



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

DeltaOffers



Presentado por Miguel Ubierna Gutiérrez
en Universidad de Burgos - 11 de junio de 2024

Tutores: Dr. César Ignacio García Osorio
Dra. Ana Serrano Mamolar

Resumen

Actualmente, las convocatorias para el Personal Docente e Investigador y el Personal de Administración y Servicios de las universidades públicas de Castilla y León, se muestran en las páginas web de las respectivas universidades. Cada universidad muestra sus propias convocatorias en su web, esto hace que las correspondientes convocatorias solo estén al alcance de aquellas personas que entran en la web de una universidad específica.

El trabajo realizado llamado DeltaOffers está compuesto por dos aplicaciones independientes. La primera, se encarga de recopilar mediante *web scraping* todas esas convocatorias de las universidades públicas de Castilla y León y las almacena en una base de datos. La segunda es una aplicación web que muestra las convocatorias de forma centralizada. Esto amplía potencialmente el rango de candidatos en las convocatorias correspondientes.

Gracias a este proyecto, se puede evitar la ausencia de candidatos para determinadas plazas del PDI y PAS. Además, como la demanda de determinadas convocatorias será mayor, se podrá conseguir personal más cualificado para determinados puestos de trabajo.

Se puede acceder a la correspondiente página web a partir del siguiente enlace: <https://deltaoffers.azurewebsites.net/>

Descriptores

Convocatorias, Personal Docente e Investigador, Personal Administración y Servicios, Universidades, Web Scraping, Cron , Base de datos , Servidores Azure, Página Web, Correo Electrónico, ICalendar.

Abstract

Currently, the calls for Teaching and Research Staff and Administration and Services Staff of the public universities of Castilla y León are displayed on the universities' websites. Each university shows its own calls on its website, this means that the corresponding calls are only available to those people who enter the website of a specific university.

The project carried out called DeltaOffers is made up of two independent applications. The first is responsible for collecting all these calls from the public universities of Castilla y León through web scraping and storing them in a database. The second is a web application that displays the calls centrally. This potentially expands the range of candidates in the corresponding calls.

Thanks to this project, the absence of candidates for certain PDI and PAS positions can be avoided. Furthermore, as the demand for certain calls will be greater, more qualified personnel will be able to be obtained for certain jobs.

The corresponding web page can be accessed from the following link: <https://deltaoffers.azurewebsites.net/>

Keywords

Calls, Teaching and Research Staff, Administration and Services Staff Universities, Web Scraping, Cron, Database, Azure Servers, Website, Email, ICalendar.

Índice general

Índice general	iii
Índice de figuras	v
1. Introducción	1
1.1. Estructura de la memoria	2
1.2. Estructura de los anexos	3
1.3. Estructura del software	3
2. Objetivos del proyecto	5
2.1. Objetivos generales	5
2.2. Objetivos técnicos	5
2.3. Objetivos personales	6
3. Conceptos teóricos	7
3.1. Web Scraping	7
3.2. PDI y PAS	8
4. Técnicas y herramientas	11
4.1. Gestión de proyectos - Scrum	11
4.2. Herramienta de control de versiones - Git	11
4.3. Repositorio remoto - GitHub	12
4.4. Entorno de desarrollo	12
4.5. Web scraping	13
4.6. MySQL	15
4.7. ASP.NET Core MVC	16
4.8. GitHub Actions	17

4.9. Microsoft Azure	18
4.10. Diseño adaptativo	19
4.11. Testing	21
4.12. Guía de estilos de código	22
4.13. Documentación	22
4.14. Lenguajes de Programación	22
5. Aspectos relevantes del desarrollo del proyecto	25
5.1. Análisis de las webs de extracción de datos	25
5.2. Integración con Calendario Personal	28
5.3. Despliegue de la aplicación	30
6. Trabajos relacionados	33
6.1. Indeed	33
6.2. Chollometro	34
6.3. Google News	34
7. Conclusiones y Líneas de trabajo futuras	35
7.1. Conclusiones	35
7.2. Líneas de trabajo futuras	36
Bibliografía	37

