

# ANÁLISIS DE LA DEMANDA DE EMPLEO UTILIZANDO TÉCNICAS DE BIG DATA



TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

Realizado por:

Roberto Barrasús Orza

Dirigido por:

Antonio Sarasa Cabezuelo

Curso académico 2017 2018



# Agradecimientos

Quiero dar las gracias a mi tutor del Trabajo Fin de Grado, Antonio Sarasa, por haber hecho posible este proyecto con su esfuerzo y dedicación, y haber aportado grandes ideas que han sido realmente útiles.

A todos mis compañeros y profesores por haber hecho de estos años de universidad los mejores.

A mi familia y amigos por disfrutar siempre de su apoyo, que es imprescindible.

# Resumen

En este proyecto se ha desarrollado un sistema capaz de difundir información sobre la demanda de empleo y el estado del mercado laboral a través de la recopilación, limpieza, almacenamiento, procesamiento y visualización de datos.

Los datos que explota este sistema son recopilados, por un lado, del Servicio Público de Empleo Estatal, entre los que se encuentra información sobre el desempleo y los contratos realizados. Por otro lado, se obtienen todas las ofertas de trabajo en tiempo real que son publicadas en la plataforma de búsqueda de empleo InfoJobs.net.

Para su implementación, han sido utilizados los sistemas de bases de datos que mejor se adaptan a las particularidades y necesidades de cada tipo de información como son el volumen, la velocidad, la variedad o la veracidad de los datos.

El sistema se ha desarrollado como una aplicación web que se adapta a diferentes tipos de dispositivos para exponer al usuario, no solo la información relevante sino además adaptada a sus condiciones.

## Palabras clave

Demanda de empleo, mercado laboral, análisis de datos, servicios web, API

# Abstract

In this Project has been developed a system capable of disseminating information on the demand for employment and the labour market. Everything is done through collecting, cleaning, storing, processing and data display.

The data provided in this system are collected through two channels. On one hand, by Public Service of State Employment, in which you can obtain information about unemployment and contracts carried out during a period. On the other hand, you can find information about job offers in real-time thanks to InfoJobs.net. It is a job search tool.

Database systems that best adapt to the particularities and necessities of each type of information have been used for its implementation. Such as volume, speed, variety or truthfulness of the data provided.

The system has been developed as a web application that adapts to several types of devices in order to show the user, both the most relevant and necessary information at each time, and the one that best adapts to their conditions and preferences.

# Keywords

Employment demand, labour market, data analysis, web services, API

# Índice

Agradecimientos	3
Resumen	4
Palabras clave	4
Abstract	5
Keywords	5
Índice	6
Índice de figuras	8
1. Introducción	10
I. Motivación	10
II. Objetivos	11
III. Estado del arte	12
IV. Estructura de la memoria	13
2. Especificación	14
3. Tecnologías empleadas	22
PHP	22
MongoDB	22
Tecnologías web	22
Tecnologías auxiliares	23
4. Arquitectura de la aplicación	24
5. Modelo de datos	25
I. Modelo relacional	25
i. Base de datos 'empleo'	25
ii. Base de datos 'login'	31
II. Modelo orientado a documentos	32
6. Diseño de la aplicación	35
7. Implementación	37
8. Conclusiones, aportación personal y trabajo futuro	43
I. Conclusiones	43
II. Aportación personal	43
III. Trabajo futuro	43
Bibliografía	45
Anexos	47
I. Manual de instalación	47

1. Instalación de paquetes necesarios	48
2. Despliegue de la aplicación	49
II. Manual de usuario	50

# Índice de figuras

1. Diagrama de casos de uso para el usuario no registrado.	20
2. Diagrama de casos de uso para el usuario registrado.	21
3. Esquema representativo de la arquitectura del sistema.	24
4. Tablas de datos sin procesar de la Base de Datos 'empleo'.	26
5. Tablas de la Base de Datos 'empleo' con los datos limpios.	27
6. Información sobre las tablas de la base de datos 'empleo'.	30
7. Base de Datos 'login'.	31
8. Documento de ejemplo de la colección 'ofertas'.	33
9. Volumetría de la colección 'ofertas'.	33
10. Tamaño de la colección 'ofertas'.	33
11. Rueda de colores de Adobe. Rango de tonos elegido en la aplicación web.	35
12. Diagrama de clases.	37
13. Establecimiento de una conexión con MongoDB.	38
14. Establecimiento de una conexión con MariaDB.	38
15. Una de las funciones de la clase Daocontratos.	38
16. Parte de la clase Contratos.	39
17. Vista de datos sobre la temporalidad de los contratos.	39
18. Paneles de la página de inicio de sesión.	40
19. Página de inicio de sesión.	40
20. Script para generar el gráfico de contratos dados de alta por sexos.	41
21. Utilización de la API de InfoJobs.	41
22. Inserción de las ofertas y tratamiento de las excepciones.	42
23. Script de despliegue del servidor.	42
24. Página de inicio de Ubuntu Server 16.04.	47
25. Instalación de MariaDB en línea de comandos.	48
26. Instalación de Apache2 en línea de comandos.	48
27. Instalación de php 7 en línea de comandos.	48
28. Instalación de paquetes adicionales en línea de comandos.	48
29. Instalación de MongoDB en línea de comandos.	48
30. Instalación de paquetes de MongoDB para php.	48
31. Acceso a mysql desde la línea de comandos.	49

32. Creación de usuario de MariaDB en línea de comandos.	49
33. Carga de script sobre MariaDB en línea de comandos.	49
34. Paneles de la página principal de la aplicación web.	50
35. Menú de la aplicación web.	51
36. Panel con la información de la demanda por categorías.	52
37. Panel de opciones de sesión.	52
38. Formulario de inicio de sesión.	53
39. Mensaje al intentar acceder a secciones de registro necesario.	54
40. Pasos para completar el registro.	55
41. Ofertas recomendadas.	56
42. Ofertas con mayor salario.	56
43. Datos de la cuenta de usuario.	57

# 1. Introducción

## I. Motivación

La motivación de este proyecto es la utilidad social que puede aportar una herramienta como esta para facilitar el acceso a la información relativa a la demanda de empleo. Este tipo de herramienta puede ser útil tanto para los futuros estudiantes universitarios como para los empleados que quieran redirigir su carrera profesional para obtener mayores oportunidades en el mercado laboral.

El auge de la ciencia de datos y la revolución digital que se está produciendo, hacen interesante el proyecto por ofrecer la posibilidad de emplear las tecnologías y técnicas empleadas en todo el ciclo de vida de los datos como son la recopilación, la limpieza, el almacenamiento, el procesamiento y la visualización.

## II. Objetivos

El objetivo principal de este Trabajo Fin de Grado ha sido desarrollar una herramienta informática con la que se pueda acceder a información sobre el mercado laboral y la demanda de empleo en tiempo real.

Los objetivos específicos que se han tenido en cuenta para el desarrollo del proyecto son:

1. Investigar y recopilar información sobre qué datos públicos y accesibles tienen valor para cumplir el objetivo principal indicado más arriba.
2. Estudiar las tecnologías necesarias para llevar a cabo su recopilación, almacenamiento, procesamiento y visualización.
3. Clasificar los datos y determinar cuáles de ellos son relevantes para mostrar directamente al usuario o para ser procesados y sacar conclusiones de ellos.
4. Desarrollar una aplicación web con una interface amigable que facilite a los usuarios su utilización, accesibilidad y aprovechamiento de toda su funcionalidad.

### III. Estado del arte

Los sistemas que han sido encontrados que pueden tener alguna similitud con la aplicación desarrollada son los siguientes:

1. Artículos de periódico y blogs en los que se listan los perfiles más demandados y más cotizados por cada categoría y describen cuales son las tareas de cada perfil [1], [2] y [3].
2. Artículos de periódico en los que se mencionan los perfiles más polivalentes [4].
3. Portal web en el que listan las carreras universitarias más demandadas en función de las ofertas publicadas en su web durante todo un año. Esta información no está desglosada por localización ni por meses [5].
4. Estudio del Ministerio de Educación Cultura y Deporte y su correspondiente artículo en un periódico en el que se analiza la información extraída de este informe en el que se analiza el éxito de los estudiantes en su búsqueda de empleo, segregado por universidades. Este informe se obtiene cruzando de los titulados universitarios con los de afiliados a la seguridad social [6] y [7].

## IV. Estructura de la memoria

Esta memoria se estructura por capítulos y a continuación se detalla brevemente el contenido de cada uno de ellos.

En el primer capítulo, la introducción, se describe brevemente el pretexto del proyecto dividido en las secciones de motivación, objetivos, estado del arte y estructura de esta memoria.

En el segundo capítulo se describe la especificación del sistema creado.

En el tercer capítulo se mencionan las tecnologías que se han empleado en el desarrollo del proyecto además de las razones por las que se han elegido cada una de ellas.

En el cuarto capítulo se describe la arquitectura del sistema detallando las ventajas que aporta.

En el quinto capítulo se especifica el modelo de datos.

En los capítulos sexto y séptimo se trata el tema del diseño y la implementación.

En el último capítulo se plantean unas conclusiones finales para este proyecto y se establecen líneas de trabajo futuro.

La memoria finaliza con una bibliografía donde se listan todas las referencias que han sido utilizadas, y dos anexos.

En el primer anexo se instruye cómo instalar paso a paso el sistema.

En el segundo anexo se explica el correcto funcionamiento de la aplicación web para los usuarios independientemente del conocimiento técnico que tengan.

## 2. Especificación

Los casos de uso son los siguientes:

Identificador del caso de uso	CU1
Nombre	Registro de usuario
Actores	Cualquier usuario
Precondiciones	
Postcondiciones	El usuario obtendrá acceso al sistema
Descripción	El usuario podrá registrarse en el sistema ingresando algunos datos como nombre de usuario, una contraseña y un correo electrónico
Flujo	<ol style="list-style-type: none"><li>1. El usuario pulsa en el botón de 'Registrarse' en cualquiera de las páginas de la aplicación web.</li><li>2. El usuario ingresa los datos que se requieren en el formulario de registro.</li><li>3. El usuario pulsa el botón de 'Registrarse'.</li></ol>

Identificador del caso de uso	CU2
Nombre	Inicio de sesión
Actores	Cualquier usuario
Precondiciones	Disponer de una cuenta de usuario habiendo realizado un registro en el sistema anteriormente.
Postcondiciones	El usuario ha ingresado en el sistema
Descripción	El usuario podrá iniciar sesión en el sistema para obtener información personalizada
Flujo	<ol style="list-style-type: none"> <li>1. El usuario pulsa en el botón de 'Iniciar sesión' que aparece en todas las pantallas de la aplicación.</li> <li>2. El usuario ingresa los datos con los que previamente se registró en el sistema.</li> <li>3. El usuario pulsa el botón 'Iniciar sesión'.</li> </ol>

Identificador del caso de uso	CU3
Nombre	Visualizar noticias
Actores	Cualquier usuario
Precondiciones	
Postcondiciones	El usuario obtiene las noticias.
Descripción	Se obtienen las noticias relacionadas con el mercado laboral.
Flujo	<ol style="list-style-type: none"> <li>1. El usuario accede a la página principal de la aplicación web.</li> </ol>
Flujo alternativo	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón 'Inicio' desde cualquier página de la aplicación web.</li> </ol>

Identificador del caso de uso	CU4
Nombre	Visualizar demanda de empleo por categoría
Actores	Cualquier usuario
Precondiciones	
Postcondiciones	El usuario obtiene las categorías más demandadas
Descripción	Se obtiene una gráfica con todas las categorías ordenadas descendientemente por la demanda que precisan.
Flujo	<ol style="list-style-type: none"> <li>1. El usuario pulsa en el botón de 'Por categoría' dentro del apartado de 'Demanda de empleo' en la barra lateral de la aplicación web.</li> </ol>

Identificador del caso de uso	CU5
Nombre	Visualizar demanda de empleo por provincia
Actores	Cualquier usuario
Precondiciones	
Postcondiciones	El usuario obtiene una lista con las provincias con más demanda de empleo.
Descripción	Se obtiene una gráfica con las provincias que más demanda de empleo tienen ordenadas descendientemente.
Flujo	<ol style="list-style-type: none"> <li>1. El usuario pulsa en el botón de 'Por provincia' dentro del apartado de 'Demanda de empleo' en la barra lateral de la aplicación web.</li> </ol>

