscador era del 96% en 2012. [6] Dos años después se han alcanzado cifras de entorno al 98%, aumentando claramente alrededor de un 2% las búsquedas realizadas. [8] Es fácilmente apreciable que Google ha monopolizado el mercado debido a las buenas prestaciones que brinda a sus usuarios.

Por último, cabe destacar que es posible acceder a una estimación del tamaño de los índices de Google y Bing gracias a la página anteriormente comentada de Maurice de Kunder, visibles en las siguientes gráficas:

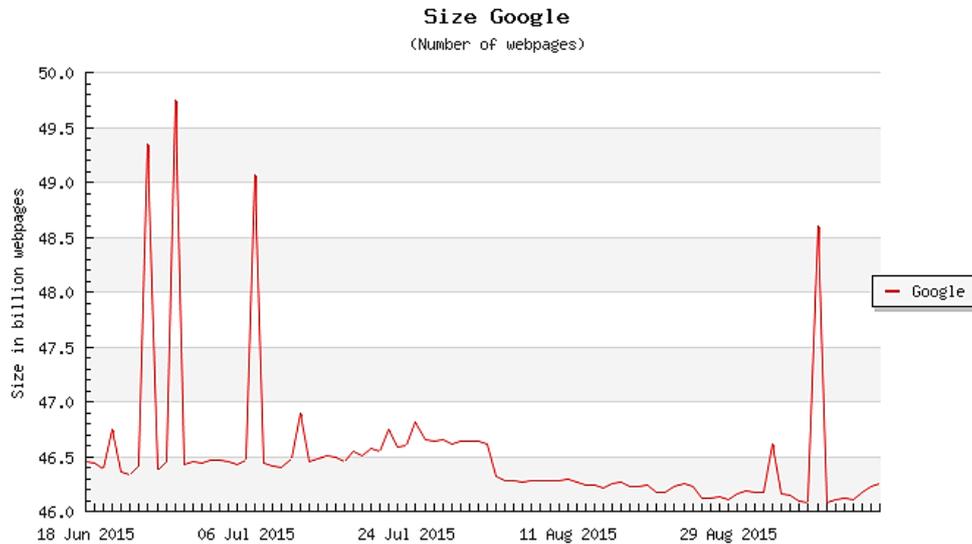


Figura 9. Tamaño de Google.

Fuente: <http://www.worldwidewebsize.com>

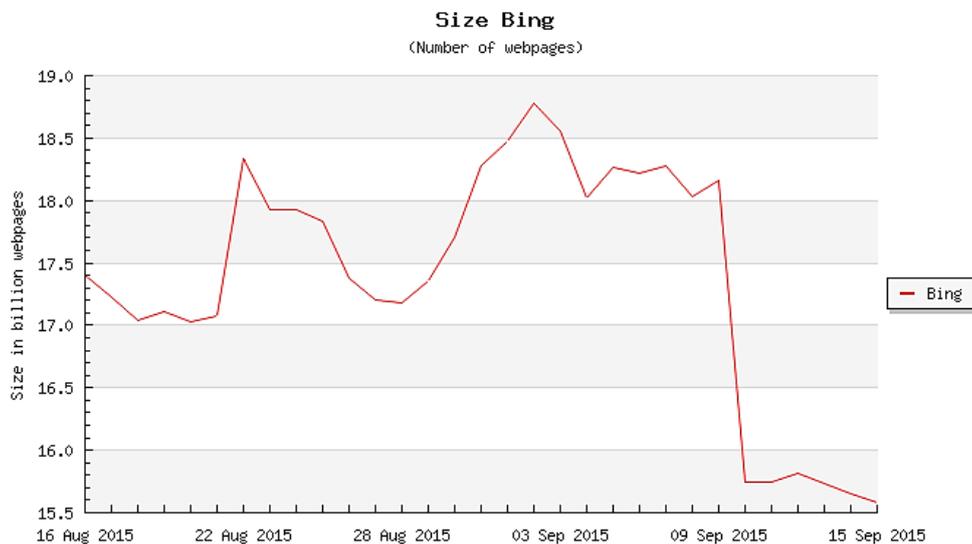


Figura 10. Tamaño de Bing.

Fuente: <http://www.worldwidewebsize.com>

Como se puede apreciar, el buscador Google presenta, con notable diferencia, un tamaño de índice mucho mayor pues se sabe que es el más potente que existe debido a que es capaz de abarcar el mayor contenido posible de la red. Parece muy difícil que otros buscadores puedan desbancarlo en un futuro.

2.2.2 Cronología de los buscadores

Internet se ha convertido en una potente herramienta para la búsqueda rápida de cualquier tipo de información en cualquier parte del mundo, con dicha finalidad han surgido los buscadores. [9] A continuación se procede a detallar la historia del desarrollo de los principales:

i. 1993 – Wandex

Se trata del primer buscador, desarrollado por Matthew Gray en el Instituto Tecnológico de Massachusetts (Massachusetts Institute of Technology, MIT) y realizado por la World Wide Web Wanderer, de ahí su nombre. Es un robot de búsqueda de índices con notables problemas de infraestructura y velocidad al alcanzar varios cientos de visitas. Pretendía medir el tamaño de la red pero se amplió para poder leer direcciones URL.

ii. 1993 – Aliweb

Martijn Koster dio a conocer un nuevo buscador en noviembre de ese año, Aliweb. Se trata de un sistema más sofisticado que Wandex con la principal novedad de ser generado como un motor de búsquedas, no un índice. Con esta aplicación ya era posible usar palabras clave en lugar de direcciones URL, además de subsanar la ralentización del motor en consultas masivas pues carece de robot de búsquedas. Destacar que hoy en día sigue en funcionamiento.

iii. 1994 – WebCrawler

Se trata del primer motor de búsqueda de texto completo, además de crear y utilizar el estándar de exclusión de robots o spiders. Esta aplicación nació como un proyecto universitario en la Universidad de Washington, Estados Unidos, y su creador es Brian Pinkerton. En 1995 Webcrawler fue vendido a AOL (America Online), en 1997 lo adquirió Excite y finalmente en 2001 lo compró InfoSpace.

iv. 1994 – Lycos

Lycos también nació como proyecto de investigación de desarrollo de un motor de búsqueda en la Universidad Carnegie Mellon, en Estados Unidos, liderado por el doctor Michael L. Mauldin. En abril de 1998, Lycos contó con la adquisición de la corporación WiseWire que destacaba por su potente software de creación de directorios generados automáticamente y en colaboración de sus usuarios. Lycos fue adquirida por Terra en el año 2000, para volver a ser vendida cuatro años después.

v. 1994 – Excite

Excite fue fundada por Ryan McIntyre, Ben Lutch, Mark Van Haren, Joe Kraus, Graham Spencer y Martin Reinfried (cinco hackers y un experto en ciencias políticas), dentro de una investigación para la Biblioteca de la Universidad de Stanford. Excite adquirió en primer lugar el motor de búsqueda Magellan y posteriormente WebCrawler. Ofrece búsquedas basadas en palabras clave o basadas en conceptos.

vi. 1994 – Yahoo!

Realmente este buscador nació a principios de 1994 en la Universidad de Stanford, pero suele entenderse que lo hizo en 1995 tras la creación del dominio yahoo.com. Sus fundadores son David Filo y Jerry Yang, Ingenieros Eléctricos. Yahoo! fue creado como directorio de páginas web pero por motivos de avance de tecnología se hizo imprescindible añadir un motor de búsqueda. Google desempeñó esa función hasta 2004, actualmente usa uno propio basado en Inktomi y Altavista.

vii. 1995 – Altavista

AltaVista nació como un proyecto de investigación en el seno del Laboratorio de Sistemas de Red de Digital Equipment Corporation (DEC), en Palo Alto (California). Sus fundadores y desarrolladores fueron Paul Flaherty, Louis Monier y Michael Burrows. Marcó un punto de inflexión en el área de los buscadores pues la superioridad que ejercía hacia el resto de los motores de búsqueda de aquella época era muy notable. Su motor de búsqueda era mucho más rápido y fue capaz de funcionar como un proceso de múltiples hilos. Yahoo! lo adquirió y dio por finalizada su etapa de vida en 2013.

viii. 1996 – Inktomi

Inktomi deriva de un buscador desarrollado por la Universidad de Berkeley, California, Estados Unidos. Sus fundadores son Eric Brewer y Paul Gauthier, estudiantes de dicha universidad. Es conocido por su potente motor de búsqueda y por la confección de directorios utilizando las páginas obtenidas por su Spider. Yahoo! lo adquirió en 2003 para ser la base de su motor de búsqueda.

ix. 1996 – HotBot

Esta aplicación nació como un servicio de la revista Wired, encargada de su diseño y operatividad. HotBot fue lanzado como una estrategia de marketing “engañosa” para atraer a nuevos enlaces de páginas web, pues dejaron sin disponibilidad a diversas páginas sin informe previo tras haberles ofrecido hospedaje gratuito. Fue uno de los primeros buscadores capaz de concretar una consulta sobre resultados de una búsqueda previa.

x. 1997 – Google

Google al igual que la mayoría de los motores de búsqueda nació como un proyecto de investigación en la Universidad de Stanford, Estados Unidos. Sus fundadores son Larry Page y Sergey Brin. En 1999 recibió alrededor de 20 millones de dólares de capitales privados, lo que le permitió colocarse entre los mejores buscadores del mercado. Google utiliza su propio sistema *PageRank* que le permite filtrar resultados irrelevantes, y gracias a los algoritmos que contiene le convirtió entre los más efectivos. En septiembre de este año se registra el dominio google.com. Un año más tarde, el 4 de septiembre de 1998 se fundó la compañía Google Inc. y fue el momento en el que se estrenó en Internet su motor de búsqueda.

xi. 1997 – Fast

El motor de búsqueda Fast Search & Transfer, conocido como FAST o AllTheWeb.com, fue desarrollado en la Universidad de Ciencia y Tecnología de Noruega (NTNU). Fue lanzado a Internet a mediados de 1998. Diez años después FAST fue adquirido por Microsoft y ahora es conocido como Microsoft Development Center Norway. Posee una de las bases de datos más grandes en la actualidad.

xii. 1997 – Ask

Ask forma parte de la compañía InterActive Corporation, fundada en 1996 por Garrett Gruener y David Warthen en Berkeley, California, Estados Unidos. Gary Chevsky implementó el programa original basado en un diseño propio. Ask.com surgió de la idea de generar un sistema capaz de responder preguntas formuladas en lenguaje natural bien mediante consultas en inglés o con palabras clave, por ello fue el primer buscador comercial del tipo pregunta-respuesta elaborado para la Web.

xiii. 1998 – DirectHit

DirectHit fue fundado por Gary Culliss y Mike Cassidy en marzo de 1998. Su funcionamiento se basa en monitorear los clics que los usuarios dan en los resultados de las consultas en otros buscadores. Es capaz de reconocer frases directamente pero no es efectivo a la hora de mostrar los resultados pues solo despliega los diez primeros hipervínculos, además de resultar imposible la realización de consultas dentro de un tema determinado. Posteriormente es adquirido por Ask Jeeves, dueño de Teoma.

xiv. 1998 – Teoma

Teoma fue implementado en 1998 en los laboratorios de la Universidad de Rutgers, New Jersey, Estados Unidos, utilizando una tecnología desarrollada por una Fundación Federal de Proyectos de Investigación. Su versión Beta se hizo pública a finales de

mayo de dicho año. Teoma utiliza un sistema parecido al de Google llamado ExpertRank, y se diferencia en que su algoritmo analiza los enlaces e hipervínculos en su contexto para calcular el ranking de importancia en un determinado tema, identificando “comunidades” en Internet. En 2001 fue adquirido por Ask Jeeves.

xv. 1998 – Open Directory Project (ODP)

Open Directory Project, conocido mayormente como DMOZ, es el directorio más grande y completo de la Web. ODP es completamente construido y mantenido por una comunidad global de editores voluntarios que evalúan los sitios web para determinar si se incluyen o no en el directorio. Nace con el espíritu de código abierto (“Open Source”) y provee de información a motores de búsqueda como Google, Lycos, AOL Search y DirectHit, entre otros.

xvi. 2009 – Bing

Bing es el buscador o herramienta web de Microsoft, con un funcionamiento y aspecto novedoso. Pasó por varios renombres como Live Search, Windows Live Search y MSN Search. El 28 de mayo de 2009 fue presentado por Steve Ballmer, director ejecutivo de Microsoft, y puesto en línea el 3 de junio del mismo año. Casi dos meses después, el 29 de julio, Microsoft y Yahoo! anunciaron que Bing reemplazaría a Yahoo Search.

El funcionamiento de Bing es llamativo pues permite a los usuarios tener una vista previa del contenido de una página en concreto, si es capaz de recopilar y obtener suficiente información de los resultados mostrados tras hacer una consulta. También tiene una opción de respuestas rápidas en la cual se despliega información relevante basada en la consulta realizada con el objetivo de ahorrar tiempo al usuario.

2.2.3 Clases de los buscadores

En la sección anterior se ha dejado entrever que existen distintas clases de buscadores según su creación y funcionamiento. En la actualidad existen diversos términos para definir sus mecanismos de rastreo, indización, recuperación y organización de información. Cada uno de ellos tiene un propósito y alcance diferente, por ello es necesario realizar una diferenciación y clasificación según su forma de trabajo [4, 12].

2.2.3.1 Índices o directorios

Este tipo de buscadores agrupan páginas web en índices conformados por temas, ideales cuando se requiere una búsqueda poco concreta con el objetivo de ver todo tipo de

información sobre un asunto en concreto. Su objetivo es encontrar sitios web que atiendan a la temática seleccionada por el usuario. Por ello organizan su base de datos por categorías para tener una guía jerárquica de directorios navegables en cada una de ellas, además que requerir soporte y mantenimiento humano.

Se trata de una tecnología no muy sofisticada pues no recorren sitios ni almacenan sus contenidos, solo registran datos como títulos y descripciones introducidas.

Algunos ejemplos de directorios son: Yahoo! y Terra.

2.2.3.2 Motores de búsqueda

Son buscadores jerárquicos que recorren páginas recopilando información sobre el contenido de las mismas. Al momento de realizar una consulta mediante frases o palabras clave, el motor de búsqueda reúne información de su base de datos vía software y la muestra al usuario según la relevancia que cada motor pueda definir. Despliegan los resultados en un listado de páginas que contengan las palabras que conforman la consulta previa.

Se diferencian de los directorios en que utilizan un robot de búsqueda que cada cierto tiempo revisa los sitios que contiene su base de datos con el objetivo de actualizarlos, por ello es frecuente visualizar información desactualizada.

Estos buscadores jerárquicos tienen un repertorio potente de programas software dividido, generalmente, en tres clases:

1. Programas que exploran la red, conocidos como arañas o spiders.
2. Los que son capaces de construir y alimentar la base de datos.
3. Los que son utilizados por el usuario desde la interfaz gráfica del buscador con el objetivo de explotar la base de datos.