Índice de figuras

4.1. Pasos Web Scraping con Beautiful Soup y Requests	14
4.2. Aplicación media queries en CSS	19
4.3. Visualización convocatorias en pantalla de ordenador	19
4.4. Visualización Convocatorias Pantalla Tablet	20
4.5. Visualización Convocatorias Pantalla Smartphone	21
4.6. Porcentaje de cada lenguaje de programación en el proyecto	23
5.1. Convocatoria en plazo ULE	27
5.2. Fichero Adjunto a Convocatoria	27
5.3. Estado Convocatoria Incorrecto	27
5.4. Convocatorias con campos vacíos	28
5.5. Diagrama arquitectura general del sistema.	31

1. Introducción

En las universidades españolas, existen dos colectivos que garantizan el desarrollo académico y administrativo. El **Personal Docente e Investigador**, es el encargado de desempeñar las funciones de docencia, investigación y gestión académica de las universidades. Este colectivo está formado por profesores, investigadores, ayudantes y catedráticos.

Por otro lado, el **Personal de Administración y Servicios**, es el encargado de la gestión administrativa, la prestación de servicios y el soporte a la docencia e investigación. Este colectivo está formado principalmente por administrativos, personal de mantenimiento y servicios y personal técnico.

Ambos colectivos son fundamentales para que el funcionamiento de las universidades sea exitoso. Por esta razón, las universidades necesitan que estos puestos sean desempeñados por personas altamente cualificadas y competentes.

«DeltaOffers», es un proyecto formado por dos aplicaciones independientes. La primera, es una aplicación que realiza *web scraping* a las webs de las universidades públicas de Castilla y León para recopilar sus convocatorias y almacenarlas en una base de datos. La segunda, es una aplicación web que tiene como objetivo mostrar todas las convocatorias recopiladas de PDI y PAS de una manera centralizada.

Con la aparición de «DeltaOffers» se espera poner fin a la ausencia de candidatos para plazas de PDI Y PAS de las universidades gracias a que, cualquier persona independientemente de la provincia en la que esté residiendo, podrá enterarse de las convocatorias sin necesidad de tener que acudir a las propias webs de las universidades.

Por otro lado, la existencia de una web centralizada de convocatorias, tiene como consecuencia un mayor alcance de las mismas. Estas convocatorias

podrán llegar con una mayor facilidad a los posibles interesados lo que implicaría un aumento en la demanda de determinados puestos de trabajo y como consecuencia, se espera incrementar la calidad del personal que finalmente sea seleccionado para cubrir una determinada plaza.

Esta web no solo centraliza todas las convocatorias de estas universidades sino que también lo hace con un diseño atractivo y una alta usabilidad proporcionando además funcionalidad adicional como un sistema de avisos por correo electrónico.

Este proyecto podría suponer un gran avance para las universidades públicas de Castilla y León mediante la contribución a un Personal Docente e Investigador y Personal de Administración y Servicios de calidad.

1.1. Estructura de la memoria

La estructura de la memoria es la siguiente:

- **Introducción:** incluye una descripción del entorno del proyecto y del proyecto en sí. Además, se detalla cual es la estructura de la memoria y del resto de materiales entregados.
- **Objetivos del proyecto:** expone cuales son los objetivos que se persiguen con la realización del proyecto. La mayoría de estos objetivos fueron diseñados al comienzo del proyecto y otros se han ido añadiendo en el transcurso del mismo.
- **Conceptos teóricos:** muestra los conceptos teóricos necesarios de comprender para un mejor entendimiento global del proyecto.
- **Técnicas y herramientas:** detalla las técnicas y herramientas utilizadas para el proyecto.
- **Aspectos relevantes del proyecto:** expone los aspectos más relevantes del proyecto y las soluciones aplicadas en estos casos.
- **Trabajos relacionados:** presenta algunos proyectos relacionados con el trabajo realizado.
- **Conclusiones y líneas de trabajo futuras:** detalla las conclusiones obtenidas tras la finalización del proyecto junto con un análisis sobre cómo se podría realizar una continuación y mejora del proyecto.