Identificador del caso de uso	CU6
Nombre	Visualizar temporalidad de contratos
Actores	Cualquier usuario
Precondiciones	
Postcondiciones	Se obtienen dos gráficos sobre la temporalidad de los contratos
Descripción	El usuario visualiza dos gráficos. En el primero se muestra el porcentaje de contratos de cada tipo (temporales e indefinidos). En el segundo este mismo dato se desglosa por sexos.
Flujo	<ol style="list-style-type: none"> <li>1. El usuario pulsa en el botón de 'Por temporalidad' dentro del apartado de 'Contratos' en la barra lateral de la aplicación web.</li> </ol>

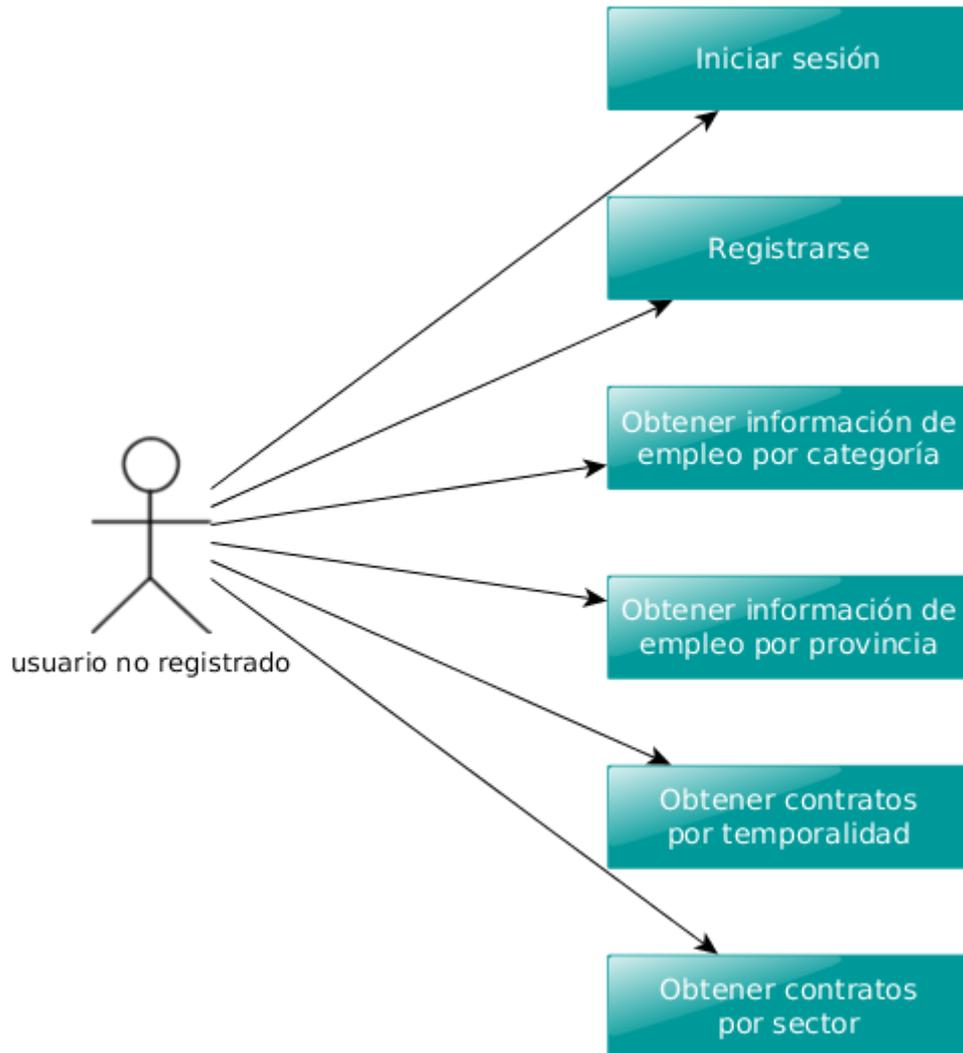
Identificador del caso de uso	CU7
Nombre	Visualizar contratos creados por sectores
Actores	Cualquier usuario
Precondiciones	
Postcondiciones	Se obtienen dos gráficos sobre los sectores de los contratos creados
Descripción	El usuario visualiza dos gráficos. En el primero se muestra el porcentaje de contratos creados en el mes pasado de cada sector (agricultura, industria, construcción y servicios). En el segundo este mismo dato de hace diez años.
Flujo	<ol style="list-style-type: none"> <li>1. El usuario pulsa en el botón de 'Por sector dentro del apartado de 'Contratos' en la barra lateral de la aplicación web.</li> </ol>

Identificador del caso de uso	CU8
Nombre	Visualización de ofertas recomendadas
Actores	Usuario registrado
Precondiciones	Disponer de cuenta de usuario
Postcondiciones	Se obtienen las ofertas recomendadas
Descripción	En base a los datos que facilita el usuario en su registro, se obtienen las ofertas que más se adaptan al usuario.
Flujo	<ol style="list-style-type: none"> <li>1. El usuario registrado accede a la página de ofertas recomendadas desde cualquier pantalla de la aplicación pulsando el botón de 'Ofertas recomendadas'.</li> </ol>

Identificador del caso de uso	CU9
Nombre	Visualización de ofertas cuyo salario es mayor
Actores	Usuario registrado
Precondiciones	Disponer de cuenta de usuario
Postcondiciones	Se obtienen las ofertas con mayor salario ordenadas descendientemente
Descripción	En base a los datos que facilita el usuario en su registro, se obtienen las ofertas relacionadas con el usuario cuyo salario es mayor.
Flujo	<ol style="list-style-type: none"> <li>1. El usuario registrado accede a la página de ofertas con mayor salario desde cualquier pantalla de la aplicación pulsando el botón de 'Ofertas con mayor salario'.</li> </ol>

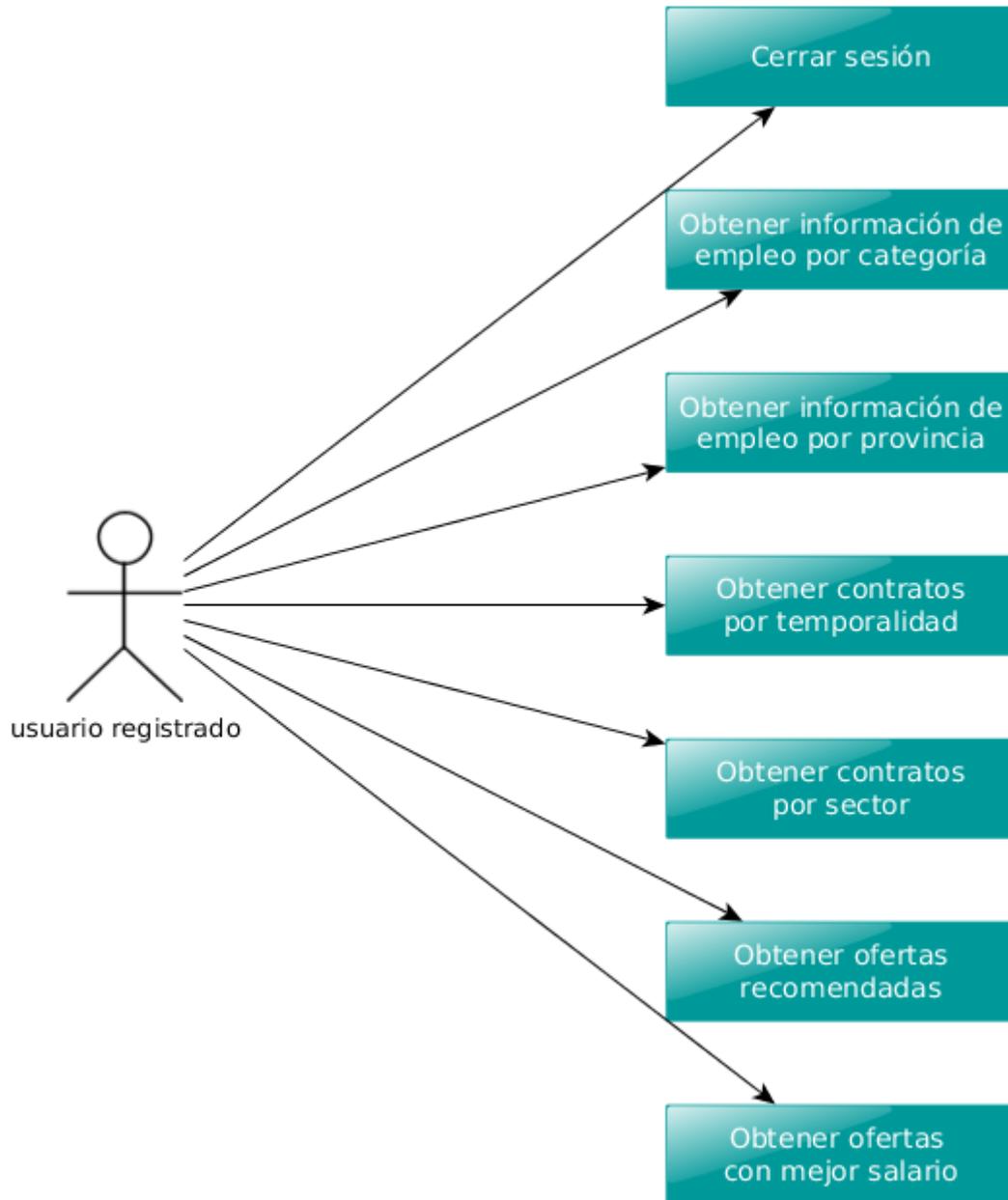
Identificador del caso de uso	CU10
Nombre	Cerrar sesión
Actores	Usuario registrado
Precondiciones	Disponer de cuenta de usuario y haber iniciado sesión
Postcondiciones	Se cierra la sesión de usuario
Descripción	Se termina la sesión en todas las pestañas en las que esté abierta la aplicación web
Flujo	<ol style="list-style-type: none"> <li>1. El usuario pulsa en el botón de 'Iniciar sesión' que aparece en todas las pantallas de la aplicación.</li> </ol>

Las funcionalidades implementadas del usuario sin registrar, como se muestra en el diagrama de casos de uso de la Figura 1, son iniciar sesión, registrarse, obtener demanda de empleo por categoría, obtener demanda de empleo por provincia, obtener temporalidad de contratos y obtener contratos por sectores.



1. Diagrama de casos de uso para el usuario no registrado.

Por otro lado, el usuario registrado, como se puede ver en el diagrama de casos de uso de la Figura 2, tiene los mismos salvo iniciar sesión y registrarse y añadiendo obtener ofertas recomendadas y obtener ofertas con mayor salario.



2. Diagrama de casos de uso para el usuario registrado.

## 3. Tecnologías empleadas

En este apartado se explican las tecnologías empleadas para el desarrollo de este proyecto.

### PHP

El script para la ingesta de información de ofertas de InfoJobs se ha realizado en **PHP** [8]. Este script utiliza una librería de PHP llamada **Httpful** [9] la cual facilita el acceso a la **API de InfoJobs** [10], la cual requiere el uso del protocolo **OAuth 2.0** [11] para realizar peticiones HTTP. Se ha elegido PHP ya que es un lenguaje ampliamente extendido, dispone de gran cantidad de documentación, y su curva de aprendizaje es baja.

### MongoDB

Para el almacenamiento de la información que se extrae de la API de InfoJobs se ha utilizado una base de datos de **MongoDB** [12]. Las ofertas que se procesan de la API están en formato **JSON** [13], tienen un tamaño elevado y son varias decenas de miles al día (unas 50.000). En este sentido, MongoDB es el sistema más apropiado para albergar este tipo de datos. También se emplearon las herramientas **Robomongo** [14] que facilita la administración de las bases de datos de MongoDB mediante una interfaz gráfica y **Postman** [15] que facilita el uso de APIs permitiendo el envío de peticiones a modo de test para aprender sobre su utilización.

### MariaDB

Por otro lado, la información que se ha extraído del Servicio Público de Empleo Estatal del Ministerio de Empleo y Seguridad Social está en formato **CSV** [16]. En este caso, se han utilizado scripts de **SQL** [17] para su ingesta en una base de datos de **MariaDB** [18]. Para la limpieza de estos datos se han creado **triggers** [19] en SQL y se ha utilizado la herramienta **phpMyAdmin** [20] para comprobar el correcto funcionamiento de los scripts y triggers de SQL. Se ha elegido relacional, ya que mediante una consulta es posible la captura de los datos desde el CSV y almacenarlo en tablas.