Existe otra forma de evitar los algoritmos del motor de búsqueda sobre la relevancia de una página web, pues es posible realizar pagos a la empresa que controla tal buscador para que dichas páginas aparezcan entre las primeras filas.

Algunos de los ejemplos de motores de búsqueda son: Google, Bing, HotBot.

2.2.3.3 Sistemas mixtos

Como el propio nombre lo dice, son sistemas heterogéneos que integran motores de búsqueda y directorios. En la actualidad, muchos de los buscadores comúnmente utilizados

tienden a constituirse como este tipo de sistemas aprovechando los beneficios de cada funcionalidad.

Algunos ejemplos de este tipo de sistemas son Excite, Hispavista y el extinto Altavista.

2.2.3.4 Verticales o especializados

Se tratan de buscadores especializados en un mercado o área en concreto. Esta funcionalidad les permite analizar la información con mayor profundidad, disponer de mejores resultados (más actualizados) y ofrecer al usuario diversas herramientas de búsqueda avanzadas, que logran mayor exactitud que las de los otros buscadores.

Algunos ejemplos de estos buscadores son: Google académico, Trovit y Wolfram Alpha.

Además, dentro de este tipo de buscadores podemos encontrar a los llamados buscadores de portal. Engloban a los especializados que indagan en busca de información dentro de su propio portal o sitio web. Podrían considerarse también como directorios. Están basados en expresiones regulares y consultas SQL.

Algunos ejemplos de estos buscadores son: Fotocasa, Idealista y Nestoria.

2.2.3.5 Metabuscadore

Como se comentó en la introducción, los metabuscadores son aquellos sistemas que carecen de base de datos propia alimentada externamente y las consultas de los usuarios son delegadas a otros motores de búsqueda e índices previamente designados, respetando su formato original.

Su función principal es realizar búsquedas en auténticos buscadores, analizando sus resultados y presentándolos al usuario. De esta manera pretenden ahorrar tiempo de búsqueda al usuario, pues ya no necesita acudir a cada uno de los buscadores en busca de la información que desea encontrar, este sistema lo consigue en una simple consulta.

Entre las ventajas más significativas está la del aumento del rango de las búsquedas por el simple motivo de acudir a más de un motor a la vez. Permite recuperar mayor cantidad de información de distinta relevancia pues cada uno de ellos utiliza un algoritmo de ranking diferente.

Entre las desventajas, una de las principales es la relativa dificultad de poder realizar consultas que satisfagan completamente al usuario pues cada buscador tiene su propia sintaxis y codificación. Además, si se compara el tiempo que se tarda en la obtención y muestra de resultados, en los metabuscadores el tiempo es mayor debido a las diversas consultas que

realizan. Por último, destacar la complicación de fijar los criterios de ordenación de resultados ya que no se saben ni los criterios de los motores a los cuales derivan sus consultas.

2.2.4 Comparadores

Una vez hemos entrado en el campo en el cual se basa este trabajo es necesario definir a los comparadores.

Un comparador es un metabuscador que además de cumplir las funcionalidades comentadas en la sección anterior, es capaz de comparar los resultados obtenidos de las consultas realizadas, evitando que lo haga el propio usuario. Facilita en mayor medida las búsquedas con respecto a los metabuscadores.

En la actualidad este tipo de aplicaciones están también a la alza. Ya es posible comparar casi cualquier producto o servicio a través de Internet, por ello no ha hecho más que crecer el interés de los usuarios por este tipo de herramienta en los últimos años. [22]

La comparación en sí no es una funcionalidad fácil pues podría darse el caso en el que no todas las páginas en las que se delegan las consultas tengan los parámetros de comparación seleccionados por el usuario. En esta ocasión no todos los resultados desplegados por la aplicación en uso cumplirían con los filtros de comparación.

2.2.5 Casos de éxito

Esta sección está dedicada a detallar algunos de los metabuscadores más conocidos y con más éxito en la actualidad. Se puede encontrar desde sistemas que basan sus búsquedas en un solo tema, hasta algunos que cuentan con un amplio repertorio posibilitando una búsqueda más exhaustiva.

Como se ha comentado con anterioridad, este trabajo va destinado al desarrollo de un metabuscador de apartamentos de alquiler habitual pues es una tendencia que los usuarios están tomando. Por ello centraré los casos de éxito sobre metabuscadores que tengan alguna relación sobre los sectores en los cuales hay alta demanda como son el turístico, el de automoción y el inmobiliario.

A continuación se procede a detallar qué tipo de búsqueda es capaz de realizar cada metabuscador bajo estudio.

CASOS DE ÉXITO						
Comparador	Viajes	Hoteles	Seguros	Paquetes vacacionales	Hogar	Telefonía
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>		
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>		
	<input type="checkbox"/>					
			<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>				
			<input type="checkbox"/>			
			<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>			
			<input type="checkbox"/>			

Tabla 1. Algunos comparadores con éxito.

Como se puede apreciar los metabuscadores relacionados con temas de paquetes vacacionales, telefonía y hogar son los que menos abundan en el mercado actual.

Dentro de los de hogar, todos son comparadores de hipotecas (iAhorro, Kelisto) o de seguros de hogar (Rastreator). Ninguno de ellos se dedica a comparar apartamentos de alquiler habitual dentro de una localidad, ni mucho menos hay un metabuscador que se dedique a recopilar ofertas de este tipo de estancia. Por ello, este trabajo puede cobrar importancia dentro de este sector con el objetivo de cubrir este tipo de búsquedas que están en aumento como se ha comentado en la introducción.

2.3 Aplicaciones web

El objetivo principal de este trabajo es desarrollar un metabuscador que genere páginas web dinámicas en las que ofrezca resultados sobre la búsqueda de apartamentos de alquiler habitual de un municipio en concreto. Un metabuscador no deja de ser una aplicación

web con conexión a Internet alojada en un servidor concreto, por ello se hace imprescindible describir las tecnologías relacionadas con el desarrollo de este tipo de aplicaciones.

2.3.1 ¿Qué es una aplicación web?

De cara al usuario que accede a Internet para realizar alguna tarea específica, suele ser difícil percibir la diferencia entre una página web y una aplicación web. Según el diccionario de la *Real Academia Española*, una aplicación es “un programa preparado para una utilización específica”. Si llevamos este concepto a la informática se obtiene que una aplicación web es uno o varios programas capaces de realizar un cometido específico en la Web, a diferencia de una página web que es una herramienta que contiene código HTML y recursos (imágenes, código JavaScript, sonidos...) y sirve para la comunicación entre usuario y aplicación (cliente-servidor) a través del protocolo HTTP (HyperText Transfer Protocol).

Por poner un ejemplo ilustrativo, al momento de acceder al buscador Google (google.com) lo primero que se ejecuta es la aplicación web y ésta con posterioridad permite generar la página principal que todos conocemos en nuestro navegador.

2.3.2 Origen y situación actual de las aplicaciones web.

El concepto de aplicación web no es relativamente nuevo, pues uno de los primeros lenguajes de programación para el desarrollo de este tipo de aplicaciones nació antes de que Internet sea totalmente accesible a cualquier usuario. Se trata del lenguaje Perl y fue inventado por Larry Wall en 1987.

En 1995 fue cuando el desarrollo de estas aplicaciones emergió gracias a la aparición de los lenguajes Java y PHP desarrollados por Sun Microsystems y Rasmus Lerdorf, respectivamente. En la actualidad muchas de las aplicaciones de gran uso como Google, Wikipedia y Facebook están basadas en dicho lenguaje. En este mismo año Netscape Communications Corporation elaboró una nueva tecnología, JavaScript, que permitía cambiar de forma dinámica el contenido estático de las páginas web de aquel entonces, haciéndolas más atractivas de cara al usuario.

Un año después, en 1996, Sabeer Bhatia y Jack Smith desarrollaron Hotmail. Se trataba de un servicio que permitía a los usuarios acceder y consultar su correo electrónico en línea desde cualquier ordenador. Este hecho es importante pues los usuarios ya no dependerían de su ordenador personal para acceder a tal servicio. Actualmente pertenece a Microsoft y ha sido renombrado a Outlook.com desde 2012.

En este mismo año apareció la primera versión de Flash, FutureSplash Animator, cuyos fundadores fueron Jon Gay y Charlie Jackson. Fue adquirida por Macromedia e hizo su

aparición como Shockwave en 1997. Con el paso del tiempo se convirtió en una plataforma para desarrollar aplicaciones web interactivas.

En 1998 Google Inc. puso en línea su primer motor de búsqueda, Google. Desde ese entonces esta compañía sigue innovando en el desarrollo de aplicaciones web como Google Maps, Gmail, etc.

Otra de las aplicaciones que mayor impacto ha tenido en Internet es Wikipedia. Apareció en 2001 como un proyecto para complementar a Nupedia, una enciclopedia en línea. Su plataforma de desarrollo se basa en un software llamado “wiki” que permite la agregación o eliminación de información por parte de cualquier usuario. Actualmente, según el ranking de Alexa.com Wikipedia es el 7º lugar más visitado del mundo. [1]

En 2004 ocurrieron varios acontecimientos de gran importancia. El primero de ellos fue la conferencia sobre la Web 2.0 a cargo de John Battelle y Tim O’Reilly, en la cual se usó el concepto de la web como plataforma en la cual abunde la operabilidad, desarrollo y colaboración entre usuarios conformando una comunidad virtual. Otro acontecimiento es el nacimiento de Facebook, una red social. Actualmente se trata de la 2º aplicación más usada en mundo [1], llegando a revolucionar completamente el concepto de vida social.

A partir de este año el aumento de aplicaciones web accesibles en la Web ha sido notable gracias al avance de la tecnología y todo lo que ha supuesto en su desarrollo. En 2005 nació YouTube, la mayor plataforma para compartir videos. En 2006 se puso en marcha Twitter y finalmente en 2007 se dio el inicio de las aplicaciones móviles tras la aparición del iPhone. [23]

Como se puede apreciar, el impacto de este tipo de aplicaciones es muy considerable y se encuentran en pleno auge. Por este motivo, se han llegado a convertir prácticamente en imprescindibles en el día a día de las personas. Las empresas también han visto una forma de generar ganancias por medio del desarrollo y uso de aplicaciones, inmersas en su medio ideal, la Web.

2.3.3 Tecnologías

Es preciso dar a conocer las tecnologías más importantes en las que están desarrolladas las aplicaciones que se encuentran en la actualidad.

2.3.3.1 Aplicaciones web vs aplicaciones de escritorio.

Se ha comentado con anterioridad que este trabajo desarrolla un metabuscador y no deja de ser una aplicación web, pero podría realizarse como aplicación de escritorio.

A continuación se expondrán las principales ventajas e inconvenientes de cada sistema para obtener una respuesta sobre la elección que se ha tenido para el desarrollo del trabajo.

A. Aplicación web

Entre las ventajas más favorables se puede encontrar que se trata de una aplicación ligera y no es necesario instalar programas en el equipo en uso, por ello no requiere de grandes prestaciones. Puede ser ejecutada prácticamente en cualquier equipo con acceso a Internet. Su actualización y mantenimiento es sencillo. Es totalmente independiente del sistema operativo instalado en el equipo del usuario.

Entre las desventajas se puede encontrar que el equipo requiere conexión a Internet para acceder a la aplicación. Se complica el desarrollo de la misma para hacerla compatible entre diversos navegadores. El servidor que contiene los programas debe tener en cuenta la conexión simultánea entre varios usuarios. El tiempo de respuesta depende en gran medida de la conexión del usuario.

B. Aplicación de escritorio

Entre las ventajas se puede encontrar que el tiempo de respuesta y el de ejecución son menores puesto que se realiza, generalmente, de forma local sin acceder al exterior. Es más estable y robusta pues no afecta en gran medida la conexión del usuario.

Entre las desventajas se puede encontrar la dependencia de un equipo para la ejecución de la aplicación. Puede depender del sistema operativo que use el equipo (hay aplicaciones de uso exclusivo en un SO en concreto). La instalación y actualización requieren intervención humana.

Teniendo en cuenta todas las características descritas anteriormente, se ha decidido que la herramienta sea desarrollada como aplicación web pues se pueden aprovechar en mayor medida todas sus ventajas.

2.3.3.2 Lenguajes de programación

En la sección anterior se ha concluido la plataforma en la que será implementado este trabajo. Ahora es el momento de decidir en qué lenguaje estará desarrollado, por este motivo es necesario exponer los principales lenguajes de programación que actualmente son usados para la elaboración de este tipo de aplicaciones.

Como se ha comentado con anterioridad, el usuario debe comunicarse con la aplicación para que sea capaz de realizar el cometido por la cual está creada. La arquitectura de este modelo de aplicación se conoce como cliente-servidor y consta de tres niveles: el superior que interactúa con el cliente (Frontend, navegador), el intermedio que procesa datos (Backend, servidor web) y el inferior que proporciona los datos (base de datos). Cada nivel tiene una función y cometido diferente. Por otro lado, se sabe que en la actualidad hay muchos lenguajes de programación y cada uno de ellos ha sido creado para un propósito específico. Por ello es factible pensar que en cada nivel de la aplicación debería usarse un lenguaje diferente cuya funcionalidad encaje con el propósito de dicho nivel.

En el lado del cliente se debe usar HTML para dar estructura al documento. No se trata realmente de un lenguaje de programación, es un lenguaje de etiquetas y es el que leen y entienden los navegadores. Se usará CSS (Cascading Style Sheets) para dar estilos, colores y dimensiones, entre otras cosas, al documento. Por último, se usará JavaScript (que si es un lenguaje de programación) para la generación de páginas web dinámicas mejorando la interfaz de la aplicación.

En el nivel inferior, el lenguaje que predomina por excelencia para acceder y modificar los datos de las bases de datos es el SQL (Structured Query Language).