1.2. Estructura de los anexos

- **Plan de proyecto software:** indica cual ha sido la planificación del proyecto junto con un estudio de viabilidad legal y económica del mismo.
- **Especificación de requisitos:** describe cómo se va a comportar el sistema que ha sido desarrollado.
- **Especificación de diseño:** expone la estructura de los datos, los procedimientos utilizados y los patrones de diseño empleados.
- **Documentación técnica de programación:** detalla guías e información técnica del proyecto.
- **Documentación de usuario:** muestra una guía de utilización del producto realizado.
- **Anexo de sostenibilización curricular:** relaciona el proyecto realizado con los objetivos del desarrollo sostenible.

1.3. Estructura del software

Junto con los anexos y la memoria, se realizará también la entrega del software desarrollado, el cual, tendrá la siguiente estructura.

- **DataCollections:** Este directorio incluye la sección del proyecto en la que se realiza el *web scraping* y el *script* con el que se actualiza la base de datos. Por otro lado, también están aquí ubicados los test unitarios de la parte de recopilación de datos.
- **DeltaOffersWeb:** Esta carpeta contiene el proyecto en .NET mediante el que se desarrolla la página web. En él se encontrarán distintos directorios para controladores, vistas y modelos junto con otros ficheros de ajustes, ejecución , etc.
- **Github Workflows:** Este directorio contiene los archivos de extensión *.yml* los cuales son los encargados de realizar las tareas programadas correspondientes al proyecto mediante *cron*.

2. Objetivos del proyecto

En este capítulo, se van a mostrar los diferentes objetivos marcados para la realización del proyecto.

2.1. Objetivos generales

- Recopilar todas las convocatorias de PDI y PAS de las universidades públicas de Castilla y León.
- Realizar una web que mostrase de manera centralizada las convocatorias mencionadas.
- Mejorar el atractivo, la usabilidad y la estructura frente a las webs que muestran las convocatorias actualmente.
- Añadir nuevas funcionalidades en comparación con las webs actuales existentes.

2.2. Objetivos técnicos

- Recopilación de las convocatorias del PDI Y PAS mediante la realización de *web scraping* con Python a las secciones de las páginas webs en las que las universidades publican las convocatorias.
- Tratamiento de los datos obtenidos e implementación de mejoras en cuanto a estructura y formato de los datos para un posterior almacenamiento de manera segura.

- Implementación de *testing* mediante el uso de herramientas como *Unittest* o *MagicMock* para la simulación de objetos.
- Desarrollo de una página web con ASP .NET Core mediante la utilización de un Patrón MVC (Modelo-Vista-Controlador) que recopilase las convocatorias.
- Añadir funcionalidad a la web mediante un sistema de avisos por correo electrónico y la utilización de ICalendar.
- Realizar un despliegue completo de la aplicación mediante servidores Azure y Github Actions.
- Utilización de un sistema de control de versiones como Git y utilización de GitHub para alojamiento del código en la nube.

2.3. Objetivos personales

- Emplear metodologías ágiles como Scrum en un proyecto real.
- Utilización de distintas herramientas y aprendizaje del uso de las mismas.
- Aplicar los conocimientos adquiridos durante el grado.
- Comprobar mi capacidad de resolución de problemas en proyectos de mayor calibre.
- Desarrollo y despliegue de mi primera página web.
- Mejorar mi habilidad con las herramientas Git y GitHub.

3. Conceptos teóricos

Para un entendimiento general de este proyecto, es necesario conocer previamente algunos conceptos teóricos. Estos conceptos, van a ser desarrollados a continuación.

3.1. Web Scraping

En este proyecto, el proceso de recopilación de datos ha sido realizado mediante la técnica llamada *web scraping*. Esta técnica consiste en la extracción de contenidos, datos e información de las páginas web mediante la utilización de herramientas software.

Esta metodología es muy común debido a su facilidad de extraer grandes cantidades de datos de una manera automatizada.

Actualmente, este proceso se puede realizar en un lenguaje de programación en concreto mediante la utilización de bibliotecas o con herramientas existentes dedicadas a realizar estas labores. El inconveniente de estas herramientas es que algunas no son de suficiente calidad o tienes que pagar por ellas por lo tanto se recomienda que si se tienen conocimientos de programación, se realice este proceso mediante código.