### Tecnologías web

Para el desarrollo de la aplicación web se han utilizado las siguientes tecnologías: **HTML5** [21], **CSS3** [22] y **JavaScript** [23]. Además, se ha utilizado **Bootstrap** [24] (un framework de CSS para el diseño de la interfaz), **Jquery** [25] (otro framework de JavaScript). Estas tecnologías son muy utilizadas para el desarrollo web, dado que facilitan la creación de aplicaciones accesibles desde cualquier navegador moderno o características tan deseables

como ser responsive (es decir, se adaptan al tamaño de pantalla del dispositivo desde el que son accedidas).

Con respecto a la generación de gráficos, después de un estudio comparando las diferentes librerías se ha elegido **C3.js** [26] ya que su uso es muy simple y se obtienen buenos resultados. Esta librería se ha usado para representar visualmente los datos que han sido obtenidos de las bases de datos de MongoDB y de MariaDB.

## Tecnologías auxiliares

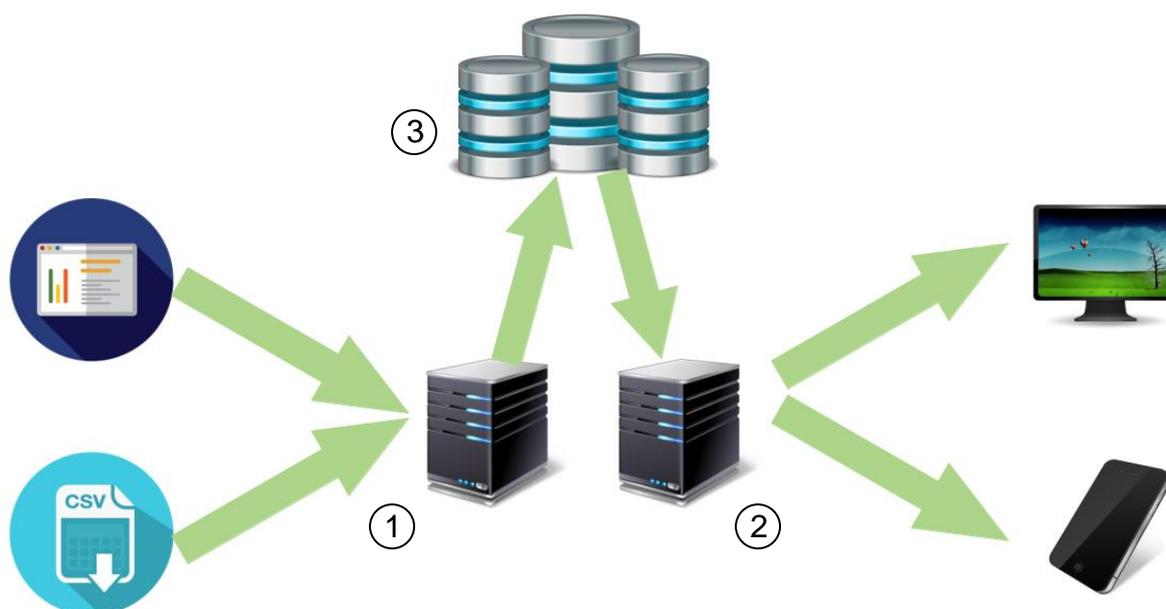
Finalmente, para la puesta en marcha del proyecto, se ha utilizado **Ubuntu Server** [27] en su versión 16.04 LTS como sistema operativo del servidor, además de un script en **Bash** [28] para el despliegue y del demonio **Cron** [29] para la ejecución periódica de los scripts de recopilación de información. Todo ello tanto en una máquina virtual en **Oracle VirtualBox** [30] como en un servidor real posteriormente. La elección de Ubuntu Server sobre otras opciones como Windows Server, se debe a la mayor seguridad, mejor rendimiento (sobre todo en máquinas con pocos recursos), mayores opciones de configuración y la rapidez con la que se instalan los paquetes necesarios.

Se ha creado una máquina virtual adicional con Ubuntu Desktop para facilitar el acceso a la aplicación para su revisión.

## 4. Arquitectura de la aplicación

En este apartado se detalla la arquitectura del sistema y por qué se ha decidido estructurar de esta forma.

Se trata de una aplicación que sigue el modelo Cliente-Servidor [31]. Como se puede observar en la Figura 3, la aplicación está formada por tres módulos. El primero es el de recopilación de los datos que se ejecuta periódicamente, el segundo es la aplicación web que se ejecuta cuando se accede a ella desde un cliente y finalmente el tercero es el sistema de bases de datos que alberga diferentes tipos de ellas.



3. Esquema representativo de la arquitectura del sistema.

El sistema, que utiliza información sensible, sigue un patrón Modelo-Vista-Controlador [32] que aporta mayor seguridad tanto para evitar ataques para obtener información de los usuarios como para evitar el acceso por parte de intrusos al sistema.

El sistema de gestión de usuarios está compuesto por dos controladores, dos vistas y dos modelos que se corresponden al registro y al acceso de los usuarios.

El sistema de visualización de datos tiene un controlador, una vista y un modelo para cada uno de los catálogos de datos que ofrece la aplicación web como por ejemplo el de contratos.

## 5. Modelo de datos

En esta sección se detalla el modelo de datos implementado en la aplicación, así como las tecnologías empleadas para cada tipo de información.

Para almacenar la información que gestiona el sistema se ha decidido utilizar una solución políglota donde se combinan dos modelos de datos.

Por una parte, un modelo relacional donde se ha almacenado la información que procede del Servicio Público de Empleo Estatal [33] (Ministerio de Empleo y Seguridad Social [34]) y por otro lado, la información del sistema de usuarios. Ambas fuentes de información, se componen de datos estructurados en campos y registros, por lo cual se adaptan muy bien al modelo relacional.

Por otra parte, un modelo documental donde se almacena información referida a las ofertas de empleo que se publican en el portal de InfoJobs.net [35].

### I. Modelo relacional

Las dos bases de datos que se han creado han sido las siguientes:

#### i. Base de datos 'empleo'

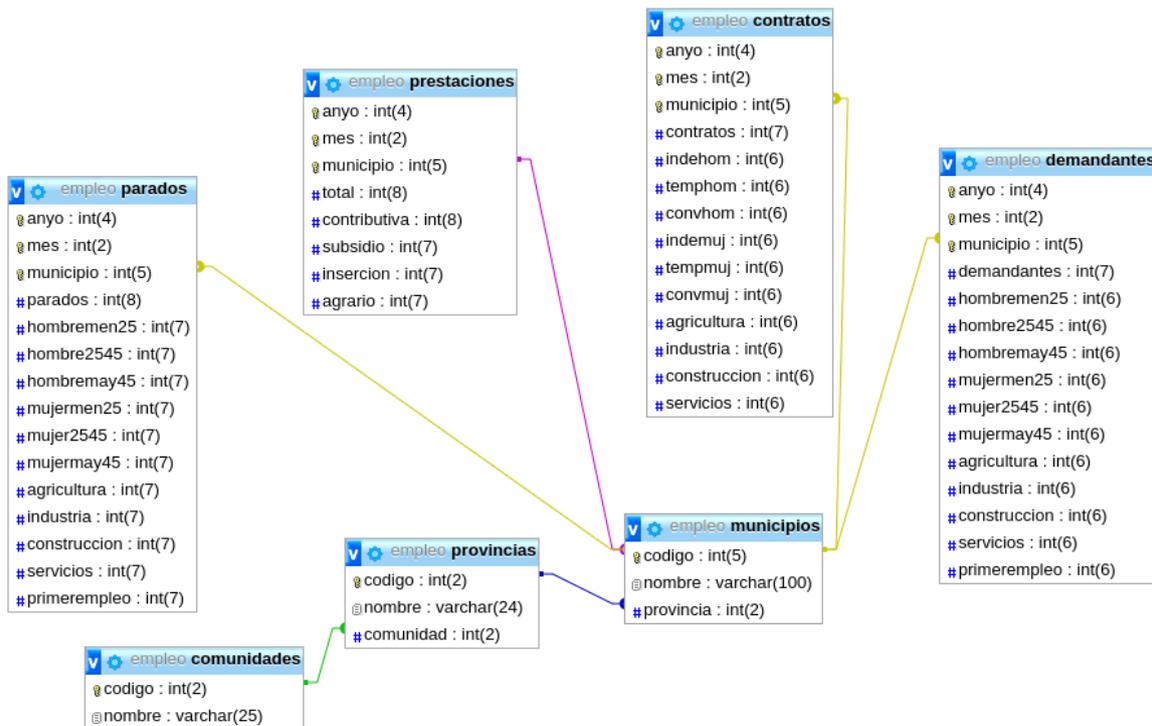
La primera de ellas, **empleo**, tiene once tablas en total. De todas ellas, las cuatro que se muestran en la Figura 4, se utilizan para almacenar los datos cargados de los ficheros CSV del Servicio Público de Empleo Estatal. Posteriormente se realiza una limpieza de estos datos, se inserta la información relevante e imprescindible en otras cuatro tablas y finalmente se vacían las tablas con los datos en crudo.

empleo rawdemandantes	empleo rawparados	empleo rawcontratos	empleo rawprestaciones
fecha : int(6)	fecha : int(6)	fecha : int(6)	fecha : int(6)
mesanyo : varchar(20)	mesanyo : varchar(20)	mesanyo : varchar(20)	mesanyo : varchar(20)
codigocomunidad : int(2)	codigoccaa : int(2)	codigocomunidad : int(2)	codigoccaa : int(2)
comunidad : varchar(25)	ccaa : varchar(25)	comunidad : varchar(25)	ccaa : varchar(25)
codigoprovincia : int(2)	codigoprovincia : int(2)	codigoprovincia : int(2)	codigoprovincia : int(2)
provincia : varchar(24)	provincia : varchar(24)	provincia : varchar(24)	provincia : varchar(24)
codigomunicipio : int(5)	codigomunicipio : int(5)	codigomunicipio : int(5)	codigomunicipio : int(5)
municipio : varchar(100)	municipio : varchar(100)	municipio : varchar(100)	municipio : varchar(100)
demandantes : int(8)	parados : int(8)	contratos : int(8)	total : int(8)
hombremen25 : int(7)	hombremen25 : int(7)	iniindehom : int(7)	contributiva : int(8)
hombre2545 : int(7)	hombre2545 : int(7)	initemphom : int(7)	subsidio : int(7)
hombremay45 : int(7)	hombremay45 : int(7)	conindehom : int(7)	insercion : int(7)
mujermen25 : int(7)	mujermen25 : int(7)	iniindemuj : int(7)	agrario : int(7)
mujer2545 : int(7)	mujer2545 : int(7)	initempmuj : int(7)	
mujermay45 : int(7)	mujermay45 : int(7)	conindemuj : int(7)	
agricultura : int(7)	agricultura : int(7)	agricultura : int(7)	
industria : int(7)	industria : int(7)	industria : int(7)	
construccion : int(7)	construccion : int(7)	construccion : int(7)	
servicios : int(7)	servicios : int(7)	servicios : int(7)	
primerempleo : int(7)	primerempleo : int(7)		

4. Tablas de datos sin procesar de la Base de Datos 'empleo'.

Este proceso de carga y limpieza se realiza una vez al mes, cuando se publican los datos del mes anterior. Primero se ejecuta el script que carga los datos del nuevo mes en las cuatro tablas en crudo y entonces saltan los triggers que cargan las cuatro tablas limpias.

Finalmente se dispone de otras tres tablas con la información de municipios, provincias y comunidades autónomas. Junto con las tablas de parados, prestaciones, contratos y demandantes que vemos en la Figura 5, forman parte de las tablas que almacenan los datos procesados y limpios.



5. Tablas de la Base de Datos 'empleo' con los datos limpios.

Las cuatro tablas *raw* tienen en común los campos que se describen a continuación:

- *fecha*: año y mes numérico. Ejemplo: '201802'
- *mesanyo*: mes y año en texto. Ejemplo 'Febrero de 2018'
- *codigocomunidad*: código numérico de dos cifras que identifica a la comunidad autónoma
- *comunidad*: nombre de la comunidad autónoma
- *codigoprovincia*: código numérico de dos cifras que identifica a la provincia
- *provincia*: nombre de la provincia
- *codigomunicipio*: código de cinco cifras que identifica al municipio
- *municipio*: nombre del municipio

La limpieza de las tablas *raw* consiste en separar la fecha en un campo año y otro mes ambos numéricos y en eliminar los campos *comunidad*, *codigocomunidad*, *provincia*, *codigoprovincia*, y *municipio* ya que con el código del municipio y las tablas de municipios, provincias y comunidades es suficiente para conocer el nombre del municipio, a qué provincia pertenece y ésta a su vez a qué comunidad.