Finalmente, en el nivel intermedio es donde se concentran la mayor parte de los programas y dan vida a la aplicación. En este nivel se pueden encontrar muchos lenguajes de programación que sean extensamente usados, tales como Python, PHP y Java. TIOBE es una compañía especializada en comprobar la calidad de sistemas software. En su página web se puede encontrar el índice de uso de los principales lenguajes de programación, estando visibles en la siguiente figura:

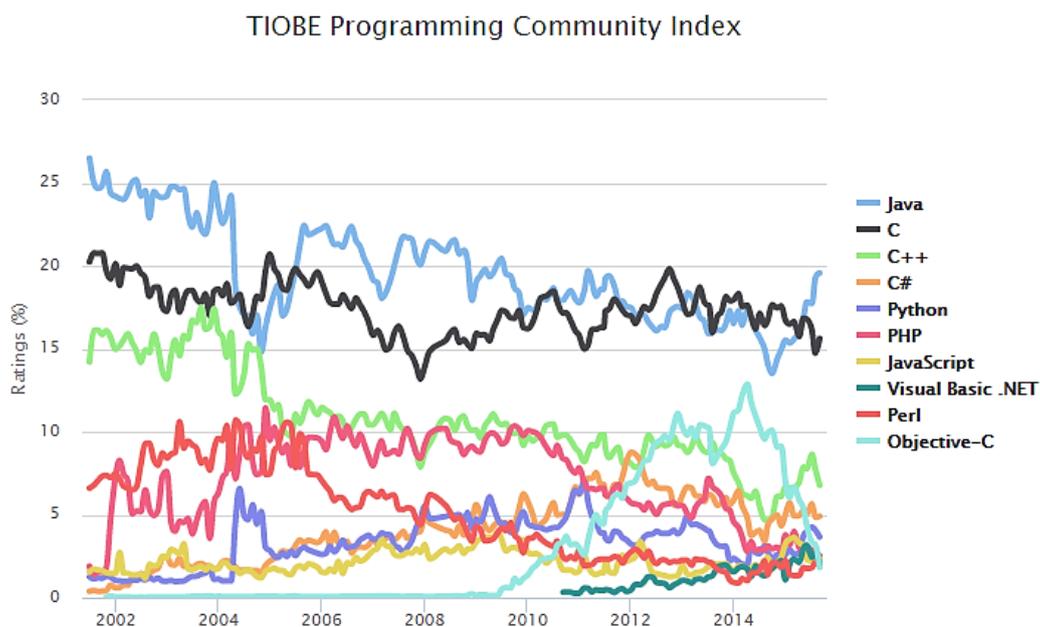


Figura 11. Clasificación de los principales lenguajes de programación. Septiembre de 2015.

Fuente: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

Se puede apreciar con claridad que Java y C destacan ante el resto de los lenguajes. Además se observa que entre los lenguajes anteriormente comentados sobre servidores web, Java sigue ganando la partida a Python y PHP. A continuación, se procede a detallar las principales características de cada uno de ellos para decidir cuál usar en este trabajo.

A. Java

Se trata de un lenguaje de programación orientado a objetos, que toma gran parte de su sintaxis de los lenguajes C y C++. Fue elaborado por Sun Microsystems en 1995 (actualmente pertenece a Oracle), con la intención de permitir a los desarrolladores de aplicaciones escriban una sola vez el programa y sea ejecutado en cualquier dispositivo (“WORA, Write Once, Run Anywhere”), no necesita ser recompilado. Desde 2012 es uno de los lenguajes más usados en aplicaciones con arquitectura cliente-servidor. Entre las ventajas se puede discernir que es un lenguaje con larga trayectoria, es multiplataforma y elimina la manipulación directa de la memoria, pues es gestionada implícitamente por un recolector de basura que se encarga de reservar, liberar y compactar espacios. La mayoría de equipos o dispositivos tienen instalado una máquina virtual en donde puede ser compilado código Java.

B. Python

También se trata de un lenguaje de programación interpretado orientado a objetos, además de soportar programación imperativa y funcional. Fue creado en los 90’s por Guido van Rossum. La sintaxis de Python es intuitiva y simple. Utiliza un tipado estático en el cual es posible cambiar el tipo de una variable después de haberla definido. Muchos desarrolladores argumentan que debido a esto es posible incurrir en continuos errores. Para ejecutar programas Python es necesario un compilador que lo convierta a código que entienda el sistema operativo a usar.

C. PHP

Por último, este lenguaje de programación fue originalmente diseñado para el desarrollo web de contenido dinámico. Nació también en 1995 a manos de Rasmus Lerdorf. Se le considera uno de los lenguajes más flexibles, potentes y de alto rendimiento (Facebook basa sus servidores en este lenguaje). Entre las ventajas se puede discernir que es multiplataforma y fácilmente incorporable de forma directa en un documento HTML.

Una vez vistos algunos detalles sobre los tres tipos de lenguajes más usados, se ha decidido utilizar Java. Todos los lenguajes poseen similitudes y aportan prácticamente las mismas funcionalidades, por lo que la diferencia en este caso la marca el sistema operativo en el que será desarrollado el metabuscador y los conocimientos previos adquiridos sobre el

lenguaje decidido. Se utilizará el sistema operativo GNU/Linux Ubuntu 14.04 LTS Trusty Tahr, por lo que la creación de ficheros con extensión .java es muy sencilla y la compilación de éstos podría hacerse con el mismo intérprete de comandos (shell, consola), demostrando a la vez que es posible generar una aplicación web de estas características sin necesidad de utilizar entornos de desarrollo como Eclipse.

2.3.3.3 Servidor Web

En la sección anterior se ha comentado el lenguaje de programación en el cual estarán basados los programas que contenga el servidor, por ello es necesario decidir el tipo de servidor web se utilizará. A continuación se mencionarán los más utilizados con el objetivo de decidir cuál de ellos se adecúa mejor al desarrollo de este trabajo.

A. Apache

Se trata de un servidor web HTTP de código abierto y disponible para plataformas Unix, Microsoft Windows y Macintosh, entre otras. Se desarrolló en 1995 y desde entonces es el más común y más utilizado en todo el mundo. Es gratuito y de código abierto. [31]

B. Microsoft IIS

Se trata de un servidor web que ofrece servicios FTP, SMTP, NNTP y HTTP/HTTPS. Es muy completo y potente, pero sólo disponible para Microsoft Windows. [13]

C. GlassFish

Es un servidor de aplicaciones de código libre desarrollado por Sun Microsystems (compañía adquirida por Oracle), que implementa tecnologías definidas en la plataforma Java EE. Puede servir como contenedor de Servlets [12]

Una vez se han visto rápidamente algunas de las características de los servidores, se descarta claramente Microsoft IIS pues no es compatible con la plataforma en la que se implementará el metabuscador. Finalmente se ha decidido usar Apache debido al uso y conocimiento previo en esta tecnología, y a que Apache posee más años de desarrollo que GlassFish. Concretamente se utilizará el servidor Apache Tomcat 8.0.24 pues servirá de contenedor para los servlets y JSPs (JavaServer Pages) que se generarán. [3] Es el servidor ideal para el correcto desarrollo del metabuscador. Sin embargo es importante resaltar que GlassFish también cumpliría con los requisitos de la aplicación.

2.3.3.4 Gestor de base de datos

Una vez seleccionado el tipo de servidor llegó la hora de decidir qué sistema gestor de base de datos (SGBD) se utilizará, pues el metabuscador precisa de una base de datos donde se almacene la información que es capaz de obtener.

Un SGBS es un software que permite la definición de bases de datos y el almacenamiento, modificación y extracción de información. Por ello proporciona herramientas para añadir, borrar, modificar y analizar los datos. Permite además el acceso de varios usuarios al mismo tiempo, y garantiza la confidencialidad, calidad, seguridad e integridad de los datos que contienen. A continuación se procede a detallar los SGBD más usados en la actualidad, con el objetivo de decidir cuál de ellos utilizar.

A. MySQL

MySQL es un SGBD relacionales rápido, sólido y flexible. Es idóneo para la creación de bases de datos con acceso desde páginas web dinámicas. Está desarrollado en C y C++, por lo que permite integrar con facilidad aplicaciones desarrolladas en dichos lenguajes. Además cuenta con conectores para varios lenguajes como Java. Entre las ventajas se puede encontrar que es de código libre. En la actualidad grandes corporaciones como YouTube, PayPal, Google y Facebook lo utilizan. [25]

B. Microsoft SQL Server

Se trata de un SGBD relacionales producido por Microsoft. Funciona como una extensión natural del sistema operativo Windows. Proporciona integridad, optimización de consultas, control de concurrencia y backup. Utiliza servicios de Windows para enviar, recibir mensajes y gestionar la seguridad de la conexión. La edición Microsoft SQL Server Express es de uso gratuito, no libre. [24]

C. PostgreSQL

Se trata de un SGBD relacionales orientadas a objetos. Es de código libre y brinda concurrencia multi-version que permite manipular grandes volúmenes de datos. Soporta parte de la sintaxis SQL y cuenta con enlaces de diversos lenguajes de programación. Es totalmente compatible con la configuración ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad). [27]

Una vez vistos los detalles más importantes de estos tres SGBD, se debe descartar a Microsoft SQL Server pues sólo funciona en Microsoft Windows. De los dos restantes, y a igualdad de condiciones, se ha decidido utilizar MySQL al uso y conocimiento previo de dicha tecnología, además de tener mayor uso en aplicaciones web.

Capítulo 3 DISEÑO E IMPLEMENTACIÓN

El principal objetivo de este capítulo es relatar los pasos seguidos en el desarrollo e implementación del metabuscador de apartamentos de alquiler habitual. Se realizará un análisis más concreto de las tecnologías utilizadas y se explicarán las decisiones más importantes que se han tomado, con la única finalidad de cumplir todos los objetivos marcados al inicio de este trabajo.

El capítulo está dividido en las siguientes secciones:

- Visión general de la solución
- Plataforma de desarrollo
- Estructuración del código
- Tecnología empleada
 - Client-side
 - Server-side: Java Servlets
 - Apache Tomcat
 - MySQL

3.1 Visión general de la solución

Antes de proceder a detallar la configuración y la arquitectura final del metabuscador, es necesario mostrar, en primer lugar, una visión general del sistema que se desea desarrollar. Este apartado servirá para comprender su funcionamiento ante la interacción entre el usuario y la aplicación web, con la finalidad de discernir en los apartados siguientes la arquitectura en la que se basa el producto de este trabajo.

El usuario es el que tiene el rol principal, pues es el que accede a la aplicación y decide realizar una consulta determinada. En este momento es cuando se realiza la primera interacción con el metabuscador y es el punto de partida para la activación de las herramientas que contiene el servidor web. Éste recibe diversos datos, realiza las comprobaciones pertinentes y los procesa. En este instante es capaz de delegar funciones a algún servlet como por ejemplo el que se encarga de descargar información de la Web y almacenarla en la base de datos, además devuelve el control a la aplicación para la posterior modificación de documentos JSP (JavaServer Pages) con el objetivo de mostrar los resultados al usuario, concluyendo de esta manera su ejecución.

Es posible ver el flujo de ejecución del metabuscador en la siguiente figura:

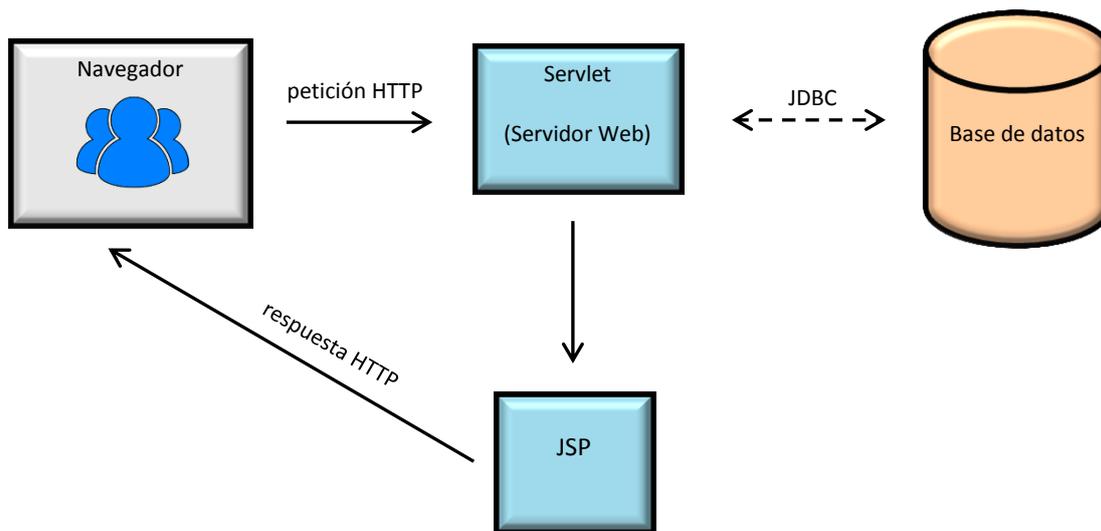


Figura 12. Modelo de funcionamiento de la aplicación web.

3.2 Plataforma de desarrollo

Ante el objetivo de desarrollar una aplicación web surge en primer lugar la necesidad de decidir qué plataforma usar para el desarrollo de este metabuscador.

Las plataformas con disponibilidad inmediata son Microsoft Windows y Linux. A igualdad de condiciones ya que es posible implementar este trabajo en cualquiera de las dos opciones se ha decidido utilizar Linux pues no necesita entornos de desarrollo, aunque los hay, para compilar y ejecutar programas escritos en Java (Java Servlets). La propia consola lo permite con previa instalación y configuración del JDK de java.

En concreto, el equipo disponible para este acometido tiene instalado Ubuntu 14.04 LTS Trusty Tahr, con soporte de 5 años.



Figura 13. Logotipo del sistema operativo Ubuntu 14.04 Trusty Tahr.

3.3 Estructuración del código

El objetivo principal de este apartado es dar una visión global de la estructura del código que permite el correcto funcionamiento de DreamHome, la aplicación web a desarrollar. Por ello, se muestra a continuación su configuración.