Para la extracción de datos mediante *web scraping* de manera automática podemos encontrar tres principales métodos:

- **Scraping mediante la utilización de bots:** Los cuales extraen la información de la web automáticamente.
- **Scraping a través de un analizador sintáctico:** En este método, se dispone de un parser que es el encargado de convertir texto a una

estructura para el almacenamiento de información. Este ha sido el método usado en este proyecto.

- **Scraping mediante análisis de textos:** Consiste en la utilización de la función *grep* de Unix para encontrar la información en la web.

Esta tecnología enormemente potente, es popularmente utilizada en el mundo de la industria. Esto se debe a que se pueden recopilar grandes volúmenes de datos de una manera sencilla con la utilización de la misma.

Estos datos posteriormente pueden ser tratados y analizados por analistas los cuales se encargan de obtener conclusiones en función de los datos obtenidos. Esas conclusiones obtenidas pueden conllevar decisiones estratégicas y de negocio dentro de las empresas y como consecuencia un aumento en la facturación en muchas de ellas. A continuación, se van a detallar algunos casos reales en los que se utiliza el *web scraping* en el mundo empresarial:

- Estudios de mercado.
- Automatización de negocios.
- Generación de clientes potenciales.
- Análisis y seguimientos en los precios.
- Recopilación de noticias.

3.2. PDI y PAS

Dado que el proyecto no tiene abundantes conceptos teóricos, se ha decidido añadir a este capítulo, un apartado en el que se va a explicar qué es el Personal Docente e Investigador y el Personal de Administración y Servicios. Además, se van a comentar cuales son los procesos de selección que necesita superar una persona para acabar logrando alguna de estas plazas.

Personal Docente e Investigador

El Personal Docente e Investigador es el encargado de desarrollar las funciones de enseñanza e investigación en las universidades [26]. Está compuesto de personal docente e investigador funcionario y de personal contratado.

Personal Docente e Investigador funcionario

Para desempeñar este puesto, en primer lugar se debe tener aprobada una oposición. Además, también es necesario tener la acreditación nacional para el cuerpo correspondiente. Dentro del PDI funcionario encontramos dos tipos:

- Catedráticos
- Profesores titulares

Personal Docente e Investigador contratado

La Ley Orgánica de Universidades, es la encargada de regular las modalidades de contratación para estos profesionales. Al igual que en el PDI funcionario, existen dos grupos bien diferenciados. Estos son, el PDI contratado indefinidamente y el PDI contratado temporalmente. Este segundo grupo no puede ser superior en cuanto a personal que el 40 % de la plantilla docente de una universidad.

Los tipos de PDI contratado que podemos localizar son los siguientes:

- Profesor Ayudante Doctor
- Ayudante
- Profesor Contratado Doctor
- Profesor Visitante

Además, también podemos encontrar la figura del profesor asociado. Estos deben de ser profesionales con una notoria experiencia profesional y se encargan de impartir docencia [\[13\]](#).

Para ser elegido como miembro del PDI de una universidad pública, es necesario pasar por un proceso administrativo previo. Este proceso se basa en lo siguiente:

- **Publicación de la convocatoria:** Cada universidad publicará en su propia web las convocatorias que ofertan en las que indican detalles como la titulación requerida, plazas ofertadas, etc.
- **Presentación de candidatos:** Los interesados en determinadas convocatorias, deberán entregar la documentación requerida en el plazo indicado.

- **Evaluación de méritos:** Los evaluadores, serán los encargados de determinar a quiénes se van a asignar las plazas en función de la documentación aportada.
- **Resolución de la convocatoria:** Una vez determinado el miembro o los miembros para ocupar determinadas plazas, se publicará un listado de admitidos y excluidos.
- **Plazo de subsanación:** En ocasiones es posible que se abra un plazo de subsanación por si se hubiese producido algún error en la selección de los candidatos admitidos.

Personal de Administración y Servicios

El PAS, es el encargado de desarrollar la gestión técnica, económica y administrativa de la Universidad [25]. Son personas fundamentales para el correcto funcionamiento de las instituciones universitarias y principalmente, podemos dividir este grupo en dos categorías:

- **PAS laboral:** son aquellos que tienen un contrato laboral vinculado con la universidad. En este grupo se encuentran los auxiliares de servicios, personal de mantenimiento, técnicos informáticos, etc.
- **PAS funcionario:** son trabajadores que han superado una oposición y están vinculados a la Administración Pública.