Por lo tanto, los campos que tienen en común las tablas ya limpias son los siguientes:

- *anyo*: año en formato de cuatro dígitos
- *mes*: mes numérico en dos dígitos
- *municipio*: código de cinco dígitos que identifica al municipio

A continuación, se detalla cada tabla y su contenido.

1. **Rawcontratos.** Se utiliza únicamente para almacenar temporalmente la información que proviene del Servicio Público de Empleo Estatal hasta que, mediante un trigger de la tabla, se inserta en la tabla *contratos* la información limpia y se vacía la tabla *raw*.
2. **Contratos.** Alberga la información de los contratos realizados que ofrece el Servicio Público de Empleo Estatal limpia, es decir, se eliminan fechas duplicadas e identificación duplicada de municipios, provincias y comunidades autónomas. Se reducen de 19 a 14 el número de columnas respecto de *rawcontratos* y pasa a ocupar un 35% menos, conteniendo la misma información.

Los campos que tienen ambas tablas que corresponden a un municipio y a un mes determinado son los siguientes:

- *contratos*: número total de contratos realizados
  - *indehom*: número de contratos indefinidos de hombres
  - *temphom*: número de contratos temporales de hombres
  - *convhom*: número de contratos convertidos de temporales a indefinidos de hombres
  - *indemuj*: número de contratos indefinidos de mujeres
  - *tempmuj*: número de contratos temporales de mujeres
  - *convmuj*: número de contratos convertidos de temporales a indefinidos de mujeres
  - *agricultura*: número de contratos realizados en el sector de agricultura
  - *industria*: número de contratos realizados en el sector de industria
  - *construccion*: número de contratos realizados en el sector construcción
  - *servicios*: número de contratos realizados en el sector servicios
3. **Rawdemandantes y Rawparados.** Son tablas en las que se almacenan temporalmente la información de los demandantes de empleo y de los parados tal y como aparece en los ficheros del Servicio Público de Empleo Estatal. Mediante los *triggers after insert* de cada tabla, se inserta la información limpia en sus correspondientes tablas y éstas, se vacían.

La diferencia entre ambas tablas reside en los colectivos a los que hacen referencia: los demandantes son los trabajadores que se inscriben en los servicios públicos de empleo para la búsqueda de un empleo o mejorar el que ya poseen (intermediación laboral), para recibir otros servicios ajenos al empleo (orientación para el autoempleo, formación ocupacional,...) o por la obligatoriedad de inscribirse para percibir una prestación contributiva o un subsidio [35].

Sin embargo, la definición de paro registrado consiste en el número de demandantes de empleo pendientes de satisfacer el último día del mes, es decir, que aún no han obtenido resultado en su búsqueda de empleo.

4. **Demandantes y Parados.** Contienen la información procesada de las tablas *rawdemandantes* y *rawparados*. Se reducen de 20 a 15 el número de columnas

respecto a las tablas en *raw* y pasan a ocupar un 35% menos, conteniendo la misma información y los mismos registros.

Las columnas que tienen estas cuatro tablas (*rawdemandantes*, *rawparados*, *demandantes* y *parados*) en las que cada registro corresponde a un municipio y a un mes determinado son las siguientes:

- *demandantes*: número total de demandantes empleo
- *hombremen25*: número de demandantes de empleo hombres menores de 25 años
- *hombre2545*: número de demandantes de empleo hombres que tienen entre 25 y 45 años
- *hombremay45*: número de demandantes de empleo hombres mayores de 45 años
- *mujermen25*: número de demandantes de empleo mujeres menores de 25 años
- *mujer2545*: número de demandantes de empleo mujeres entre 25 y 45 años
- *mujermay45*: número de demandantes de empleo mujeres mayores de 45 años
- *agricultura*: número de demandantes de empleo que han trabajado en el sector de agricultura
- *industria*: número de demandantes de empleo que han trabajado en el sector de industria
- *construcción*: número de demandantes de empleo que han trabajado en el sector de construcción
- *servicios*: número de demandantes de empleo que han trabajado en el sector servicios
- *primerempleo*: número de demandantes de empleo que nunca han tenido trabajo anteriormente

La volumetría de las tablas *contratos*, *demandantes* y *parados*, como se puede ver en la Figura 4 es de 1.201.568 filas cada una hasta la carga del mes de Abril de 2018 en la que se incluyen los datos desde Enero de 2006 hasta Marzo de 2018. En crecimiento de todas ellas es de 8.126 registros nuevos al mes. El tamaño de las tablas con los datos contenidos anteriormente mencionados, como también se ve en la Figura 4, es el siguiente:

- *contratos*: 153,8 MB
- *demandantes*: 158,9 MB
- *parados*: 161,8 MB

5. **Rawprestaciones.** Esta tabla contiene los datos del gasto en prestaciones por provincias y meses desde 2010. Al igual que en los casos anteriores, esta tabla se utiliza para la limpieza de los datos. En ella se cargan los registros del CSV del Servicio Público de Empleo Estatal mientras el trigger prestaciones introduce los datos limpios en la tabla correspondiente.
6. **Prestaciones.** Alberga los datos limpios de gasto en prestaciones segregado en varios tipos de ayudas.

Las columnas que contienen estas dos tablas son las siguientes:

- *contributiva*: Gasto en prestación contributiva. Es la ayuda que se percibe tras la pérdida involuntaria de un empleo.
- *subsidio*: Gasto en subsidio de desempleo. Es la ayuda que corresponde a las personas que han agotado su prestación contributiva y cumplen ciertos requisitos.
- *insercion*: Gasto en renta activa de inserción. Es una ayuda que ciertos colectivos perciben (como por ejemplo desempleados de larga duración) cuando se han agotado las dos anteriores.
- *agrario*: Gasto en subsidio eventual agrario. Es una prestación por desempleo de las personas trabajadoras eventuales del Sistema Especial para Trabajadores por Cuenta Ajena Agrarios de la Seguridad Social pertenecientes a las comunidades autónomas de Extremadura y Andalucía.

7. **Comunidades.** Catálogo de las comunidades autónomas del país. Tras la carga inicial en la que se insertan datos no se vuelve a actualizar. Contiene dos columnas: '*codigo*' que es un entero de dos cifras y es su clave primaria y '*nombre*' que es una cadena de tipo *varchar* de 25 caracteres. Tiene una volumetría de 19 debido a la suma de las 17 comunidades autónomas y las 2 ciudades autónomas. Ocupa 16 KB.
8. **Provincias.** Registro de todas las provincias. Solo recibe datos en la carga inicial y no se refrescan sus datos más adelante. Contiene tres campos: '*código*', un entero de dos cifras y la clave primaria de la tabla, '*nombre*' que es una cadena *varchar* y '*comunidad*' que es una clave foránea de la clave de *comunidades* y vincula a cada provincia con la comunidad a la que pertenece. Su volumetría es de 52 registros y requiere 16 KB en disco.
9. **Municipios.** Tabla con todos los municipios existentes. Al igual que con *comunidades* y *provincias* solo se cargan datos una vez y desde entonces no es necesario que sea actualizada. Dispone de tres columnas que son '*codigo*' que es un entero de 5 cifras y la clave primaria, '*nombre*' que es una cadena *varchar* de 100 caracteres y '*provincia*' que hace referencia a la clave de la tabla de provincias. Tiene 8.126 registros y ocupa 368 KB.

Tabla	Filas	Tipo	Cotejamiento	Tamaño
comunidades	19	InnoDB	latin1_spanish_ci	16 KB
contratos	1,201,568	InnoDB	latin1_spanish_ci	153.8 MB
demandantes	1,201,568	InnoDB	latin1_spanish_ci	158.9 MB
municipios	8,126	InnoDB	latin1_spanish_ci	368 KB
parados	1,201,568	InnoDB	utf8_spanish_ci	161.8 MB
prestaciones	5,148	InnoDB	utf8_spanish_ci	16 KB
provincias	52	InnoDB	latin1_spanish_ci	16 KB

6. Información sobre las tablas de la base de datos 'empleo'.

## ii. Base de datos 'login'

La segunda de las bases de datos relacionales es la que almacena los datos de los usuarios llamada **login**. Esta base de datos, como se ve en la Figura 7, tiene una tabla.

1. **Users**. Alberga toda la información para el registro y acceso de los usuarios a la aplicación web. Tiene ocho columnas:

- *user\_id*: Identificador del usuario.
- *user\_name*: Nombre del usuario.
- *user\_password\_hash*: Contraseña del usuario codificada.
- *user\_email*: Correo electrónico del usuario.
- *user\_category*: Categoría laboral a la que se dedica el usuario.
- *user\_province*: Provincia donde reside el usuario.
- *user\_experience*: Años de experiencia que tiene el usuario.
- *move\_availability*: Disponibilidad (booleano) que tiene el usuario para trabajar en otra ciudad.

Recibe consultas cuando los usuarios entran al sistema e *inserts* cuando se registran. Su volumetría y tamaño dependen linealmente de los usuarios que haya registrados.



Column	Type
user_id	int(11)
user_name	varchar(64)
user_password_hash	varchar(255)
user_email	varchar(64)
user_category	enum(...)
user_province	enum(...)
user_experience	int(1)
move_availability	tinyint(1)

7. Base de Datos 'login'.

## II. Modelo orientado a documentos

Los datos que se reciben a través de la API de InfoJobs, tienen estructura de JSON por lo que lo óptimo sería insertarlas en una base de datos orientada a documentos. Además, tanto la gran cantidad de datos insertados como la alta periodicidad con la que se acceden e insertan nuevos datos requiere un sistema gestor de bases de datos rápido, consistente y escalable. Haciendo un análisis de los sistemas de bases de datos existentes, finalmente se decide que MongoDB es el más apropiado para el sistema.

Dentro de la base de datos 'db' de MongoDB hay una colección llamada 'ofertas' donde se introducen todas las ofertas que se publican en la web de InfoJobs.net gracias a un script que se ejecuta periódicamente en el servidor. Se hace coincidir el campo '*id*' de la oferta devuelta por la API con el '*\_id*' que es el identificador que utiliza MongoDB para indexar los documentos. De esta forma, se evita que se introduzca varias veces en la colección ofertas repetidas, ya que el campo '*id*' si podría estar repetido en la colección, pero en ningún caso el campo '*\_id*'.

Por otro lado, a pesar de que MongoDB no requiere que los documentos que se inserten tengan un esquema definido, en este caso las ofertas sí lo van a tener, lo cual simplifica las consultas. El esquema de una oferta, como se puede observar en la Figura 8, contiene veintiún campos, de los cuales, algunos de ellos contienen varios subcampos.