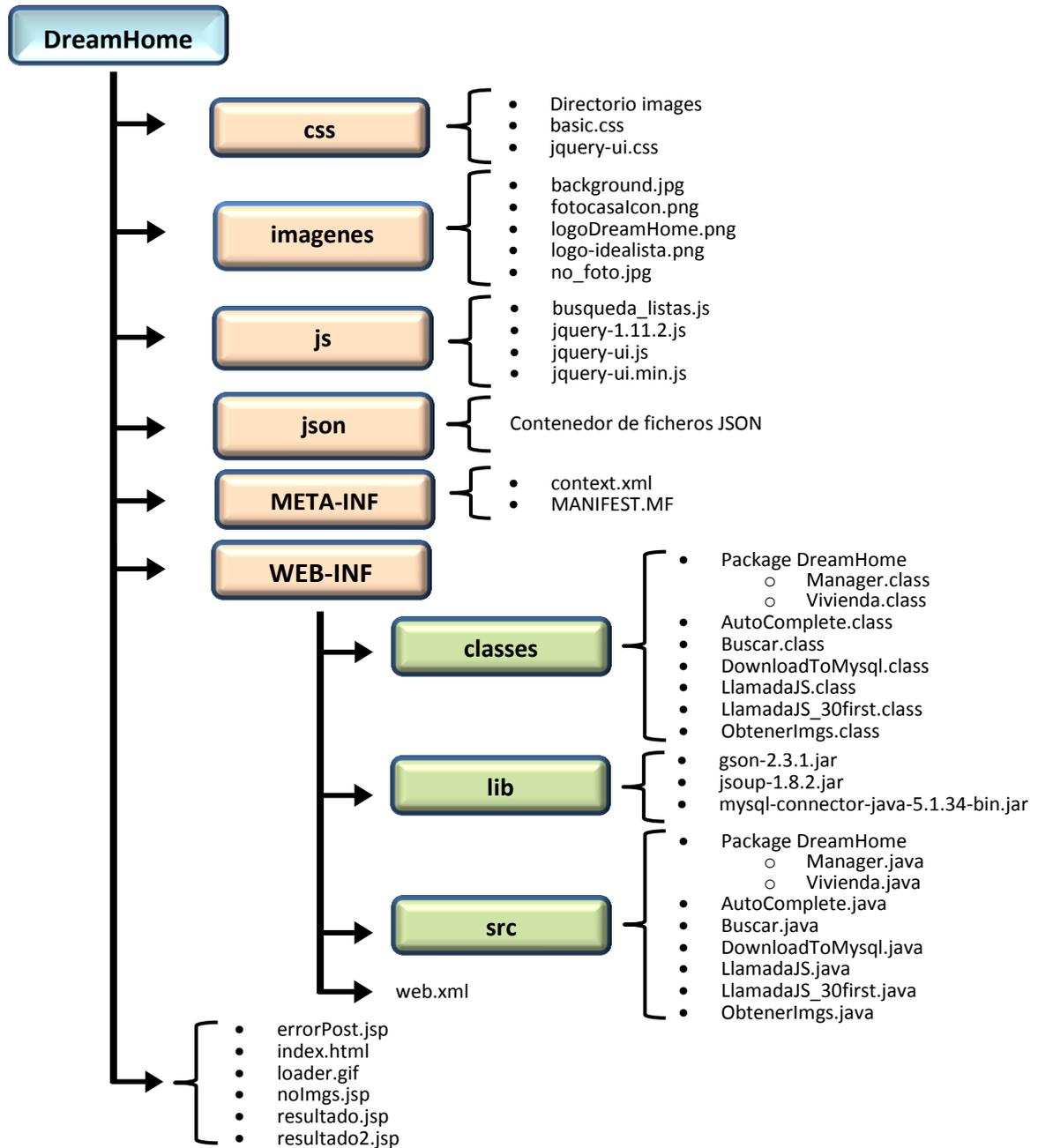


Figura 14. Estructuración del código.

Esta sería la estructura original, la cual notará una tímida modificación con el uso de la aplicación pues los directorios “imagenes” y “json” serán rellenadas con los ficheros correspondientes en función de la búsqueda que realice el usuario. Es decir, en el primero de ellos se generarán todas las imágenes de los anuncios que sea capaz de encontrar el

metabuscador, mientras que en el segundo se crearán los ficheros .json que contendrán información como la localidad, precio, número de planta, habitaciones, etc.

Los puntos a destacar son el directorio “src” pues contiene el código fuente de la aplicación, el directorio “classes” pues contiene los ficheros compilados listos para ser ejecutados y el archivo *web.xml* que describe el despliegue de la aplicación web.

3.4 Tecnología empleada

Una vez se ha visto cual es la estructura de la aplicación, es necesario explicar las herramientas y recursos empleados que han dado lugar a tal configuración.

3.4.1 Client-side

Como se ha comentado anteriormente, en la parte del cliente se precisa de un navegador (Google Chrome, Firefox) que disponga de conexión a Internet. Esta herramienta permitirá la visualización de páginas web dinámicas generadas por la aplicación. Contendrán código HTML, CSS y scripts programados en JavaScript.

En el capítulo 2 se ha mencionado el lenguaje HTML. Se trata de un lenguaje basado en marcas (etiquetas), cuya aparición fue en 1991 a manos de Tim Berners-Lee. [19] Su sintaxis es sencilla e intuitiva. Comparte el año de origen con el de la Web pues se trata del lenguaje que sirve para crear páginas web. Se puede apreciar a modo de ejemplo un pequeño fragmento del documento *index.html*:

```
<body>
  <div id="dreamHome">
    
  </div>
  <div id="introduccion">
    <p>Estimado usuario para evitar errores en la búsqueda, <b>SELECCIONE</b> alguna de nuestras sugerencias.</p>
  </div>

  <div id="form">
    <form action="Buscar" method="post">
      <div class="ui-widget">
        <div id="datos">
          <div>
            <label for="tags">Introduzca el municipio</label>
            <input id="tags" type="text" placeholder="Leganés, Móstoles..." name="munYprov" size="45px" maxlength="70"></input>
          </div>
          <div id="select">
            <label for="price">Precio máximo</label>
            <select id="price" name="price">
              <option>Sin precio máximo</option>
              <option>Hasta 350 €</option>
              <option>Hasta 500 €</option>
              <option>Hasta 650 €</option>
              <option>Hasta 800 €</option>
              <option>Hasta 1000 €</option>
              <option>Hasta 1250 €</option>
              <option>Hasta 1500 €</option>
            </select>
          </div>
        </div>
      </div>
    </form>
  </div>
```

Figura 15. Fragmento del documento *index.html*.

Otro de los lenguajes que se utilizarán es Javascript. Nació en los 90's a manos de Brendan Eich, un programador que trabajaba en Netscape. Originalmente se denominaba *LiveScript* pero Netscape firmó una alianza con Sun Microsystems desembocando en el posterior cambio de nombre a JavaScript por razones de marketing. Este lenguaje se ejecuta en el navegador del cliente evitándole tiempo de espera. Permite la generación de páginas web dinámicas. Su sintaxis es parecida a lenguajes como Java o C pero tiene más diferencias que semejanzas. [20]

Durante el desarrollo de este trabajo se ha utilizado JavaScript para desplegar animaciones como el loader.gif, manejar eventos como "clic" y "keyup", o realizar conexiones entre los distintos servlets mediante AJAX (Asynchronous JavaScript And XML). Éste es el nombre que se aplica al uso combinado de JavaScript y la API XMLHttpRequest con el objetivo de realizar peticiones HTTP en segundo plano evitando la necesidad de cargar de nuevo la página completa. Para implementar esta tecnología se ha utilizado jQuery, una librería de JavaScript, pues proporciona una API (Application Programming Interface) de alto nivel sobre XMLHttpRequest con funciones más cómodas para el programador, además de ser útil para generar y modificar parámetros. Es capaz de tratar con datos en formato XML, HTML, JSON, texto plano y Scripts. Su funcionamiento puede verse reflejado en la siguiente figura:

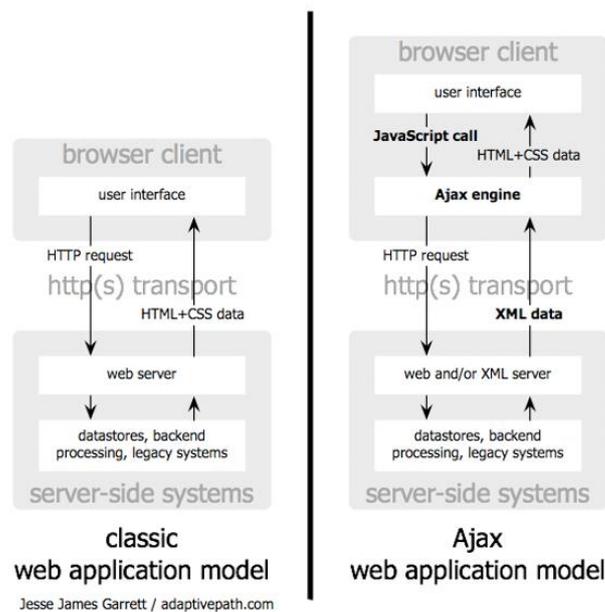


Figura 16. Modelo clásico de aplicaciones web vs. modelo Ajax de aplicaciones web.

Fuente: <http://adaptivepath.org/ideas/ajax-new-approach-web-applications/>

Además, se ha empleado jQuery UI [15] para el autocompletado de los municipios de búsqueda. Se trata de una librería jQuery que ayuda a crear interfaces de usuario interactivas, capaces de realizar efectos visuales, contener widgets y llevar a cabo interacciones entre los diversos elementos de una página web.

El script `busqueda_listas.js` es el que se ejecuta en el navegador del cliente, sus funciones más importantes se pueden apreciar en las siguientes imágenes:

```

if( arg.numFotosTot == 1 ){
    //solo hay un anuncio
    if( arg.precio == "Sin precio máximo" && arg.rooms != "No importa" ){
        h2NumFotos = '<h2 id="numFotosTot">Hay '+arg.numFotosTot+' anuncio dispon
        +arg.rooms+'.</h2>';
    }else if( arg.precio == "Sin precio máximo" && arg.rooms == "No importa" ){
        h2NumFotos = '<h2 id="numFotosTot">Hay '+arg.numFotosTot+' anuncio dispon
    }else if( arg.precio != "Sin precio máximo" && arg.rooms == "No importa" ){
        h2NumFotos = '<h2 id="numFotosTot">Hay '+arg.numFotosTot+' anuncio dispon
        +un precio máximo de '+arg.precio.split(" ")[1]+'€.</h2>';
    }else{
        h2NumFotos = '<h2 id="numFotosTot">Hay '+arg.numFotosTot+' anuncio dispon
        + con un precio máximo de '+arg.precio.split(" ")[1]+'€ y con '+arg
    }
}else{
    if( arg.precio == "Sin precio máximo" && arg.rooms != "No importa" ){
        h2NumFotos = '<h2 id="numFotosTot">Hay '+arg.numFotosTot+' anuncios disp
        +arg.rooms+'.</h2>';
    }else if( arg.precio == "Sin precio máximo" && arg.rooms == "No importa" ){
        h2NumFotos = '<h2 id="numFotosTot">Hay '+arg.numFotosTot+' anuncios disp
    }else if( arg.precio != "Sin precio máximo" && arg.rooms == "No importa" ){
        h2NumFotos = '<h2 id="numFotosTot">Hay '+arg.numFotosTot+' anuncios disp
        + un precio máximo de '+arg.precio.split(" ")[1]+'€.</h2>';
    }else{
        h2NumFotos = '<h2 id="numFotosTot">Hay '+arg.numFotosTot+' anuncios disp
        + con un precio máximo de '+arg.precio.split(" ")[1]+'€ y con '+arg
    }
}
$(h2NumFotos).insertBefore( $('div#main-content') );

//Obtengo el fichero de fotocasa, leo los datos y genero la lista
$.getJSON(fotocJSON, function(data){
    //Obtengo el fichero de idealista, leo los datos y genero la lista.
    $.getJSON(idealJSON, function(data){

```

Figura 17. Fragmento de la función `createList`.

```

$.ajax({
    url: "LlamadaJS", //genera .json con todos los anuncios
    type: "POST",
    data: {
        idealista: idealista,
        fotocasa: fotocasa,
        municipio: municipio,
        precio: datos_select.precio, //precioMaxByUser
        rooms: datos_select.rooms
    },
    success: function(data, textStatus, xhr){
        //data puede valer: 1, "NOFOTOS" o un string vacío
        if(data == 1){
            //ya hay imágenes en la base de datos
            $.ajax({
                url: "ObtenerImgs",
                type: "POST",
                data: {
                    municipio: municipio,
                    precio: datos_select.precio, //precio
                    rooms: datos_select.rooms
                },
                success: function(data, textStatus, xhr){
                    var datos = data.split("/");
                    //datos[0] == 1 -> ya había imágenes
                    //datos[1] -> num fotos total con pr
                    //datos[2] -> num páginas
                    var arg = {
                        precio: datos_select.precio, //p
                        rooms: datos_select.rooms,
                        list30first: 0,

```

Figura 18. Fragmento de la función `select`.

```

if(bool){
    mun = datos[0].split("/"); //mun1 | mun2
    var parametros = {
        roomsFotoc: roomsFotoc,
        roomsIdeal: roomsIdeal,
        precio: price,
        munic: mun,
        prov: provincia };
    var links = updateMunicipios(parametros);
    if( links != null){
        idealista = links[0];
        fotocasa = links[1];
    }else{
        idealista = null;
        fotocasa = null;
        console.log("Ha ocurrido un error al elaborar los links");
    }
    //para cualquier tipo de tratamiento con el nombre del municipio
    //siempre utilizaré el primero de ellos
    municipio = normalize(mun[0]);
}else{
    municipio = normalize(datos[0]);
    if( price == 10000 ){
        //URLs para descargar toda la información disponible del municipio seleccionado
        fotocasa = "http://www.fotocasa.es/alquiler/casas/"+municipio+"/listado?roomsFotoc";
        idealista = "http://www.idealista.com/alquiler-viviendas/"+municipio+"/"+provincia;
    }else{
        //URLs para descargar toda la información disponible en función del precio máximo
        fotocasa = "http://www.fotocasa.es/alquiler/casas/"+municipio+"/listado?maxp="+price;
        idealista = "http://www.idealista.com/alquiler-viviendas/"+municipio+"/"+provincia;
    }
}

```

Figura 19. Fragmento de la función `_setup`.

```

function autoCompleteletar(){
    $('#tags').keypress(function () {
        $.ajax({
            url: "AutoComplete",
            type: "POST",
            success: function(data, textStatus, xhr){
                $('#tags').autocomplete({
                    source: data
                });
            }
        });
    });
};
//end autoCompleteletar()

```

Figura 20. Función `autocompletar`.

Se utilizará CSS (conocido también como hojas de cascada) para dar mejor presentación de estos documentos. Nació en 1996 a manos de la World Wide Web Consortium y actualmente se puede encontrar la versión CSS3. Su sintaxis es muy sencilla pues sólo requiere de palabras clave para dar propiedades a los elementos que conforman las páginas web. Se puede apreciar a modo de ejemplo un pequeño fragmento del fichero `basic.css`:

```

/** Modificaciones sobre el logotipo */
div#dreamHome {
    text-align: center;
    padding: 20px 0px;
}
div#dreamHome2 {
    text-align: center;
    padding: 20px 0px 15px;
}
div#dreamHome2 img {
    height: 60px;
}

/** Modificaciones sobre el formulario */
.ui-widget input{
    font-family: verdana, sans;
    font-size: 14px;
}
.ui-widget select {
    border: 2px solid #000000;
    background-color: #FFFFFF;
    font-family: arial,"Times New Roman";
    font-size: 16px;
}

```

Figura 21. Fragmentos de código del archivo `basic.css`.