El proceso de selección para ser miembro del PAS es muy similar al del PDI mencionado previamente. Sin embargo, en el proceso de selección del PDI, se suelen valorar más los aspectos académicos y la experiencia previa en docencia o investigación mientras que, para el PAS se centran en capacidades más técnicas, habilidades u otras competencias [12].

4. Técnicas y herramientas

4.1. Gestión de proyectos - Scrum

Scrum es un marco de trabajo ágil utilizado para la gestión de proyectos software. Este marco de trabajo está basado en *sprints* de entre una y cuatro semanas de duración. En estas iteraciones se genera un incremento de software que sea potencialmente entregable [33].

En este proyecto, se ha utilizado esta metodología pero algo adaptada, dado que es un proyecto individual. Las reuniones retrospectivas propias de Scrum, se tenían con los tutores del proyecto cada dos semanas. En estas reuniones se evaluaban las tareas realizadas hasta la fecha y se establecían unos nuevos objetivos para el próximo sprint. Estas tareas, fueron detalladas en el apartado de *Issues* en la plataforma GitHub.

4.2. Herramienta de control de versiones - Git

Git es un sistema de control de versiones distribuido. Los desarrolladores dispondrán de copias completas del repositorio permitiendo de esta manera trabajar sin conexión o de forma remota con facilidad [18].

Los desarrolladores deben hacer *commit* para almacenar su trabajo de manera local y posteriormente se sincronizará la copia del repositorio con la del servidor. Esto nos proporciona gran facilidad para tener un control total sobre los cambios realizados en nuestro código.

En el proyecto, se ha trabajado con esta herramienta mediante el uso de la terminal *Git Bash*. Además, se realizaban actualizaciones periódicas

al repositorio de GitHub para, de esta manera, que permaneciesen tanto el repositorio local como el repositorio remoto actualizados.

4.3. Repositorio remoto - GitHub

GitHub es una herramienta web que permite el alojamiento de proyectos en repositorios remotos. Además, utiliza Git para el control de versiones lo cual es tremendamente útil para tener un control de los cambios realizados a lo largo del ciclo de vida del proyecto.

Los repositorios son los lugares donde se almacena el código del proyecto en la nube. Estos pueden ser públicos o privados en función de la decisión del usuario [29].

En este proyecto se ha decidido utilizar esta herramienta debido a la facilidad de uso y las enormes ventajas que nos proporciona. Además es una herramienta sobre la cual es fundamental tener amplios conocimientos debido a la importancia que tiene en el mundo laboral.

El enlace al repositorio de GitHub utilizado en este proyecto, es el siguiente: <https://github.com/miguelUbierna/DeltaOffers>

4.4. Entorno de desarrollo

Visual Studio Code

Visual Studio Code es un editor de código desarrollado por Microsoft disponible para diferentes Sistemas Operativos [34]. Ofrece numerosas funcionalidades como autocompletado inteligente, refactorización o depuración de código.

Este editor de código ha sido el utilizado para realizar la aplicación en la que se realiza el *web scraping* y se actualiza la base de datos. El lenguaje de programación utilizado en esta herramienta ha sido Python, para ello, Visual Studio Code nos proporciona una extensión de Python para facilitarnos el desarrollo en este lenguaje.

Visual Studio Community

Microsoft Visual Studio es un entorno de desarrollo disponible para diferentes sistemas operativos. Este nos permite programar en múltiples lenguajes y entornos de desarrollo.

Este entorno ha sido utilizado para realizar la aplicación web del proyecto mediante la utilización del *framework* ASP .NET Core MVC. Se eligió este entorno de desarrollo por la facilidad de creación de aplicaciones web con el mismo [32].

4.5. Web scraping

El *web scraping* es la técnica utilizada para la extracción de información y datos de las páginas web [35]. Los usuarios que interactuamos con las aplicaciones web, podemos ver la información que aparece en las mismas. Sin embargo, los datos e información en la mayoría de páginas webs, no están disponibles para su descarga. Es ahí donde el *web scraping* juega un papel importante [2].