Key	Value	Type
(1) dfc5a497ae4b8f9ecc2370f62e9012	{ 21 fields }	Object
_id	dfc5a497ae4b8f9ecc2370f62e9012	String
title	Programador PHP Senior - Ecommerce - Magento	String
province	{ 2 fields }	Object
id	33	Int32
value	Madrid	String
city	Madrid (Madrid)	String
link	https://www.infojobs.net/madrid-madrid/programador-php-senior-ecommerce-magento/...	String
category	{ 2 fields }	Object
id	150	Int32
value	Informática y telecomunicaciones	String
contractType	{ 2 fields }	Object
id	1	Int32
value	Indefinido	String
subcategory	{ 2 fields }	Object
id	3124	Int32
value	Programación	String
salaryMin	{ 2 fields }	Object
id	290	Int32
value	36.000 €	String
salaryMax	{ 2 fields }	Object
id	290	Int32
value	36.000 €	String
salaryPeriod	{ 2 fields }	Object
id	3	Int32
value	Bruto/año	String
experienceMin	{ 2 fields }	Object
id	6	Int32
value	Al menos 2 años	String
workDay	{ 2 fields }	Object
id	1	Int32
value	Completa	String
study	{ 2 fields }	Object
id	35	Int32
value	Ciclo Formativo Grado Medio	String
published	2018-04-15T17:52:40.000Z	String
updated	2018-05-27T20:30:00.000Z	String
author	{ 3 fields }	Object
id	98565452575049101671110014134800310126	String
name	ECOMMBITS	String
uri	http://www.infojobs.net/api.infojobs.net/ecommbits/em-i9856545257504910167111001...	String
requirementMin	Experiencia demostrable de más de 3 años en desarrollo de proyectos ecommerce con ...	Boolean
bold	false	Boolean
urgent	false	Boolean
applications	12	String

8. Documento de ejemplo de la colección 'ofertas'.

La volumetría de esta colección, como se ve en la Figura 9, es de 94.219 ofertas. Cada día se insertan unas 2000 nuevas ofertas (depende de las ofertas que hayan sido dadas de alta) y aproximadamente un número similar se dan de baja en el sistema. Además el tamaño, como se puede observar en la Figura 10, es de 120.845.549 Bytes, lo que equivale a 115,25 MB.

```
> db.ofertas.find().count();
94219
```

9. Volumetría de la colección 'ofertas'.

```
> db.ofertas.dataSize();
120845549
```

10. Tamaño de la colección 'ofertas'.

El periodo de actualización de las ofertas se regula en función de las ofertas insertadas ya que no se dan de alta el mismo número de ellas un día de diario a las 13 horas de la tarde que un fin de semana a las 3 horas de la madrugada. Es por esto que el script de recopilación se ejecuta entre una vez cada minuto (como máximo) a una vez cada hora (como mínimo).

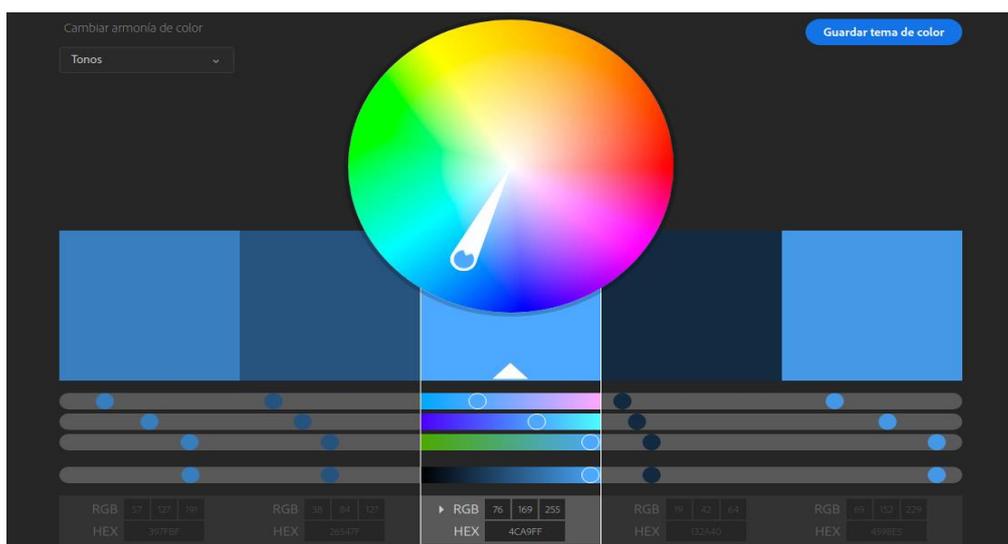
Como se ha indicado anteriormente, un gran número de ofertas se dan de baja en el sistema cada día porque ya han sido cubiertas sus vacantes o bien porque ya tienen un número de candidatos que la entidad considera suficientes. Para que esto no afecte a la fiabilidad de los datos (si una empresa publica y elimina continuamente una oferta para un puesto concreto, en el sistema se tendría una demanda inflada de ese tipo de empleo ya que no se corresponde con la realidad) se debe eliminar estas ofertas que ya no están publicadas en InfoJobs.net [36].

Para llevar a cabo este proceso, hay dos opciones: la primera es consultar si cada una de las ofertas continúa estando activa en InfoJobs.net y si no, borrarla. Este proceso es lento ya que su coste es elevado, por lo que se elige la otra opción: vaciar la base de datos y volver a cargarla de nuevo. Este procedimiento se ejecuta una vez cada día.

## 6. Diseño de la aplicación

En este apartado se expone el diseño de la aplicación web. Se trata de un diseño, que en su versión de escritorio está organizado visualmente en dos paneles principales, la barra lateral y el panel principal. El cuadro con los botones de inicio de sesión y de registro aparece desplegado cuando pulsamos el botón en la parte superior derecha en cualquiera de las páginas de la aplicación, que nos llevará a una página específica para estas acciones.

Ha sido utilizada una paleta de colores azules que podemos ver en la Figura 11, con la ayuda de Color Wheel [37], una herramienta de Adobe en la que se combinan colores oscuros sobre fondo claro para facilitar la visualización de la aplicación web.



11. Rueda de colores de Adobe. Rango de tonos elegido en la aplicación web.

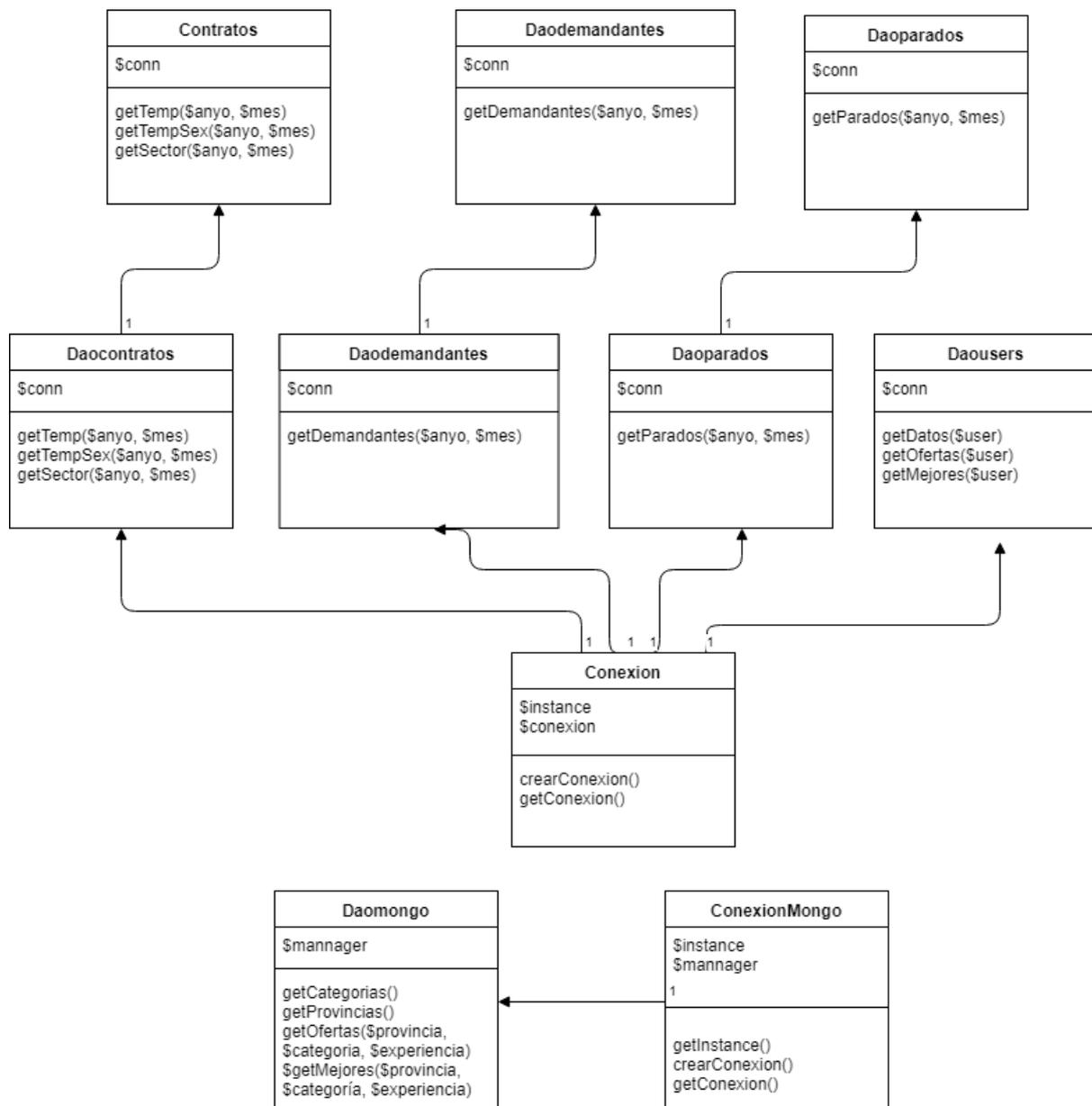
Han sido desarrolladas dos versiones: una para escritorio y otra para dispositivos móviles mediante la utilización del framework Bootstrap 4. Este framework hace posible que los contenedores se adapten al tamaño de la pantalla del dispositivo desde el que se visualiza el contenido.

En la versión de escritorio aparecen en la pantalla los paneles del contenido, el menú lateral y el botón que despliega el menú de inicio de sesión y registro o de opciones de cuenta y cerrar sesión (en el caso de ya haber iniciado sesión). En el panel de contenido, los paneles se agrupan en filas de dos columnas (por ejemplo, en la vista de ofertas recomendadas) mientras que en la versión móvil aparecen en una única columna.

En el diseño de la versión móvil, solo vemos un panel principal adaptado al tamaño y resolución de la pantalla de los dispositivos móviles. Aparece un botón de menú en la parte superior derecha de la pantalla el que, si es pulsado, despliega un menú desde la parte superior ocupando toda la pantalla y que desaparece cuando pulsamos en una de las opciones. El resto de funciones son idénticas a la versión de escritorio, salvo que la apariencia cambia, apareciendo menos paneles por cada fila.

## 7. Implementación

Para la implementación de esta aplicación se ha utilizado el patrón DAO para realizar las conexiones a las Bases de Datos, tanto de MongoDB como de MariaDB. En primer lugar, como se ve en las Figura 13 y 14, se han implementado las clases ConexionMongo y ConexionMariadb además de una clase DAO para contratos (Figura 15), otra para demandantes, otra para parados y otra para ofertas.



12. Diagrama de clases.

```

function crearConexion() {
    try {
        $this->manager = new MongoDB\Driver\Manager("mongodb://localhost:27017");
    }
    catch(PDOException $e)
    {
        echo "Error en la conexión con la base de datos Mongo: " . $e->getMessage();
    }
    return $this->manager;
}

```

13. Establecimiento de una conexión con MongoDB.

```

static function crearConexion() {
    $host = DB_HOST;
    $user = DB_USER;
    $password = DB_PASS;
    $dbname = DB_NAME_EMPLEO;
    try {
        $con = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8", $user, $password);
        $con->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    }
    catch(PDOException $e)
    {
        echo "Error en la conexión con la base de datos: " . $e->getMessage();
    }
    return $con;
}

```

14. Establecimiento de una conexión con MariaDB.

```

function getTemp($anyo, $mes) {
    try {
        $sql = "SELECT SUM(indehom)+SUM(convhom)+SUM(indemuj)+SUM(convmuj) AS 'Indefinidos', SUM(
            temphom)+SUM(tempmuj) AS 'Temporales' FROM contratos WHERE anyo = '$anyo' AND mes = '$mes'";
        $stmt = $this->conn->prepare($sql);
        $stmt->execute();
        $res = $stmt->fetch();
    } catch(PDOException $e) {
        echo "Error en Daocontratos (temporalidad): " . $e->getMessage();
    }

    return $res;
}