Por último, se utilizarán JSPs para visualizar los resultados que el metabuscador sea capaz de encontrar. No es obligatorio pero estos documentos pueden cumplir la funcionalidad de servlets. Se tratan de páginas HTML que incorpora elementos dinámicos como fragmentos de código que son evaluados en el servidor. En concreto, durante el desarrollo de este trabajo se ha incluido código en Java para la recepción y envío de parámetros compartidos entre los distintos servlets. Un ejemplo de ello se puede ver en la siguiente figura:

```

<body>
  <div id="dreamHome2">
    <a href="index.html"></a>
  </div>

  <% String munYprov = (String) request.getAttribute("munYprov");
     String numRedirect = (String) request.getAttribute("numRedirect");
     String no30first = (String) request.getAttribute("no30first");
     String precio = (String) request.getAttribute("precio");
     String rooms = (String) request.getAttribute("rooms"); %>

  <div id="introduccion">
    <p>Estimado usuario, estas son las viviendas que hemos encontrado en
      <b class="result"><%= munYprov %></b> (pág. <b id="redirect"><%= numRedirect %></b>).</p>
  </div>
  <div id="loading">
    
  </div>
  <div id="main-content">
    <div id="fotocasa"></div>
    <div id="idealista"></div>
  </div>
  <div id="back"></div>
  <div id="no30first"><input type="hidden" value="<%= no30first %>"></input></div>
  <div id="precio"><input type="hidden" value="<%= precio %>"></input></div>
  <div id="rooms"><input type="hidden" value="<%= rooms %>"></input></div>
  <div id="pagination"></div>

```

Figura 22. Fragmento de código del archivo resultado.jsp.

3.4.2 Server-side

Una vez se ha visto la tecnología empleada en el cliente, se procede a detallar la empleada en el lado del servidor.

Como se ha comentado con anterioridad, se ha decidido utilizar *Java Servlets* pues cumplen totalmente con los requisitos sobre las peticiones y respuestas HTTP usadas en el metabuscador. Un servlet es un programa escrito en Java que se ejecuta en un servidor (contenedor de servlets) y extiende su funcionalidad. Sus características más importantes a destacar son que cada petición recibida se procesa en un hilo diferente e invoca a un método específico del mismo (`doPost()`, `doGet()`...), se ejecutan en una máquina virtual de Java (JVM), y por último la capacidad de extender cualquier tipo de servidor como FTP, por ejemplo, aunque lo más común es HTTP.

Existen varias APIs de Servlets que contienen clases e interfaces y permiten la interacción con el servidor. La versión más actual es la API Servlet 3.1. En dichas APIs se puede encontrar el paquete genérico: *javax.sevLet* válido para cualquier protocolo (HTTP incluido).

La aplicación a desarrollar usará un servidor HTTP, por lo que se usará la interfaz *javax.sevlet.HttpServlet* exclusivo para este protocolo. De dicha interfaz se usarán las clases *ServletContext*, *HttpServletRequest* y *HttpServletResponse*.

Cada servlet que contiene la aplicación lleva a cabo una serie de tratamiento de datos con un fin concreto. A continuación se procede a comentar el funcionamiento principal de cada uno de ellos, acompañados de diversos fragmentos de código.

i. **Autocomplete.java**

Es el servlet que implementa la función de autocompletado de los municipios según la localidad que el usuario desee. Su función principal es obtener de la base de datos toda la lista de municipios con sus respectivas provincias y enviarlas con formato JSON a *index.html*.

```
public class AutoComplete extends HttpServlet {

    public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

        ArrayList<String> lista = new ArrayList<String>();
        Manager m = null;

        try{
            m = new Manager();
            lista = m.lista_names();

            if(lista != null){
                //utilizo la libreria GSON de Google
                JSONArray json = new JSONArray();
                for(int i=0; i<(int)lista.size(); i++){
                    //no se puede añadir a un JSONArray un String, sólo un JsonEleme
                    //caso conviene usar un JsonPrimitive que representa un valor en
                    JsonPrimitive jp = new JsonPrimitive(lista.get(i));
                    json.add(jp);
                }
                //se envía la lista a index.html
                response.setContentType("application/json");
                response.getWriter().print(json);
            }
        }
    }
}
```

Figura 23. Fragmento de código del archivo *AutoComplete.java*.

ii. **Buscar.java**

Servlet encargado de comprobar todos los datos introducidos por el usuario en el formulario principal. Tras esta comprobación redirecciona al JSP correspondiente (*errorPost.jsp*, *resultado.jsp* o *resultado2.jsp*). Se utiliza el redireccionamiento interno “forward” en caso de ser correctos los datos, o por el contrario “sendRedirect” que internamente envía un código de respuesta 3XX al cliente.

```

public class Buscar extends HttpServlet {

    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        Manager m = null;

        try{
            //Obtengo los datos enviados mediante ajax
            request.setCharacterEncoding("UTF-8"); //todos los caracteres que recib
            String munYprov = request.getParameter("munYprov");
            String precio = request.getParameter("price");
            String rooms = request.getParameter("rooms");

            //Compruebo el precio y el número de habitaciones siempre para evitar q
            if( ( precio.compareTo("Sin precio máximo") == 0 || precio.compareTo("H
                precio.compareTo("Hasta 500 €") == 0 || precio.compareTo("Hasta 6
                precio.compareTo("Hasta 800 €") == 0 || precio.compareTo("Hasta 1
                precio.compareTo("Hasta 1250 €") == 0 || precio.compareTo("Hasta
                &&
                ( rooms.compareTo("No importa") == 0 || rooms.compareTo("1 habitaci
                rooms.compareTo("2 habitaciones") == 0 || rooms.compareTo("3 habi
                rooms.compareTo("4+ habitaciones") == 0 ) ){

            //Separo [municipio, provincia] para comprobar si la selección del
            String[] tmp = munYprov.split(", ");

```

Figura 24. Fragmento de código del archivo Buscar.java.

iii. DownloadToMysql.java

Servlet cuya función principal es descargar las imágenes de los anuncios, además de generar objetos “Vivienda” definidos en el paquete (package) DreamHome e insertar la información que contienen en la base de datos local.

```

public class DownloadToMysql extends HttpServlet {

    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        //Obtengo los datos enviados mediante ajax
        request.setCharacterEncoding("UTF-8"); //todos los caracteres que recibo los
        String municipio = request.getParameter("municipio");
        String precio_max_user = request.getParameter("precio");
        String numHab_user = request.getParameter("rooms");
        String no_foto = "/home/jaime/Documentos/TFG/apache-tomcat-8.0.24/webapps/Dr
        String ruta = null;

        try{
            if( municipio != null && !municipio.isEmpty() && !precio_max_user.isEmpt:

                ServletContext context = getServletContext();
                //Leo las listas y posteriormente las elimino del ServletContext par:

                ArrayList<String> srcs_fotoc = (ArrayList<String>)context.getAttribu
                context.removeAttribute("srcs_fotocasa_"+municipio);

                ArrayList<String> srcs_ideal = (ArrayList<String>)context.getAttribu
                context.removeAttribute("srcs_idealista_"+municipio);

                ArrayList<Vivienda> viviendas = new ArrayList<Vivienda>();
                String no_foto_fotoc = "http://s.ftcs.es/Image/Placeholders/NoPhoto/";
                //Si hay problema con alguna img en fotocasa su atributo "src" puede
                String startsWith = "data:image";

```

Figura 25. Fragmento de código del archivo DownloadToMysql.java.

mientras la descarga completa de datos se realiza en segundo plano, completamente transparente al usuario.

```
//IDEALISTA
doc = testURL(idealista); //obtengo todo el documento HTML de la URL idealista
if( doc != null ){
    //Compruebo que hay información descargada (URL correcta)
    okIdeal = true;
    photos = doc.select("article[data-adid]"); //cantidad de fotos primera página
    if( (int)photos.size() > 0){ //si hay anuncios...

        fotosTotIdeal = Integer.parseInt(doc.select("div#h1-container > span").text().trim());
        String hostname = "http://www.idealista.com";
        String data, numHab, planta = null;
        int count = 0;

        for(Element foto: photos){
            //selecciono información
            if( !foto.select("div > img.vertical").isEmpty() ){
                src = foto.select("div > img.vertical").attr("data-ondemand-img");
            }else{
                src = "imagenes/no_foto.jpg";
            }
            data = foto.select("div.item-info-container > a.item-link").attr("href");
            data_url = hostname+data;
            numHab = foto.select("span.item-detail").first().text();
            metrosC = foto.select("span.item-detail").eq(1).text();
            planta = foto.select("span.item-detail").eq(2).text();
        }
    }
}
```

Figura 27. Fragmento de código del archivo LlamadaJS_30first.java.

En esta clase también se ha utilizado la herramienta Jsoup para la descarga de información.

vi. ObtenerImgs.java

Servlet encargado de generar las imágenes de los anuncios encontrados en el directorio “imagenes” del servidor Tomcat, pasando a estar disponibles en posteriores búsquedas. De esta manera se evita volver a descargar información.

```
public class ObtenerImgs extends HttpServlet {

    public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException{

        //Obtengo los datos enviados mediante ajax
        request.setCharacterEncoding("UTF-8"); //todos los caracteres que recibo los c
        String municipio = request.getParameter("municipio");
        String precio_max_user = request.getParameter("precio");
        String numHab_user = request.getParameter("rooms");
        ArrayList<Boolean> exitos = new ArrayList<Boolean>();
        Manager m = null;
        int cuenta = 0;
        int[] existe = new int[3];

        try{
            if( municipio != null && !municipio.isEmpty() && !precio_max_user.isEmpty()

                //genero los prefijos del precio y las habitaciones para las rutas
                //en las cuales se generarán las imágenes
                String endPrice = null;
                if(precio_max_user.compareTo("Sin precio máximo") == 0){
                    endPrice = "noMaxp";
                }else{
                    endPrice = "maxp"+precio_max_user.split(" ")[1]; //maxp350, maxp50
                }
            }
        }
    }
}
```

Figura 28. Fragmento de código del archivo ObtenerImgs.java.

3.4.3 Apache Tomcat

Apache Tomcat es uno de los servidores HTTP de código abierto (licencia Apache 2.0) más utilizados con soporte para Java Servlets, JavaServer Pages y tecnología WebSocket. Desarrollado por Apache Software Foundation bajo un entorno abierto y participativo en el cual colaboran grandes desarrolladores de este sector. Es un proyecto libre programado en Java, por lo que le hace perfectamente compatible con la aplicación implementada. Incluye el compilador *Jasper 2 JSP Engine* que compila JSPs y los convierte en verdaderos servlets. A partir de la versión 4.0 el contenedor de servlets es llamado "Catalina". Para este trabajo se utilizará la versión Apache Tomcat 8.0.24 de julio de 2015 sobre Java 7, que implementa Java Servlet 3.1, JSP 2.3 y WebSocket 1.1. [4]

A continuación se procede a detallar la estructura de directorios que compone este servidor, el cual tiene una jerarquía predefinida:

- **bin** -> contiene principalmente los scripts encargados del arranque y cierre.
- **conf** -> almacena diversos archivos XML, tales como `server.xml` (configuración principal de Tomcat) y `web.xml` (configuración de valores por defecto).
- **lib** -> contiene las librerías (JAR) de Tomcat. Las que vayan a ser utilizadas por alguna aplicación deben estar incluidas en el classpath.
- **logs** -> almacena los diversos logs de Catalina y de las aplicaciones.
- **temp** -> contiene ficheros temporales.
- **webapps** -> almacena las aplicaciones web. Contiene el directorio principal `DreamHome`, la aplicación desarrollada.
- **work** -> contiene temporalmente directorios y ficheros.



Figura 29. Apache Tomcat.

3.4.4 Base de datos

En esta sección se pretende dar a conocer en mejor medida el SGBD utilizado en este trabajo. También se explicará el diseño de la base de datos local.

3.4.4.1 MySQL

En el capítulo anterior se ha tomado la decisión de utilizar MySQL como sistema gestor de base de datos. Es un software de código libre, potente, rápido y flexible. Su uso en el sector de aplicaciones web con bases de datos relacionales es muy notable con respecto a sus principales competidores.

El origen de MySQL (My Structured Query Language) se remonta a la década de los ochenta, cuando Michael Widenius “Monty” empezó a desarrollar un sistema de almacenamiento de archivos que le resultara satisfactorio. En 1995, en colaboración con David Axmark, implementaron un producto que usaba el lenguaje SQL desarrollado por programadores de IBM. En el mismo año fundaron la empresa MySQL AB y actualmente es la que da soporte a este SGBD.

Entre las ventajas más importantes de este software se puede encontrar que permite acceder a las bases de datos con diferentes lenguajes de programación gracias al desarrollo de *drivers* conectores, como es el caso de Python, C++, C, PHP y Java. En concreto, para este trabajo se ha necesitado el conector JDBC que permite el acceso a bases de datos relacionales mediante SQL desde programas Java. Se ha hecho uso de los paquetes *javax.sql* y *java.sql* que requieren de dicho driver denominado *MySQL Connector/J*.

La aplicación en uso contiene un único programa Java que realiza todas las conexiones, consultas y desconexiones pertinentes con la base de datos local. Se trata del archivo *Manager.java* y está incluido en el paquete llamado “DreamHome”. Los servlets que desean interactuar con la base de datos local deben incluirlo. Las principales clases e interfaces usadas son *java.sql.Connection*, *java.sql.Context*, *java.sql.DriverManager*, *java.sql.ResultSet*, *java.sql.PreparedStatement* y *javax.sql.DataSource*.

Es importante destacar la última de estas interfaces pues es la que permite el *pool de conexiones*, posibilitando un manejo más eficiente y una gestión más sencilla de las conexiones con la base de datos local. [14] Para disfrutar de tal funcionalidad es preciso realizar ciertas configuraciones en los documentos *web.xml*, *context.xml* y *Manager.java*, que de detallan a continuación.

A. web.xml

En el documento WEB-INF/web.xml se ha añadido el siguiente código:

```
<resource-ref>
  <description>
    Resource reference to a factory for java.sql.Connection
    instances that may be used for talking to a particular
    database that is configured in the Context
    configurartion for the web application.
  </description>
  <res-ref-name>
    jdbc/buscador
  </res-ref-name>
  <res-type>
    javax.sql.DataSource
  </res-type>
  <res-auth>
    Container
  </res-auth>
</resource-ref>
```

Figura 30. Configuración del pool de conexiones en el archivo web.xml.

B. context.xml

En el documento META-INF/context.xml se ha configurado lo siguiente:

```
<Context>
  <Resource name="jdbc/buscador"
    auth="Container"
    type="javax.sql.DataSource"
    username="USERNAME"
    password="PASSWORD"
    driverClassName="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost/buscador"
    maxActive="8"
    maxIdle="4"/>
</Context>
```

Figura 31. Configuración del pool de conexiones en el archivo context.xml.

C. Manager.java

Finalmente en este documento se ha agregado lo siguiente:

```
private void connect() throws SQLException, NamingException {
    Context initCtx = new InitialContext();
    Context envCtx = (Context) initCtx.lookup("java:comp/env");
    DataSource ds = (DataSource) envCtx.lookup("jdbc/buscador");
    connection = ds.getConnection();
} //end connect
```

Figura 32. Configuración del pool de conexiones en el archivo Manager.java.

De esta manera, queda configurado correctamente el *pool de conexiones* que permite realizar más de una consulta sin necesidad de cerrar la conexión después de cada una de ellas.

3.4.4.2 Diseño de la base de datos

El objetivo de esta sección es detallar el diseño que se ha llevado a cabo de la base de datos y todas las tablas que contiene.

La base de datos local se ha denominado “buscador”, accesible al administrador a través de un usuario y una contraseña previamente configurados. Se le ha otorgado todos los permisos para poder generar, modificar y eliminar tablas.

En la siguiente figura se puede ver el contenido de la base de datos:

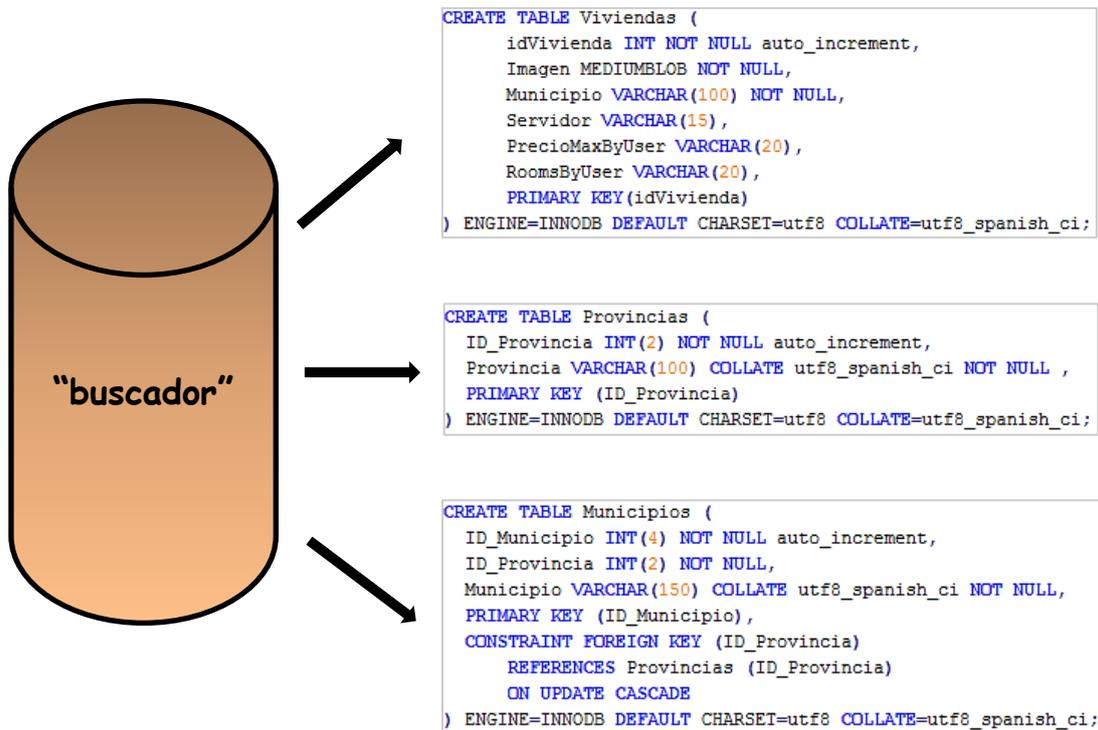


Figura 33. Tablas contenidas en la base de datos relacional.

Como se puede apreciar, la aplicación web cuenta con tres tablas y cada una de ellas está definida para almacenar tipos de datos concretos. La tabla “Viviendas” contendrá la información necesaria de los anuncios que el metabuscador pueda descargarse, las tablas “Provincias” y “Municipios”, como sus propios nombres lo dicen, contienen todas las provincias y los municipios que se pueden encontrar en España. Se ha utilizado CHARSET=utf8 para definir la manera en la que se guardan los datos y COLLATE=utf8_spanish_ci para determinar la forma de tratar (comparar, ordenar...) los datos que contienen las tablas.

3.4.5 Jsoup

El objetivo de este apartado es detallar la herramienta utilizada para la obtención de datos tras la descarga de información, técnica conocida como *Web Scraping*. La tecnología empleada para tal efecto es *Jsoup*, un parser de HTML. [16]

Jsoup es una librería de Java que provee la extracción y manipulación de datos de una página web en concreto (también de ficheros y strings), mediante el uso de DOM (Document Object Model), selectores CSS y métodos similares a jQuery.

En concreto parsea el código HTML a DOM tal como lo hacen los navegadores modernos, con el objetivo de poder manejar la información descargada y disponible. Entre las ventajas hay que destacar que esta herramienta está diseñada para hacer frente a cualquier versión de HTML (2.0, 3.2, 4.01, 5.0, etc), además de ser código libre y estar a disposición de cualquier usuario bajo licencia MIT. Para este trabajo se ha utilizado la versión 1.8.2 lanzada en abril de 2015.

Capítulo 4 RESULTADOS

Este capítulo tiene como objetivo mostrar los distintos resultados que un usuario final puede obtener al utilizar DreamHome haciendo uso del navegador Google Chrome o Firefox. Por ello se expondrán los diversos casos de uso del metabuscador.

4.1 Página principal

Antes de proceder a tal exposición es necesario mostrar la interfaz gráfica o página principal de la aplicación web desarrollada para observar los distintos tipos de búsqueda que el usuario puede seleccionar. Nada más acceder al metabuscador se obtiene lo siguiente:

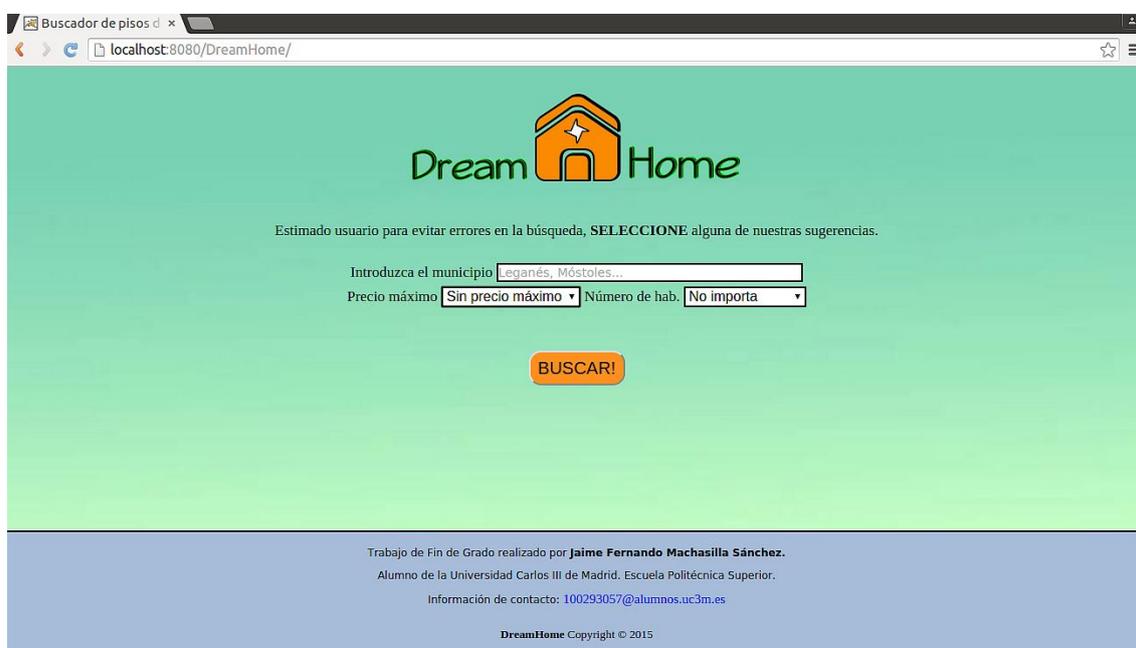


Figura 34. Página de bienvenida de la aplicación web DreamHome.

Como se puede apreciar, esta página contiene en primer lugar el logotipo del metabuscador denominado DreamHome. Acto seguido se encuentra el formulario principal, donde es imprescindible introducir la *localidad* (municipio) en la que se desea encontrar apartamentos de alquiler habitual para la consecución de una búsqueda correcta.

Los campos de *precio máximo* y el *número de habitaciones* son opcionales, y cuentan con las siguientes alternativas según la preferencia del usuario:

<div style="border: 1px solid black; padding: 2px;">Sin precio máximo ▾</div> <div style="background-color: #007bff; color: white; padding: 2px;">Sin precio máximo</div> <p>Hasta 350 €</p> <p>Hasta 500 €</p> <p>Hasta 650 €</p> <p>Hasta 800 €</p> <p>Hasta 1000 €</p> <p>Hasta 1250 €</p> <p>Hasta 1500 €</p>	<div style="border: 1px solid black; padding: 2px;">No importa ▾</div> <div style="background-color: #007bff; color: white; padding: 2px;">No importa</div> <p>1 habitación</p> <p>2 habitaciones</p> <p>3 habitaciones</p> <p>4+ habitaciones</p>
---	--

Figura 35. Alternativas en los campos precio máximo y número de habitaciones.

4.2 Casos de uso

En este apartado se pretende mostrar al lector los distintos casos de uso de la aplicación implementada en función de lo que el usuario seleccione o desee buscar.

4.2.1 Introducción correcta del municipio

El usuario introduce de forma correcta únicamente el municipio. Se obtiene una búsqueda de apartamentos satisfactoria en caso de existir anuncios en dicha localidad. Es posible ver los detalles en los portales en los que se encuentran publicados (Fotocasa o Idealista) dando un simple *click* en las imágenes.

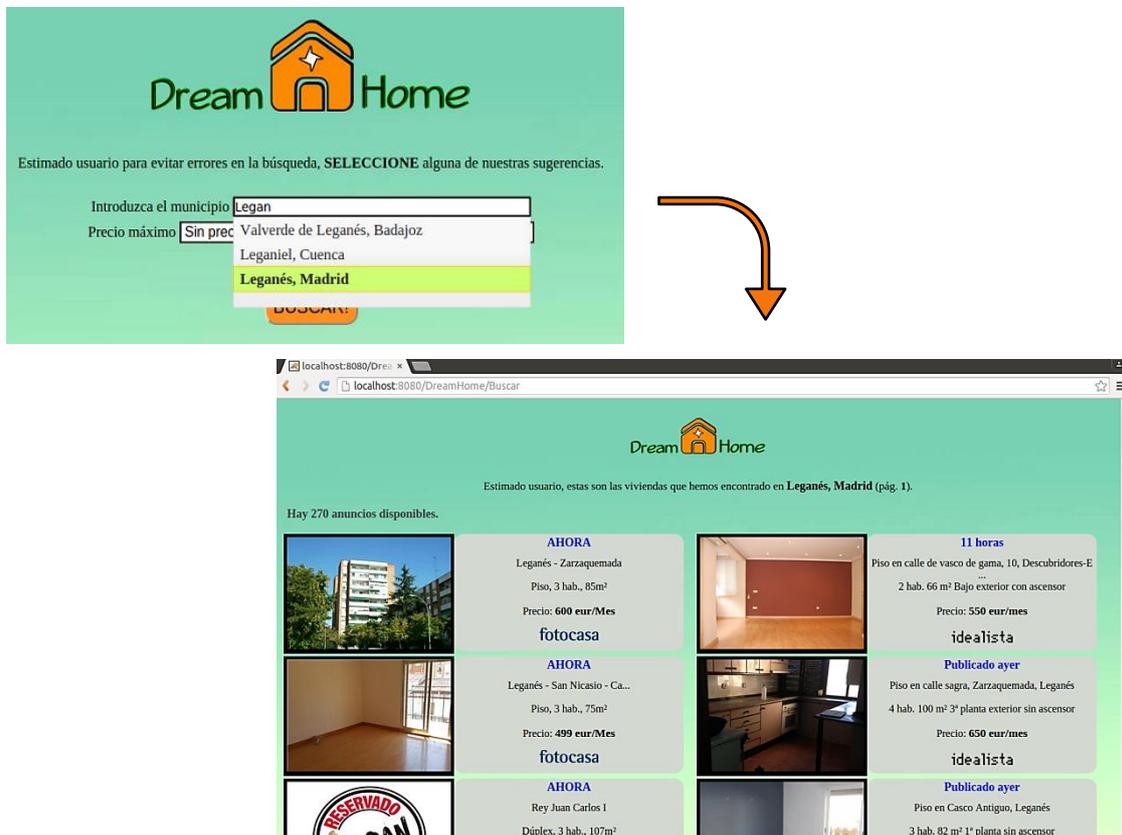


Figura 36. Introducción del municipio. Muestra de resultados.

4.2.2 Introducción correcta del municipio y precio máximo

En este caso el usuario decide introducir correctamente el municipio y selecciona alguna alternativa del precio máximo. La búsqueda de apartamentos finaliza satisfactoriamente en caso de existir anuncios que cumplan los parámetros establecidos. También es posible acceder a los detalles de cada uno de ellos en los portales en los que se encuentran publicados mediante un *click* en las imágenes.

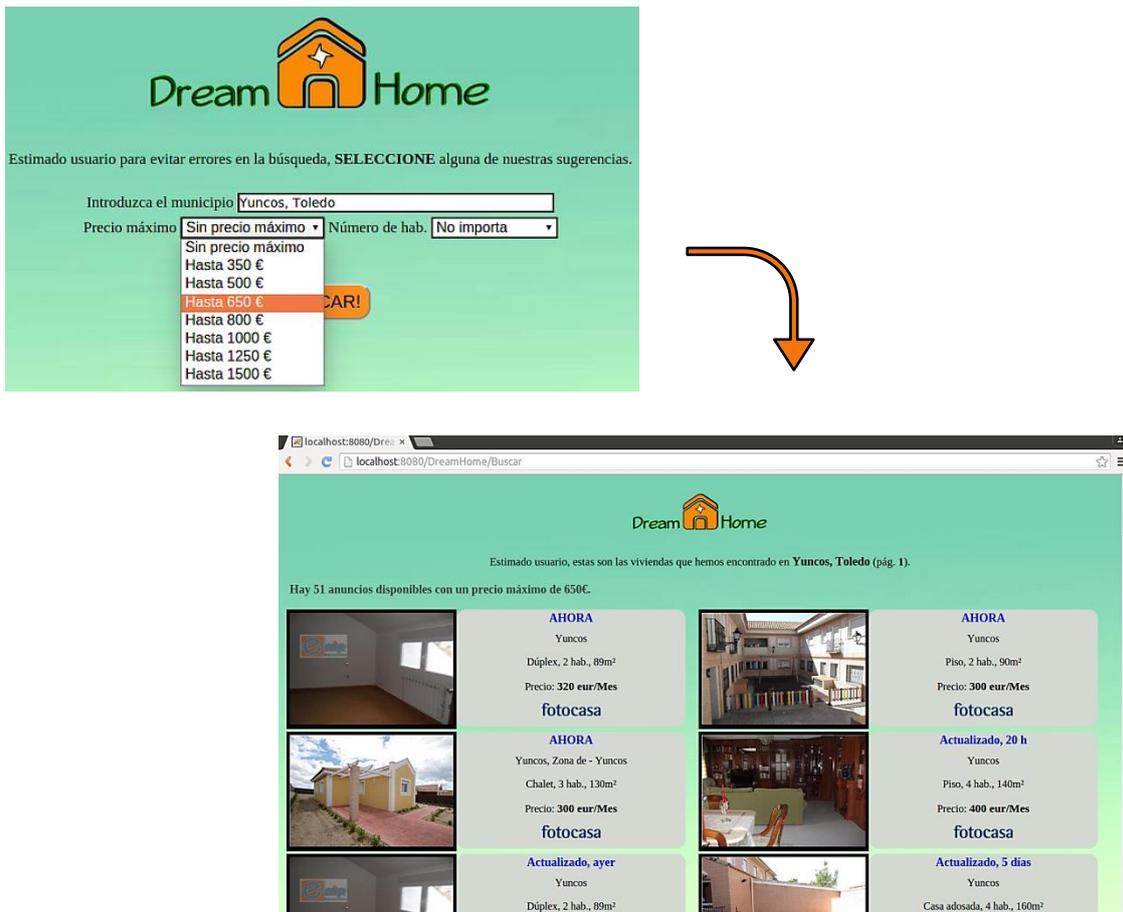


Figura 37. Introducción del municipio y precio máximo. Muestra de resultados.