```

15. Una de las funciones de la clase Daocontratos.

Por otro lado, se implementan clases por cada una de las entidades que se emplean en el sistema como por ejemplo, como se ve en la Figura 16 la clase Contratos. Todas estas clases son las que se ejecutan para finalmente obtener la vista de “Tipos de contrato por temporalidad” como vemos en la Figura 17.

```

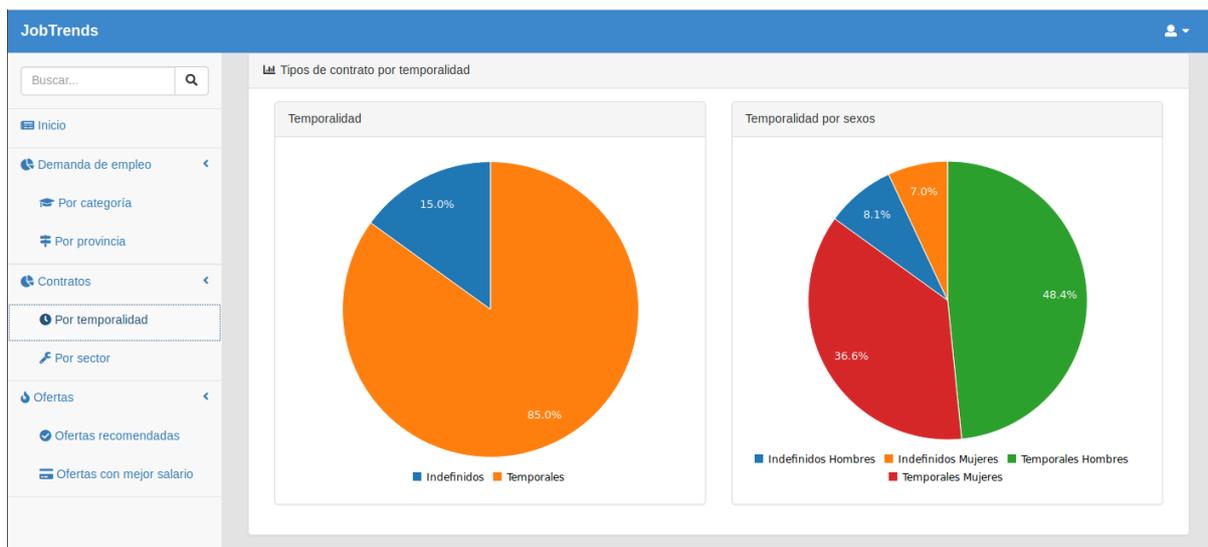
Class Contratos{
    private $daoContratos;

    function Contratos() {
        $this->daoContratos = new Daocontratos();
    }

    function getContratos($anyo, $mes){
        $cont = $this->daoContratos->getTemp($anyo, $mes);
        return $cont;
    }
}

```

16. Parte de la clase Contratos.



17. Vista de datos sobre la temporalidad de los contratos.

Otros ficheros de la aplicación web a destacar son las vistas, como por ejemplo, la que se puede observar en las Figuras 18 y 19, de la página de inicio de sesión o los scripts utilizados para generar las gráficas con la librería C3.js como se ve en la Figura 20.

```

<div class="panel-heading">
  <h3 class="panel-title">Iniciar sesión</h3>
</div>
<div class="panel-body">
  <form method="post" action="login.php" name="loginform">
    <fieldset>
      <div class="form-group">
        <input id="login_input_username" placeholder="Usuario" class="
          login_input form-control" type="text" name="user_name" required autofocus
        >
      </div>
      <div class="form-group">
        <input id="login_input_password" class="login_input form-control" type="
          password" name="user_password" autocomplete="off" placeholder="Contraseña
          " required>
      </div>
      <input type="submit" class="btn btn-lg btn-success btn-block" name="login"
        value="Iniciar sesión" />
      <br>
    </fieldset>
  </form>
</div>

```

18. Paneles de la página de inicio de sesión.

The screenshot shows a login form titled "Iniciar sesión". It contains two input fields: "Usuario" and "Contraseña". Below the fields is a green button labeled "Iniciar sesión". The form is centered on a light gray background.

19. Página de inicio de sesión.

```

<script type="text/javascript">
    var chart = c3.generate({
        data: {
            columns: [
                ['Indefinidos', <?php echo $temporalidad[0]?>],
                ['Temporales', <?php echo $temporalidad[1]?>]
            ],
            type : 'pie'
        }
    });

    var chart2 = c3.generate({
        data: {
            columns: [
                ['Indefinidos Hombres', <?php echo $sexo[0]?>],
                ['Indefinidos Mujeres', <?php echo $sexo[1]?>],
                ['Temporales Hombres', <?php echo $sexo[2]?>],
                ['Temporales Mujeres', <?php echo $sexo[3]?>]
            ],
            type : 'pie'
        }, bindto: '#chart1'
    });
</script>

```

20. Script para generar el gráfico de contratos dados de alta por sexos.

La funcionalidad de recolectar ofertas del portal InfoJobs.net está implementada en un único script para que sea ejecutado periódicamente por el servidor. En este script, como se observa en la Figura 21, se conecta con la API de InfoJobs mediante autenticación OAuth 2.0 [11], se crea la conexión con la base de datos de MongoDB, se insertan las ofertas en la colección como se muestra en la Figura 22, se implementan las excepciones y se crea un log para almacenar la información sobre cuántas ofertas han sido insertadas en cada ejecución.

```

//Se crea el manager con la conexión al servidor mongodb
$manager = new MongoDB\Driver\Manager($inst.$host);
//Se crea un fichero de logs en el mismo directorio
$fichero = fopen($ruta.$log, "a");
//Se crea la url de petición a la API
$suri = "https://api.infojobs.net/api/1/offer?maxResults=".$results."&page=".$page;
//Se solicita la petición
try {
    $response = \Httpful\Request::get($suri)
        ->authenticateWith('correo@gmail.com', 'contraseña')
        ->addHeader('Authorization', "Basic YmJkODRkM2ExYjk4NDdlMGJlN2I4Mzg4OWQ0WY2Zj")
        ->send();

    //Se crea el bulk para insertar datos
    $bulk = new MongoDB\Driver\BulkWrite(['ordered' => false]);
    $oferts = $response->body->currentResults;
    if($results > $oferts){
        echo "Se han devuelto solo $oferts ofertas de las $results solicitadas.\n";
    }
}
//Se guarda en un cada oferta un nuevo elemento bulk para que maneje bien el id de

```

21. Utilización de la API de InfoJobs.

```

    $repes = count($e->getWriteResult()->getWriteErrors());
    $num = $oferts-$repes;
    $s .= "Ofertas insertadas: ". $num. ", ";
    $s .= "Ofertas repetidas: ". $repes. "\n";
    fwrite($fichero, $s);
    fclose($fichero);
}
catch (MongoDB\Driver\Exception\ConnectionTimeoutException $e) {
    echo "No está levantado el servidor MongoDB en $host\n";
}
catch (MongoDB\Driver\Exception\InvalidArgumentException $e) {
    echo "Se ha accedido a posiciones vacías del array\n";
}
echo $s."\n";
}
catch (Httpful\Exception\ConnectionErrorException $e) {
    echo "No hay conexión a internet o el servicio está caído\n";
}
}

```

22. Inserción de las ofertas y tratamiento de las excepciones.

Por último, como se ve en la Figura 23, se ha desarrollado un script en Bash para el despliegue de la aplicación en el servidor que consiste en la instalación de los paquetes necesarios, establecimiento de la dirección ip estática, levantamiento de los servicios de bases de datos y carga de éstos.

```

75 sudo /etc/init.d/ssh start >/dev/null
76 #Log
77 Hora=`date +"%Y-%m-%d/%T"`
78 echo "$Hora: Se instala servidor y cliente de mariadb" >> $REGISTRO;
79 echo "$Hora: Instalando servidor y cliente de mariadb...";
80 #/Log
81 sudo apt-get -y install mariadb-server mariadb-client >/dev/null
82 #Log
83 mysql_secure_installation
84 Hora=`date +"%Y-%m-%d/%T"`
85 echo "$Hora: Se instala apache2" >> $REGISTRO;
86 echo "$Hora: Instalando apache2...";
87 #/Log
88 sudo apt-get -y install apache2 >/dev/null
89 #Log
90 Hora=`date +"%Y-%m-%d/%T"`
91 echo "$Hora: Se instala php7" >> $REGISTRO;
92 echo "$Hora: Instalando php7...";
93 #/Log
94 sudo apt-get -y install php7.0 libapache2-mod-php7.0 >/dev/null
95 #Log
96 Hora=`date +"%Y-%m-%d/%T"`
97 echo "$Hora: Se reinicia apache" >> $REGISTRO;
98 echo "$Hora: Reiniciando apache...";
99 #/Log
100 sudo service apache2 restart
101 sudo service apache2 reload
102 #Log
103 Hora=`date +"%Y-%m-%d/%T"`
104 echo "$Hora: Se instalan paquetes de php" >> $REGISTRO;

```

23. Script de despliegue del servidor.

## 8. Conclusiones, aportación personal y trabajo futuro

### I. Conclusiones

Se ha desarrollado una aplicación web que permite acceder a información sobre el mercado laboral. La aplicación facilita a los usuarios la búsqueda de empleo y la orientación laboral utilizando datos reales y actuales siendo expuestos en una interfaz amigable y adaptada diferentes tipos de dispositivos.

### II. Aportación personal

A lo largo del desarrollo de este proyecto he mejorado mis habilidades sobre planificación ya que un proyecto de estas características lo requieren. Por otro lado, he aprendido acerca del desarrollo de aplicaciones web y utilización de APIs además de afianzar mis conocimientos en sistemas de gestión de Bases de Datos tanto relacionales como documentales y despliegue y utilización de servidores.

Debido al carácter individual de este proyecto, mi aportación personal a él ha sido íntegra con sus ventajas e inconvenientes. Entre las ventajas destaco la oportunidad de conocer al 100% un proyecto. Por otro lado, el mayor inconveniente es la dificultad para desbloquear situaciones en las que con una sola visión es complejo desatascar como por ejemplo cuestiones técnicas.

### III. Trabajo futuro

La aplicación desarrollada presenta algunas limitaciones tales como la seguridad, debido a que no se comprueba la veracidad del correo electrónico de los usuarios registrados o a que no se cifran las comunicaciones con el servidor.

En este sentido, se plantean como líneas de trabajo futuro las siguientes:

1. Añadir un sistema de verificación creando un servidor de correo para confirmar el email y utilizando la API de *Twilio* [38] para confirmar el número de teléfono mediante el envío de un sms para evitar que los usuarios puedan registrarse en el sistema con estos datos falsos.

2. Sería conveniente añadir el protocolo SSL/TLS [39] que combinado con HTTP [40] nos permite securizar la aplicación web mediante cifrado de las comunicaciones para evitar ataques como el espionaje.
3. Se podría incluir un sistema de etiquetas sobre los *skills* o cualidades más demandados extrayendo los términos relevantes o '*topics*' de la descripción y requisitos de las ofertas con la API *MeaningCloud* [41].
4. Crear un sistema de alertas con nuevas ofertas personalizadas que lleguen al usuario en el momento de su publicación a través de la posibilidad de las notificaciones push en los navegadores modernos.
5. Añadir la posibilidad de crear informes descargables en *PDF* con los datos de la aplicación web y establecer un boletín opcional enviado periódicamente a través del correo electrónico para los suscriptores. Además, este boletín podría ser adaptado al usuario.