4.2.3 Introducción correcta del municipio, precio máximo y número de habitaciones

El usuario introduce correctamente el municipio y selecciona alternativas del precio máximo y número de habitaciones. La búsqueda de apartamentos finaliza satisfactoriamente en caso de existir anuncios que cumplan los parámetros establecidos. También es posible acceder a los detalles de cada uno de ellos en los portales en los que se encuentran publicados mediante un *click* en las imágenes.

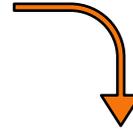


Figura 38. Introducción del municipio, precio máximo y nº de habitaciones. Muestra de resultados.

4.2.4 Introducción correcta de todos los campos sin resultados

En este caso el usuario introducen correctamente todos los campos (o únicamente la localidad), pero el metabuscador no es capaz de mostrar resultados pues las páginas fuente donde se delegan las búsquedas (Fotocasa e idealista) no contienen anuncios con dichos parámetros. La búsqueda finaliza correctamente sin mostrar ningún anuncio.

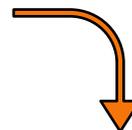




Figura 39. Introducción correcta de datos. No se encuentran resultados.

4.2.5 Introducción errónea de datos

En este apartado hay que valorar dos opciones.

- A. Aunque se seleccione alguna alternativa del precio máximo o del número de habitaciones, si el usuario introduce de forma errónea el municipio la búsqueda no se efectúa.
- B. El usuario no introduce ningún dato en el formulario. La búsqueda tampoco se efectúa.



Figura 40. Introducción errónea de datos (opción A parte superior izquierda, opción B parte superior derecha).

Capítulo 5 PLANIFICACIÓN Y PRESUPUESTO

En este capítulo se llevará a cabo la explicación del ciclo de vida de este trabajo, se identificarán las fases del desarrollo y las actividades contempladas en cada una de ellas. Todas las actividades realizadas tienen asociados determinados costes, por lo que al final del capítulo se brindará un presupuesto final del trabajo.

5.1 Ciclo de vida del trabajo

Según *PMBok Guide*, este trabajo puede entenderse y desarrollarse como un proyecto, pues está compuesto de un conjunto de actividades temporales para crear un servicio o producto único, un metabuscador para la búsqueda de apartamentos de alquiler habitual. Al ser un proyecto es posible fragmentarlo en fases con entidad propia y un fin concreto. En la siguiente imagen se puede apreciar de manera gráfica su ciclo de vida:

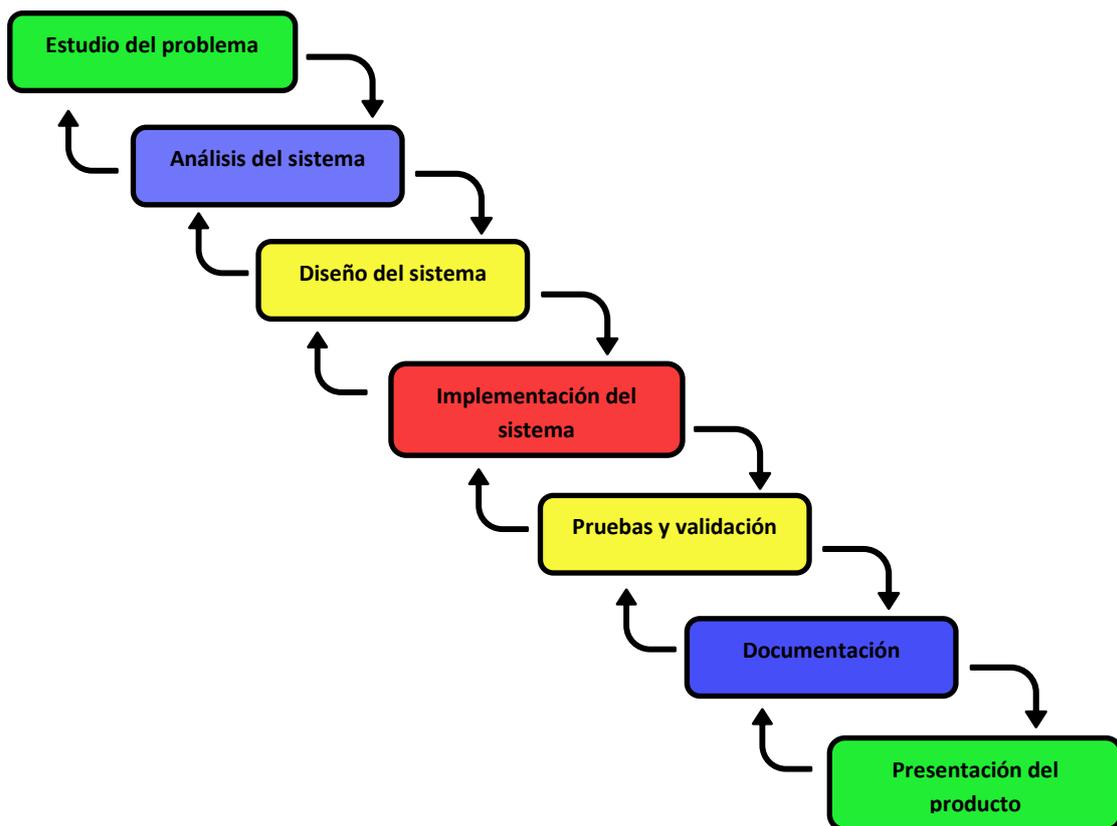


Figura 41. Ciclo de vida en cascada del trabajo.

Como se puede apreciar, las fases siguen una secuencia marcada y tienen la posibilidad de retroalimentarse. Esta última característica es esencial pues hay la oportunidad de

retroceder fases si se precisa de alguna modificación para la correcta finalización del trabajo. Este modelo es conocido como ciclo de vida en cascada con retroalimentación.

5.2 Fases de desarrollo

El objetivo de este apartado es detallar en mejor medida todas las fases que componen el desarrollo completo e íntegro del metabuscador DreamHome. Para tal fin, se ha necesitado una herramienta de gestión de proyectos. En concreto se ha utilizado el programa software *ProjectLibre* pues es muy eficiente y de código libre.

El desarrollo e implementación de este trabajo está compuesto por seis fases, apreciables en la figura X del apartado anterior. En alguna de ellas se ha tenido que regresar a la etapa anterior para mejorar o corregir diversas funcionalidades ya implementadas. En concreto, durante el transcurso del trabajo se ha reconfigurado ciertas partes de la arquitectura del sistema pues al momento de programar las herramientas presentes en client-side (usando JavaScript) surgieron ciertos contratiempos y dificultades para la consecución de los objetivos marcados al inicio del proyecto.

A continuación se procede a mostrar la planificación real que se ha llevado a cabo, desde el inicio hasta la finalización de la aplicación web, visible en la siguiente figura:

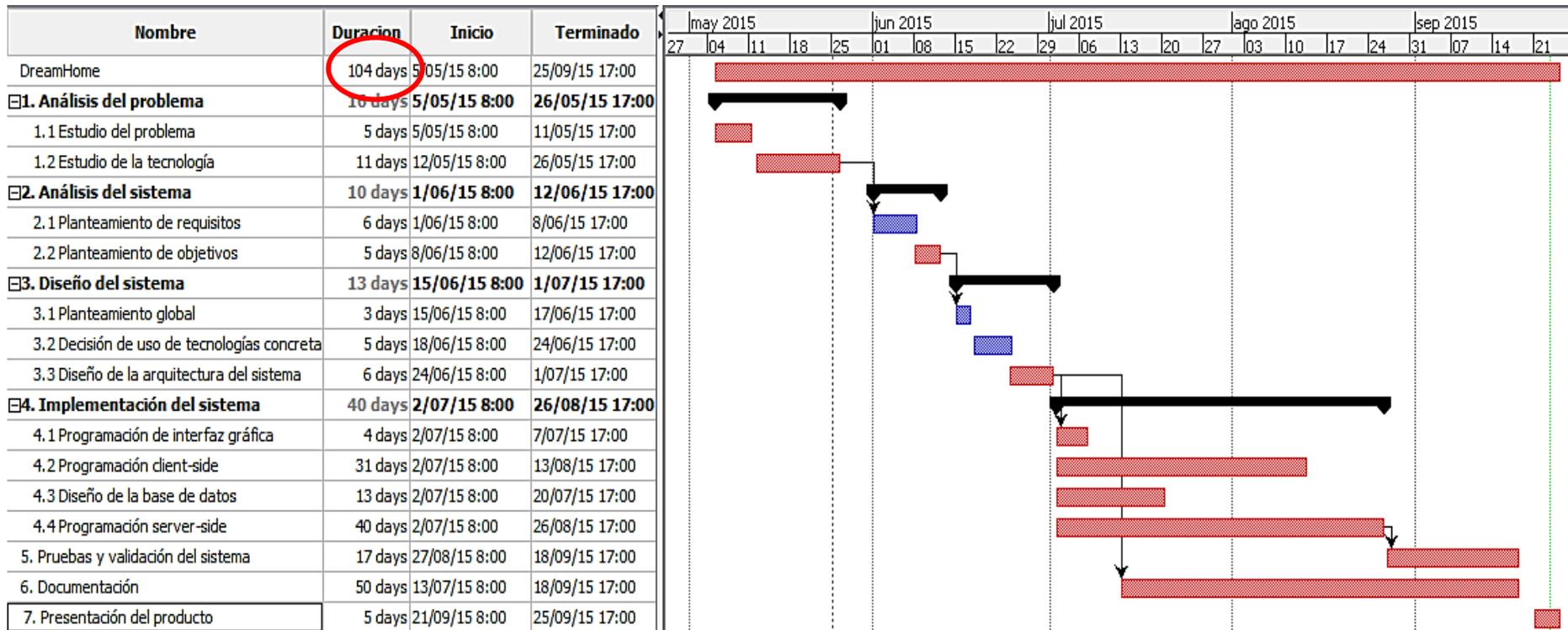


Figura 42. Diagrama de Gantt del metabuscador.

Como se puede apreciar, se ha utilizado la herramienta gráfica diagrama de Gantt con el objetivo de definir las estimaciones de duración temporal para cada fase y sus actividades correspondientes. Hay actividades como la 4.1, 4.2, 4.3 y 4.4 que se solapan pues son independientes unas de otras, no necesitan tareas predecesoras para su comienzo. En el caso opuesto, se tiene que la actividad 2.1 depende de la 1.2 para empezar a desarrollarse y la 3.1 depende de la 2.2, entre otras.

El dato rodeado en rojo considera que la realización de este trabajo se ha llevado a cabo en 104 días, pero no es una cifra del todo real pues *ProjectLibre* no contempla los días de fines de semana. Sin embargo, es posible concretar que la duración de este trabajo es de aproximadamente cuatro meses y medio.

5.3 Presupuesto

En este apartado se detallarán los costes aplicados debido a las actividades contempladas en las fases de desarrollo y realizadas por el personal junto con los del material necesario para llevar a cabo un proyecto de esta índole, para finalmente desglosar el presupuesto final.

5.3.1 Coste de personal

En primer lugar se debe detallar el personal que ha sido necesario para abordar el desarrollo e implementación de esta aplicación.

Se han necesitado únicamente dos personas: el tutor que cumple la función de cliente y el alumno que integra a todo el personal encargado de la elaboración del producto que dicho cliente desea. Por ello, el alumno será el que se ocupe de analizar, diseñar e implementar la aplicación, además de asumir el rol intrínseco de jefe de proyecto.

En la figura X se observa que el desarrollo de este producto se realiza en 104 días sin contar los fines de semana y se ha estimado un tiempo razonable de 4 meses y medio para finalizarlo. Si se asume que cada mes tiene 4 semanas, entonces habrá un total de 36¹ días que deberían sumarse a los 104 contemplados por ProjectLibre. En total se tarda 140 días en entregar el producto final al cliente. Se ha decidido tener una jornada laboral de 4 horas (de media) al día. Se puede asumir que esta cifra se ha cumplido porque aunque no se trabaje en el producto todos los días hay otros en los cuales se invierten muchas más horas de dicha media. Con las cifras anteriores se obtiene un total de 560 horas dedicadas al desarrollo de la aplicación. El detalle de los costes de personal se puede apreciar en la siguiente tabla:

COSTES TOTALES DEL PERSONAL			
Rol	Horas	Coste (€/hora)	Coste total (€)
Jefe de proyecto	35	50	1.750
Analista	40	37	1.480
Diseñador	85	33	2.805
Programador	400	20	8.000
TOTAL			14.035

Tabla 2. Costes totales del personal.

¹ En 4,5 meses hay 4 semanas*2 días*4 meses + 2 semanas*2 días = 36 días de fines de semana (sábados y domingos) que ProjectLibre no contempla.

5.3.2 Coste de material

Se muestra a continuación el coste derivado del material necesario para poder llevar a cabo el desarrollo de esta aplicación web.

COSTES TOTALES DE MATERIAL			
Material	Precio (€/unidad)	Período de Amortización	Coste total (€)
Equipo Intel Pentium processor 2.1 GHz	499	5 años	83,8
Microsoft Windows 7	160	6 años	24,0
Microsoft Office 2010	96	5 años	10,0
TOTAL			117,8

Tabla 3. Costes totales del material.

5.3.3 Resumen de costes

Finalmente, en este apartado se obtiene el coste total de la realización del proyecto siendo la suma de los costes de los apartados anteriores:

COSTE TOTAL DE DREAMHOME	
Tipo de coste	Coste total (€)
Personal	14.035,0
Material	117,8
TOTAL	14.152,8

Tabla 4. Costes totales asociados a DreamHome

Como se puede apreciar, los costes finales del desarrollo e implementación de este metabuscador ascienden a 14.152,8 € en el transcurso de 4 meses y medio.

Capítulo 6

CONCLUSIONS AND FUTURE LINES / CONCLUSIONES Y LÍNEAS FUTURAS

6.1 Conclusions and future lines

6.1.1 Conclusions

Once completed this work and has done all the necessary checks for proper operation, it is possible to get some conclusions on the various issues that have been addressed in it.

It is correct to say that the Web is a huge domain of data, highly variable and therefore it has a very high complexity. For most people the working of the Internet and of the Web is completely transparent. Currently it is inconceivable not to access it at any place and time because it is a universal service and is always available (or very often). However, this service until not long ago was very limited, so it is possible to appreciate his growth, the great impact caused and the enormous penetration that the Internet has in the lives of people.

Most of them access at the Web through the Internet to find any type of information. Therefore they are widely used web browsers. Among them the most used are the search engines due to they are easy use and quick to obtain concrete results based in titles or keywords. However, there are also directories that contain distributed information in very efficient categories, but they are actually underutilized with respect to the above named. Another technology that is on the rise is the metasearch engine or comparator. They are widely used for information (attractive offers to the user) in sector with high demand such as tourism (travels, hotels, vacation packages...), the automotive (car insurance) and housing sector (apartments for rent).

On the other hand, is palpable the great future that this web applications will have. These do not stop developing through implementing new technology and offering specific services to Internet users.

Specifically, the technology used in this work has allowed developing and implementing successfully the desired web application, a metasearch engine for search apartments for usual rent programmed in Java. With this the main objective has been fulfilled. It has also developed a simple, intuitive and easy use user interface through JavaScript and CSS. Therefore, again, it has been successfully achieved the secondary objective. Finally, the proper functioning of this application implicitly carries the compliance of the rest of marked objectives at the beginning of this work, making it accessible to any user.

In the personal aspect is possible to say that the development of this work has been satisfactory and wealth-producing. The acquired Knowledge along the degree has been established, beating continuing problems. In fact it has managed to increase the interest on the development of these technologies very present on the Internet, besides having checked in firsthand the great importance of carrying out a preliminary investigation before starting a work or a project.

6.1.2 Future lines

Although the finished product complies with all requirements and objectives is possible integrate various functions to improve the application, making it more complete, safe and attractive to the user. The possible improvements will be detailed below:

- **Expanding sources.**

The final version of metasearch engine DreamHome performs searches on the websites of rental apartments: *fotocasa.es* and *idealista.com*. It can be a very small range compared to others metasearch engines that include 3, 4 or up to 5 websites on a single query. If is wanted to position this application as one of the most used it is essential to increase this range by adding other pages such as:

- enalquiler.com
- pisos.com
- tucasa.com

If this implementation is possible should be extra careful with how to manipulate and insert information in the local database.

- **Expanding languages.**

The final version only includes searches in Spanish language. Integrate more languages such as English, Catalan, French or Portuguese would increase demand in queries.

- **Expanding geographic.**

This web application provides searches only in Spanish territory, If the claims are high is would shuffled the possibility of expanding the query to others countries. A clear example is *Idealista*. His website is capable of searching in Italy and Portugal.

If this improvement is carried out should be careful about the data processing, especially in the manipulation and insertion of this data in the local database. Rent or

buy servers should be considered (e.g. Amazon) on which to install the application. [33]

- **Expanding search range.**

The final version of this metasearch engine allows searching a particular town and offers two filters such as the maximum price and the number of rooms. To improve the search options could be added the filter of the number of bathrooms and the consultation could allow inserts the postcode or neighborhood.

- **Improving graphic design.**

The final product should call the visual attention of users. If it has a good design (without excess) it will be more attractive. To accomplish this improvement, frameworks or templates could be used instead of manually generating dynamic web pages.

- **Manager access.**

The final version does not include manager access in the user interface. This improvement may allow them access to the application, prior login with user and password, to perform continuous testing of maintaining. Currently the manager does it with the command console of the operating system in use.

- **Mobile interface.**

Last but not least important, a module for smartphones could be integrated to improve the display of application content. Many Internet users access the Web through these types of phones, so the user experience would increase.

6.2 Conclusiones y líneas futuras

6.2.1 Conclusiones

Una vez finalizado este trabajo y haber hecho todas las comprobaciones pertinentes para su correcto funcionamiento, es posible sacar algunas conclusiones sobre los distintos temas que han sido tratados en el mismo.

Es correcto decir que la Web es un enorme dominio de datos, muy variable y por ello tiene una complejidad muy elevada. Para la mayoría de las personas el funcionamiento de

Internet, y con ella la Web, es totalmente transparente. En la actualidad es impensable no poder acceder a ella en cualquier lugar e instante pues es un servicio universal y está siempre disponible (o suele estarlo muy a menudo). Sin embargo, este servicio hasta hace no mucho estaba muy limitado, por lo que se puede apreciar su crecimiento, el gran impacto que ha causado y el enorme grado de penetración que tienen en la vida de las personas.

La mayoría de éstas acceden a la Web por medio de Internet para buscar cualquier tipo de información. Por ello son extensamente utilizados los navegadores web y, en concreto, los buscadores. Entre ellos, los más usados son los motores de búsqueda debido a su facilidad de uso y la rapidez en la que obtienen resultados de carácter concreto basados en títulos o palabras clave. Sin embargo, existen también los directorios que contienen información distribuida en categorías muy eficientes en búsquedas de carácter más general, pero son realmente poco utilizados con respecto a los anteriormente nombrados. Otra tecnología que está en aumento es la de los metabuscadores o comparadores. Son muy utilizados para encontrar información (ofertas atractivas al usuario) en sectores con gran demanda como el turístico (viajes, hoteles, paquetes vacacionales...), el de automoción (seguros de coches) y el inmobiliario (pisos de alquiler).

Por otro lado, es palpable y esperanzador el gran futuro que tienen las aplicaciones web. No dejan de desarrollarse implementando nueva tecnología y ofreciendo servicios concretos a los usuarios de Internet.

En concreto, la tecnología usada en este trabajo ha permitido desarrollar e implementar satisfactoriamente la aplicación web deseada, un metabuscador para la búsqueda de apartamentos de alquiler habitual programado en Java. Por ello se ha cumplido el objetivo principal de este trabajo. También se ha conseguido elaborar una interfaz de usuario sencilla, intuitiva y de fácil manejo gracias a JavaScript y CSS. Por ello, de nuevo, se ha alcanzado de manera satisfactoria el objetivo secundario. Finalmente, el correcto funcionamiento de esta aplicación lleva implícitamente el cumplimiento del resto de los objetivos marcados al inicio de este trabajo, haciéndolo accesible a cualquier usuario.

En cuanto al aspecto personal es posible decir que ha resultado gratamente satisfactorio y enriquecedor el desarrollo completo de este trabajo. Se han afianzado conocimientos adquiridos en el transcurso de la carrera, superando los continuos problemas surgidos en el transcurso del mismo. De hecho ha conseguido aumentar el interés sobre el desarrollo de estas tecnologías muy presentes en Internet, además de haber comprobado en primera mano la gran importancia que tiene el llevar a cabo una investigación previa ante cualquier trabajo o proyecto.

6.2.2 Líneas futuras

Aunque el producto elaborado cumple completamente con todos los requisitos y objetivos es posible integrar diversas funcionalidades que permitan mejorar la aplicación

haciéndola más completa, segura y atractiva al usuario. A continuación se detallarán las posibles mejoras aplicables a este trabajo:

- **Ampliación de fuentes.**

La versión final de metabuscador DreamHome realiza búsquedas en las páginas web de pisos de alquiler *fotocasa.es* e *idealista.com*. En cierto aspecto puede resultar un rango muy pequeño en comparación con otros metabuscadores que contemplan 3, 4 o hasta 5 páginas en una misma consulta. Por ello, si se desea posicionar esta aplicación como una de las más utilizadas es imprescindible aumentar este rango añadiendo otras páginas como:

- enalquiler.com
- pisos.com
- tucasa.com

En caso de implementarse esta posible mejora se debería tener mayor cuidado con la forma de manipular e insertar la información en la base de datos local.

- **Ampliar el lenguaje de búsqueda.**

La versión final contempla búsquedas en castellano. Integrar un módulo de búsquedas en diversos lenguajes como inglés, catalán, francés o portugués haría aumentar la demanda de consultas.

- **Ampliación geográfica.**

Esta aplicación web contempla búsquedas sólo en el territorio español. Si las pretensiones de uso de la misma son altas debería barajarse la posibilidad de ampliar la consulta a otros países. Un ejemplo claro de ello es el portal *Idealista* pues es capaz de realizar búsquedas en Italia y en Portugal.

Si se lleva a cabo esta mejora se debería tener cuidado con el tratamiento de datos, especialmente en la manipulación e inserción de datos en la base de datos local. Se debería contemplar la opción de alquilar o comprar servidores (en Amazon por ejemplo) en los cuales alojar dicha aplicación. [33]

- **Ampliar el rango de búsqueda.**

La versión final de este metabuscador permite realizar búsquedas en un municipio en concreto y ofrece dos filtros como el precio máximo y el número de habitaciones. Para

mejorar las opciones de búsqueda pueden añadirse el filtro de número de baños y la consulta mediante la introducción del código postal o el barrio en donde se desea encontrar anuncios de este tipo de estancia.

- **Mejora del diseño gráfico.**

El producto final debe llamar la atención visual de los usuarios. Cuanto mejor diseño tenga (sin excesos) les resultará más atractivo. Para llevar a cabo esta mejoría podrían usarse frameworks o plantillas en lugar de generar manualmente las páginas web dinámicas.

- **Acceso a administradores.**

La versión final no incluye un acceso para administradores en la interfaz de usuario. Esta posible mejoría les permitiría acceder a la aplicación, con previo inicio de sesión mediante usuario y contraseña, para realizar las continuas pruebas que se deben llevar a cabo para mantener la aplicación. Actualmente el administrador lo hace desde la consola de comandos del sistema operativo en uso.

- **Interfaz móvil.**

Por último y no menos importante, podría integrarse un módulo para smartphones pues mejoraría la visualización del contenido de la aplicación. Muchos usuarios de Internet acceden a la Web a través de este tipo de teléfonos, por lo que la experiencia del usuario aumentaría.

Capítulo 7 REFERENCIAS

- [1] Alexa. *The top 500 sites on the web*. <http://www.alexa.com/topsites>
- [2] Amazon Web Services. <https://aws.amazon.com/es/ec2/?hp=tile>
- [3] Apache Tomcat. <http://tomcat.apache.org/>
- [4] Apache Tomcat Versions. <http://tomcat.apache.org/whichversion.html>
- [5] Diario económico Expansión. *Artículo sobre construcción e inmobiliario*. 2015. <http://www.expansion.com/empresas/inmobiliario/2015/08/16/55d06271268e3ee2458b456e.html>
- [6] Posicionamiento Internacional. Artículo: Cuota de mercado de buscadores por países. 2012. <http://www.posicionamientointernacional.com/cuota-de-mercado-de-buscadores-por-paises/>
- [7] Instituto Nacional de Estadística. *Encuesta sobre Equipamiento y Uso de Tecnologías de Información y Comunicación en los Hogares*. Año 2014. <http://www.ine.es/prensa/np864.pdf>
- [8] González, Carlos. ADSLZone. 2014. <http://www.adslzone.net/2014/08/01/el-9802-de-los-espanoles-usamos-google-como-buscador/>
- [9] Dolores, María. *Historia de los Buscadores*. 2011. <http://histinf.blogs.upv.es/2011/01/11/1808/>
- [10] Instituto de Estudios Económicos. *El sector inmobiliario en España*. 2014. http://www.cenavarra.es/documentos/ficheros_comunicacion/sectorinmobiliario.pdf
- [11] Flanagan, David. *JavaScript: The Definitive Guide*. 6ª Ed. O'Reilly, 2011.
- [12] GlassFish Server. <https://glassfish.java.net/>
- [13] Internet Information Services. <http://www.iis.net/>
- [14] JDBC Data Sources. Apache Tomcat. http://tomcat.apache.org/tomcat-8.0-doc/jndi-resources-howto.html#JDBC_Data_Sources
- [15] jQuery User Interface. <http://jqueryui.com/>
- [16] Jsoup: Java HTML Parser. <http://jsoup.org/>
- [17] Lamarca Lapuente, María Jesús. *Hipertexto: El nuevo concepto de documento en la cultura de la imagen. Los orígenes de la Web*. Actualizado en 2013. http://www.hipertexto.info/documentos/h_www.htm
- [18] Lamarca Lapuente, María Jesús. *Hipertexto: El nuevo concepto de documento en la cultura de la imagen. Tipos de buscadores*. Actualizado en 2013. http://www.hipertexto.info/documentos/tipos_buscador.htm

- [19] Lamarca Lapuente, María Jesús. *Hipertexto: El nuevo concepto de documento en la cultura de la imagen. HTML*. Actualizado en 2013.
<http://www.hipertexto.info/documentos/html.htm>
- [20] LibrosWeb. *Breve historia de JavaScript*.
http://librosweb.es/libro/javascript/capitulo_1/breve_historia.html
- [21] Luján Mora, Sergio. *Programación de Aplicaciones Web: Historia. Principios Básicos y Clientes Web*. España 2002.
- [22] M. Sarries, Nicolás. *Los comparadores en Internet, un negocio en alza en España*. 2014.
<http://www.20minutos.es/noticia/2036788/0/comparadores/ofertas/internet/>
- [23] Menéndez-Barzanallana Asensio, Rafael. *Historia del desarrollo de aplicaciones web*.
<http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Historia-desarrollo-aplicaciones-web.html>
- [24] Microsoft SQL Server. <http://www.microsoft.com/es-es/server-cloud/products/sql-server/>
- [25] MySQL. <https://www.mysql.com/>
- [26] Parrilla Monrocle, Marta. *La Internet que no aparece en los buscadores*. 2012.
<http://hdl.handle.net/10016/16759>
- [27] PostgreSQL. <http://www.postgresql.org.es/>
- [28] Poulos, Andreas. *Greenlight: How search engine market shares look around the world featuring Bing, Yahoo and Baidu and others*. 2010.
<http://www.greenlightdigital.com/blog/posts/how-search-engine-market-shares-look-around-the-world-featuring-bing-yahoo-and-baidu-and-others/>
- [29] ProjectLibre. <http://www.projectlibre.org/>
- [30] Proyecto Nodo50. *Orígenes y evolución de Internet*.
<http://www.nodo50.org/manuales/internet/1.htm>
- [31] The Apache Software Foundation. <http://apache.org/>