# Bibliografía

- [1] <http://www.elmundo.es/economia/2018/01/02/5a4b8667268e3e9b338b459c.html>
- [2] <https://resources-01.adecogroup.com/~media/adecogroup/brands/spring-professional/spain/media/home/notas-de-prensa/ndpxiiiinformelosbuscadosdespringprofessional2018grupoadeccooptimizado.pdf?la=es-es>
- [3] <https://www.sage.com/es-es/blog/perfiles-laborales-mas-demandados/>
- [4] <http://www.lavanguardia.com/economia/innovacion/20180222/44988094660/profesionales-demandados-ingenieria-industrial-tecnocampus.html>
- [5] <https://www.trabajando.es/contenido/noticia/1515583767/Ingenieria-Informatica-continua-siendo-la-carrera-mas-demandada-por-el-mercado-laboral.html>
- [6] <https://www.educacion.gob.es/notasdecorte/compBdDo>
- [7] <http://www.elmundo.es/especiales/educacion/empleo-universidad.html>
- [8] <http://php.net/manual/es/intro-what-is.php>
- [9] <http://phphttpclient.com/#h>
- [10] <https://developer.infojobs.net/>
- [11] <https://oauth.net/2/>
- [12] <https://www.mongodb.com/what-is-mongodb>
- [13] [https://www.w3schools.com/js/js\\_json\\_intro.asp](https://www.w3schools.com/js/js_json_intro.asp)
- [14] <https://robomongo.org/>
- [15] <https://www.getpostman.com/>
- [16] [https://es.wikipedia.org/wiki/Valores\\_separados\\_por\\_comas](https://es.wikipedia.org/wiki/Valores_separados_por_comas)
- [17] <https://es.wikipedia.org/wiki/SQL>
- [18] <https://mariadb.org/>
- [19] [https://es.wikipedia.org/wiki/Trigger\\_\(base\\_de\\_datos\)](https://es.wikipedia.org/wiki/Trigger_(base_de_datos))

- [20] <https://www.phpmyadmin.net/>
- [21] [https://www.w3schools.com/html/html5\\_intro.asp](https://www.w3schools.com/html/html5_intro.asp)
- [22] <https://www.w3schools.com/css/>
- [23] <https://www.w3schools.com/jS/default.asp>
- [24] <https://getbootstrap.com/docs/4.1/getting-started/introduction/>
- [25] <http://jquery.com/>
- [26] <http://c3js.org/>
- [27] <https://www.ubuntu.com/download/server>
- [28] <https://es.wikipedia.org/wiki/Bash>
- [29] [https://es.wikipedia.org/wiki/Cron\\_\(Unix\)](https://es.wikipedia.org/wiki/Cron_(Unix))
- [30] <https://www.virtualbox.org>
- [31] <https://es.wikipedia.org/wiki/Cliente-servidor>
- [32] <https://es.wikipedia.org/wiki/Modelo%20%80%93vista%20%80%93controlador>
- [33] <https://www.sepe.es/>
- [34] <http://www.empleo.gob.es/>
- [35] [http://www.sepe.es/contenidos/que\\_es\\_el\\_sepe/estadisticas/datos\\_estadisticos/empleo/datos/2018/marzo\\_2018/intro.pdf](http://www.sepe.es/contenidos/que_es_el_sepe/estadisticas/datos_estadisticos/empleo/datos/2018/marzo_2018/intro.pdf)
- [36] <https://www.infojobs.net/>
- [37] <https://color.adobe.com/es/create/color-wheel/>
- [38] <https://www.twilio.com/docs/api>
- [39] [https://es.wikipedia.org/wiki/Transport\\_Layer\\_Security](https://es.wikipedia.org/wiki/Transport_Layer_Security)
- [40] [https://es.wikipedia.org/wiki/Protocolo\\_de\\_transferencia\\_de\\_hipertexto](https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto)
- [41] <https://www.meaningcloud.com/developer/topics-extraction/doc/2.0>

# Anexos

## I. Manual de instalación

Para facilitar la instalación del servidor, se suministra una máquina virtual de VirtualBox para que, al ser ejecutada, ya tengamos el servidor levantado y podamos acceder a la aplicación web.

No obstante, si no disponemos de la máquina virtual o bien queremos llevar a cabo todo el proceso de despliegue, se detallan a continuación todos los pasos necesarios.

Se parte de una máquina (virtual o real) con Ubuntu Server 16.04 instalado y con conexión a internet para la descarga de paquetes necesarios.

```
Ubuntu 16.04.4 LTS ubuntu tty1
ubuntu login: user
Password:
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-116-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Pueden actualizarse 105 paquetes.
55 actualizaciones son de seguridad.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

user@ubuntu:~$
```

24. Página de inicio de Ubuntu Server 16.04.

## 1. Instalación de paquetes necesarios

### 1. Servidor y el cliente de Bases de Datos MariaDB:

```
user@ubuntu:~$ sudo apt-get -y install mariadb-server mariadb-client
```

*25. Instalación de MariaDB en línea de comandos.*

Durante la instalación, nos pedirá algunos datos como la contraseña del usuario root.

### 2. Servidor Apache en su versión 2:

```
user@ubuntu:~$ sudo apt-get -y install apache2
```

*26. Instalación de Apache2 en línea de comandos.*

### 3. Intérprete de php y librería de Apache2:

```
user@ubuntu:~$ sudo apt-get -y install php7.0 libapache2-mod-php7.0
```

*27. Instalación de php 7 en línea de comandos.*

### 4. Paquetes adicionales de php:

```
user@ubuntu:~$ sudo apt-get -y install php7.0-mysql php7.0-curl php7.0-gd php7.0-intl php-pear php-imagick php7.0-imap php7.0-mcrypt php-memcache php7.0-pspell php7.0-recode php7.0-sqlite3 php7.0-tidy php7.0-xmlrpc php7.0-xsl php7.0-mbstring php-gettext php7.0-dev php7.0-opcache php-apcu
```

*28. Instalación de paquetes adicionales en línea de comandos.*

### 5. Servidor de Bases de Datos MongoDB:

```
user@ubuntu:~$ sudo apt-get install mongodb
```

*29. Instalación de MongoDB en línea de comandos.*

### 6. Librería de MongoDB para php:

```
user@ubuntu:~$ sudo apt-get -y install php-mongodb
```

```
user@ubuntu:~$ sudo pecl install mongodb
```

*30. Instalación de paquetes de MongoDB para php.*

## 2. Despliegue de la aplicación

1. Se copian en el servidor los ficheros necesarios:
  - a. Carpeta 'jobtrends' con la aplicación web al directorio '/var/www/html/'.
  - b. Scripts de creación e inserción de datos.
2. Se crea el usuario 'userbd' con permisos en mysql:
  - a. Se accede a mysql con el usuario 'root':

```
user@ubuntu:~$ sudo mysql -u root -p
Enter password: █
```

*31. Acceso a mysql desde la línea de comandos.*

- b. Se crea un usuario y se le asignan permisos:

```
MariaDB [mysql]> CREATE USER 'userbd'@'localhost' IDENTIFIED BY
MariaDB [mysql]> GRANT ALL PRIVILEGES ON *.* TO 'userbd'@'localhost';
```

*32. Creación de usuario de MariaDB en línea de comandos.*

3. Se ejecuta el script de creación de las dos Bases de Datos, de todas sus tablas y de población de dichas tablas:

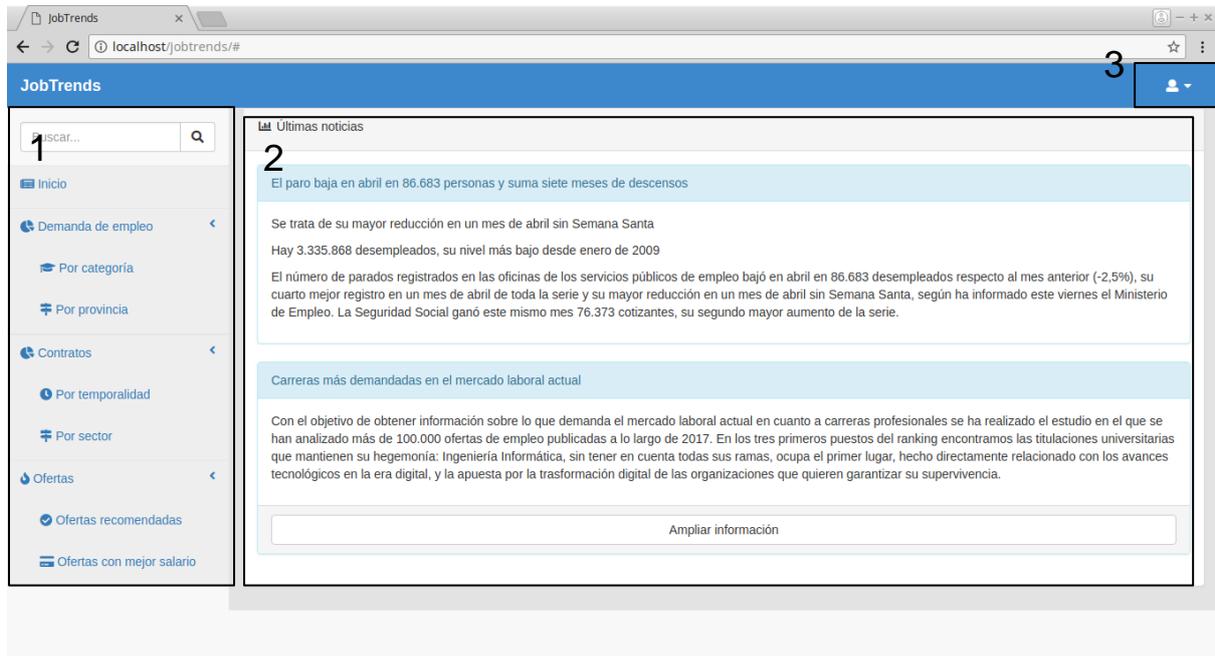
```
MariaDB [mysql]> source script.sql;█
```

*33. Carga de script sobre MariaDB en línea de comandos.*

4. Se accede a la aplicación web desde el navegador introduciendo la dirección 'ip\_servidor/jobtrends/' siempre que el servidor esté conectado en la misma red.

## II. Manual de usuario

La aplicación web se compone de tres paneles como se observa en la Figura 34.



34. Paneles de la página principal de la aplicación web.

El panel 1 es un menú lateral donde aparecen todas las opciones que se pueden realizar en la aplicación web. El panel 2 es el panel principal donde aparece la información. El panel 3 es un botón donde al pulsarlo, se despliega otro panel que permite iniciar sesión o registrarse en el sistema.

### 1. Menú lateral

En este panel disponemos de toda la funcionalidad que ofrece la aplicación web. Pulsando en cada una de las categorías de la lista se puede observar cómo cambia la información en el panel principal. Las características que encontramos son las siguientes:



35. Menú de la aplicación web.

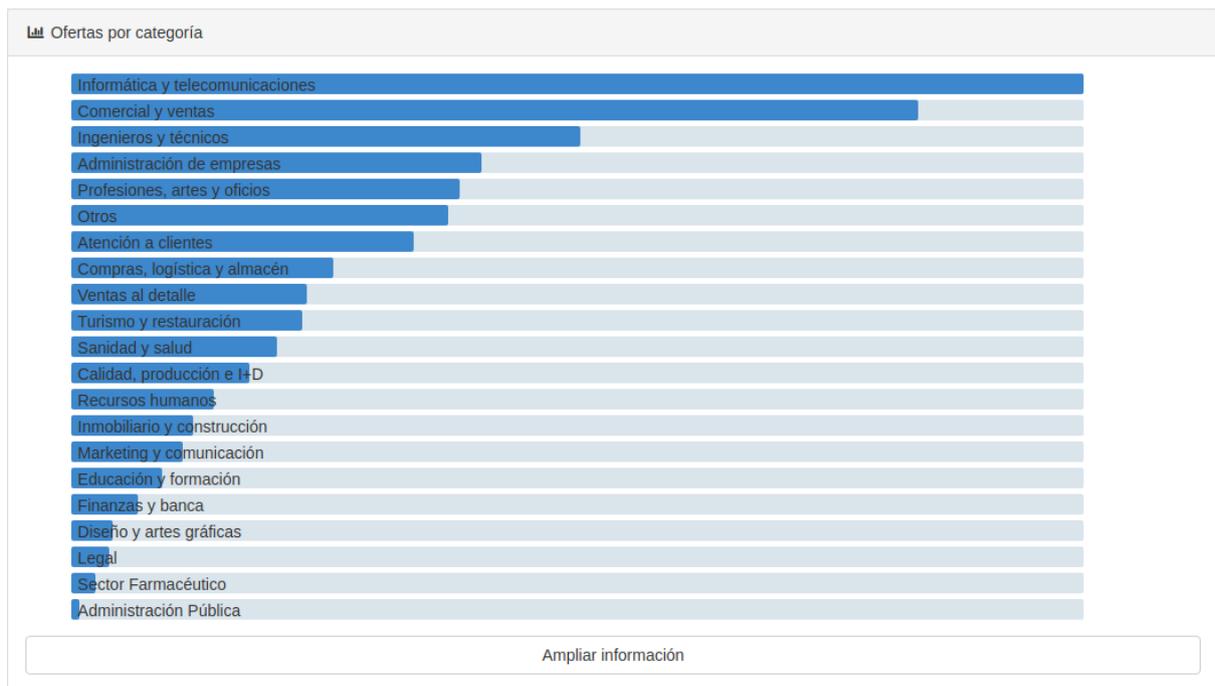
En inicio encontramos noticias. En el apartado de demanda de empleo se encuentran dos botones, el primero, 'por categoría', nos muestra la demanda de empleo que hay en cada categoría en base a las ofertas que son publicadas en el portal de InfoJobs.net. Desde el segundo botón se podrá acceder a un listado de cuántas ofertas han sido publicadas en cada provincia, en orden descendente.

En el apartado de contratos podemos obtener información en tiempo real sobre el porcentaje de contratos de los que se crean son indefinidos o temporales en el apartado 'por temporalidad'. En el segundo apartado se obtienen los datos de los contratos que han sido creados por cada sector.

Finalmente, los usuarios registrados pueden obtener información vinculada a los datos que aportaron en su registro en el sistema. Por un lado, pueden ver las ofertas recomendadas según sus características. Por otro lado, pueden ver cuáles son las ofertas con mayor salario.

## 2. Panel principal

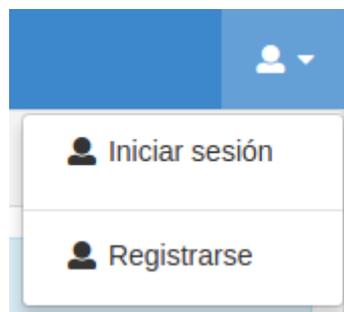
En el panel principal como se muestra en la Figura 36, se ve la información relacionada con el apartado en el que hemos accedido. En este caso se muestra la demanda de las categorías por orden ascendente.



36. Panel con la información de la demanda por categorías.

### 3. Gestión de usuarios

Finalmente, el sistema de usuarios permite registrarse, iniciar sesión y cerrar sesión. En la figura se muestra el panel en el caso en el que no se haya iniciado sesión. Es en esta ocasión en la que se puede tanto iniciar sesión si no se dispone de cuenta pulsando el primer botón, como registrarse en caso contrario, pulsando el segundo botón.



37. Panel de opciones de sesión.

Una vez se accede a alguna de las dos opciones, aparece un formulario con los datos a ingresar por el usuario y un botón para confirmar la acción de iniciar sesión o de registrarse como se puede ver en la Figura 38.

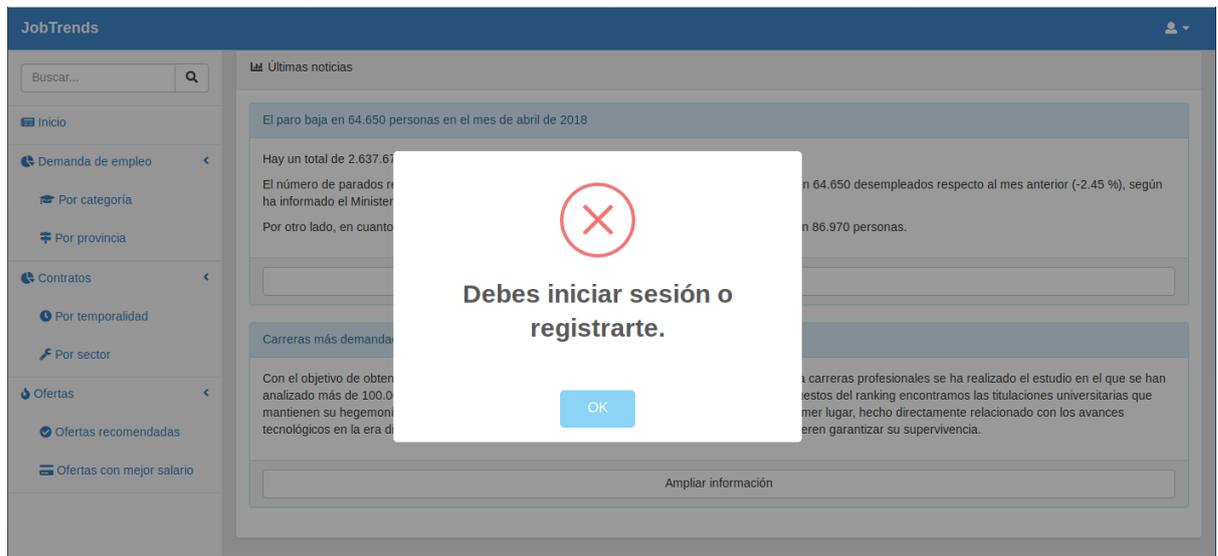
The image shows a registration form titled "Registrarse". It contains the following fields from top to bottom: a text input for "Nombre de usuario", a text input for "Correo electrónico", a text input for "Contraseña", a text input for "Repita la contraseña", a dropdown menu currently showing "Informática y telecomunicaciones", another dropdown menu currently showing "Álava/Araba", and a text input for "Experiencia". At the bottom of the form is a large green button with the text "Registrarse" in white.

38. Formulario de inicio de sesión.

#### 4. Funcionalidad de usuario no registrado

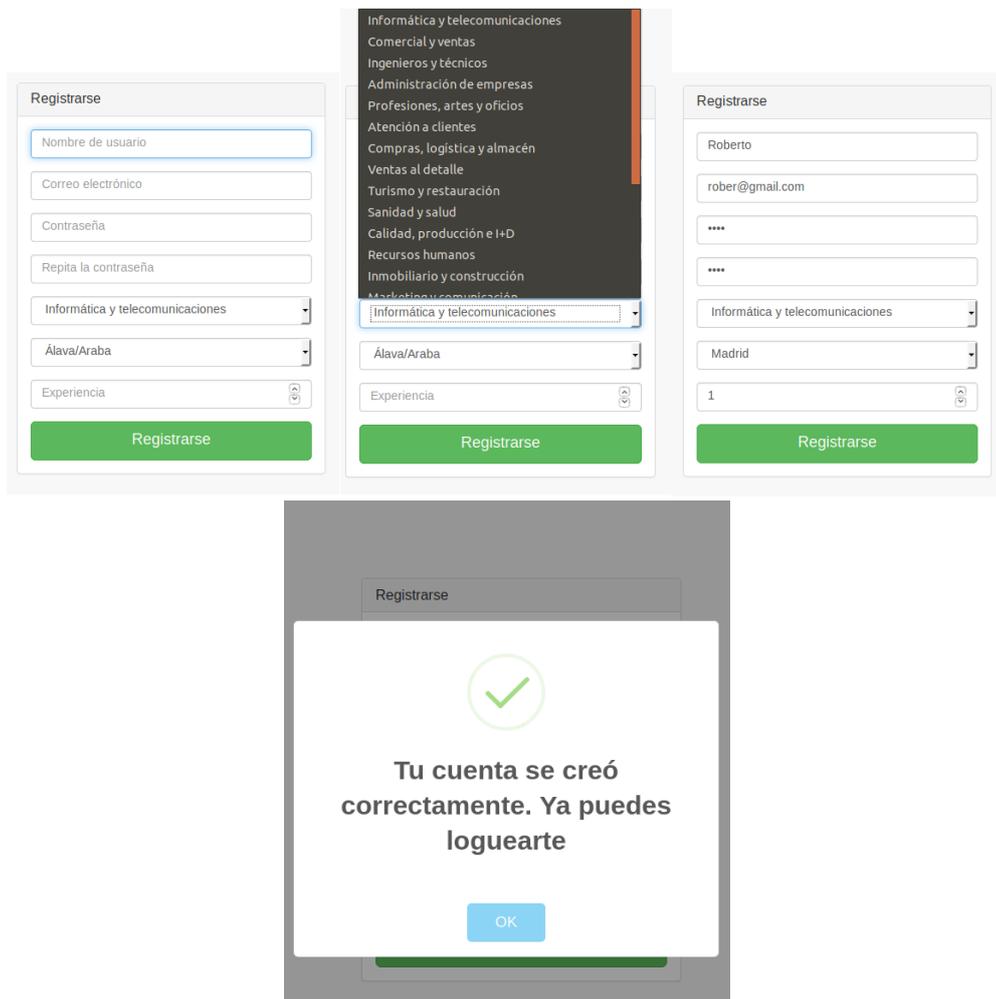
Una vez que se accede a la web, se pueden ver las páginas de inicio (noticias), demanda de empleo por categoría, demanda de empleo por provincia, contratos por temporalidad y contratos por sector.

Si se intenta acceder a la sección de ofertas recomendadas u ofertas con mejor salario, aparecerá en la ventana el mensaje que vemos en la Figura 39. que avisa de que es necesario estar registrado para poder acceder a esas secciones.



39. Mensaje al intentar acceder a secciones de registro necesario.

Para conseguir una cuenta, se debe acceder al formulario de registro pulsando el botón de usuarios situado en la esquina superior derecha de la web y posteriormente a 'Registrarse'. Entonces aparecerá el formulario, como vemos en la Figura 40, que se debe de rellenar adecuadamente introduciendo los datos solicitados como son el nombre de usuario, el correo electrónico, una contraseña, la categoría de empleo a la que se pertenece, una provincia y un número de años de experiencia en la categoría elegida.



40. Pasos para completar el registro.

Una vez la cuenta ha sido registrada, se redirige automáticamente a la página de acceso, donde se solicita usuario y contraseña. Una vez introducidos los datos, se retorna automáticamente a la página principal. Entonces ya se tiene acceso a la funcionalidad de usuario registrado.

## 5. Funcionalidad de usuario registrado

Una vez que se dispone de cuenta de usuario y se ha accedido correctamente, se puede acceder a la página de ofertas recomendadas (Figura 41) y ofertas con mayor salario (Figura 42).

**JobTrends**

Buscar...

Inicio

Demanda de empleo

Por categoría

Por provincia

Contratos

Por temporalidad

Por sector

Ofertas

Ofertas recomendadas

Ofertas con mejor salario

**Ofertas recomendadas**

**Customer Success**

Empresa: Tu-app.net  
Ciudad: Pozuelo De Alarcón(Madrid)  
Salario: 24.000 € Bruto/año  
Experiencia mínima: Al menos 2 años  
Jornada de trabajo: Completa

**ANALISTA PROGRAMADOR**

Empresa: RCM JIT  
Ciudad: Madrid(Madrid)  
Salario: 33.000 € Bruto/año  
Experiencia mínima: Al menos 2 años  
Jornada de trabajo: Completa

**Instalador autónomo de FTTH fibra óptica**

Empresa: RYP Soluciones informáticas y de comunicaciones  
Ciudad: Madrid(Madrid)  
Salario: 3.600 € Bruto/mes  
Experiencia mínima: Al menos 1 año  
Jornada de trabajo: Completa

**INGENIERO SISTEMAS**

Empresa: RCM JIT  
Ciudad: Madrid(Madrid)  
Salario: No indicado  
Experiencia mínima: Al menos 2 años  
Jornada de trabajo: Completa

**Técnico Informático de Servicios Soporte - Junior**

**Programadores junior C# .NET con MVC**

41. Ofertas recomendadas.

**JobTrends**

Buscar...

Inicio

Demanda de empleo

Por categoría

Por provincia

Contratos

Por temporalidad

Por sector

Ofertas

Ofertas recomendadas

Ofertas con mejor salario

**Ofertas con mayor salario**

**Data Scientist**

Empresa: Aderen  
Ciudad: Las Rozas De Madrid(Madrid)  
Salario: 54.000 € Bruto/año  
Experiencia mínima: Al menos 2 años  
Jornada de trabajo: Completa

**Jefe de Proyecto - Google Cloud**

Empresa: Intelligence Partner S.L.  
Ciudad: Madrid (Madrid)(Madrid)  
Salario: 48.000 € Bruto/año  
Experiencia mínima: Al menos 2 años  
Jornada de trabajo: Completa

**Programador PHP Senior - Ecommerce - Magento**

Empresa: ECOMMBITS  
Ciudad: Madrid (Madrid)(Madrid)  
Salario: 36.000 € Bruto/año  
Experiencia mínima: Al menos 2 años  
Jornada de trabajo: Completa

**Analista programador con Angular y Java**

Empresa: Cibernos  
Ciudad: Madrid(Madrid)  
Salario: 42.000 € Bruto/año  
Experiencia mínima: Al menos 2 años  
Jornada de trabajo: Completa

**Arquitecto - Programador Senior Java**

**Consultor SAP**

42. Ofertas con mayor salario.

También existe la posibilidad de acceder a la información de la cuenta (Figura 43) pulsando en el botón de usuario en la esquina superior derecha y en 'Mi cuenta' y salir de la sesión pulsando el botón de ese mismo panel 'Cerrar sesión'.

JobTrends 👤

Buscar... 🔍

Inicio

Demanda de empleo <

- Por categoría
- Por provincia

Contratos <

- Por temporalidad
- Por sector

Ofertas <

- Ofertas recomendadas
- Ofertas con mejor salario

Mi cuenta

Tus datos

Usuario: **Rober**  
Correo electrónico: **rober@gmail.com**  
Estado de la cuenta: **Activa** ✓

43. Datos de la cuenta de usuario.