En el caso del proyecto, este proceso de raspado de la web ha sido realizado mediante código, concretamente utilizando el lenguaje de programación Python.

Para ello, hemos tenido que ayudarnos de algunas librerías existentes que están dedicadas a realizar este tipo de técnicas.

En este proyecto en concreto, hemos realizado *web scraping* de las webs de las universidades públicas de Castilla y León para obtener los datos sobre sus convocatorias.

A continuación, se van a comentar las bibliotecas que nos han sido de ayuda para este proceso.

Requests

Requests es una biblioteca disponible para utilizar desde el lenguaje de programación Python que nos permite realizar solicitudes HTTP de manera muy sencilla. Esta herramienta se utiliza para obtener el contenido de páginas web o para hacer peticiones a APIs [27].

Esta biblioteca ha sido utilizada para hacer las peticiones HTTP a las webs de la Universidad de Burgos y la Universidad de León.

Para un uso correcto de la misma, en primer lugar hacíamos una petición y verificábamos que el código de estado devuelto era correcto. Posteriormente, accedíamos el texto que devolvía esa petición, el cual era el código HTML de la página web solicitada.

Una vez que tenemos el código HTML en nuestro programa, finalizaba el uso de esta biblioteca y comenzábamos a utilizar otras más específicas para la extracción de datos.

Beautiful Soup

Beautiful Soup es una biblioteca de Python utilizada para la extracción de datos de los documentos HTML Y XML [22]. Para una utilización exitosa de esta herramienta, es necesario indicar cual es el *parser*. Este va a ser el encargado de transformar los documentos HTML en un árbol de objetos. El parser utilizado en este proyecto ha sido *lxml*, este es uno de los más comúnmente utilizados .

Un paso previo antes de utilizar esta biblioteca, es encontrar y localizar los elementos de interés. Para ello, en este proyecto se realizó un proceso de análisis previo en el que se inspeccionó el código HTML de las webs a las que se quería hacer el raspado.

Por último, se accederán a los elementos deseados que han sido analizados previamente. Este acceso a elementos se puede realizar a través de las etiquetas, clases, identificadores, atributos o texto. En el caso de este proyecto, en la mayoría de ocasiones se ha realizado el acceso a elementos mediante sus clases, pero en ocasiones dado que había varias clases del mismo nombre, se ha cambiado esta estrategia [3].

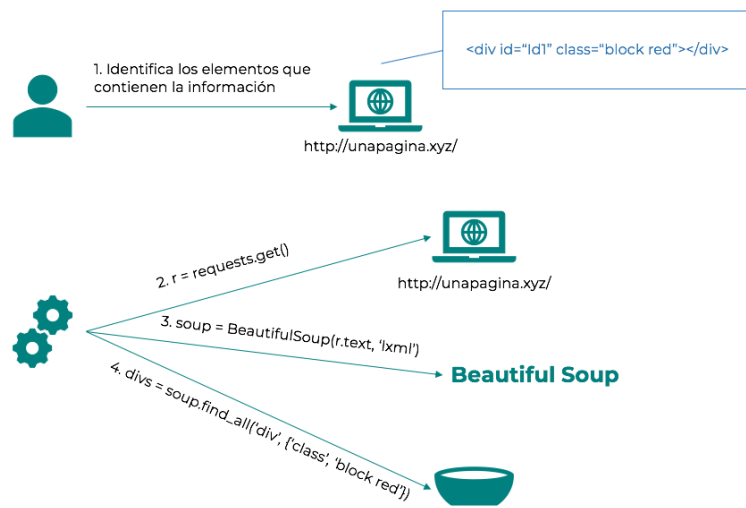


Figura 4.1: Pasos Web Scraping con BeautifulSoup y Requests

Selenium

Selenium es una herramienta que permite la automatización de navegadores web. Principalmente, es utilizada para la grabación, edición o depuración de pruebas en aplicaciones web pero también es posible utilizarla para la realización de *web scraping* [23].

Esta herramienta está disponible para múltiples lenguajes de programación, en este caso se ha utilizado con Python. Para que Selenium funcione correctamente, los usuarios necesitan dos requisitos previos:

- **Navegador Web:** Se necesita que el usuario tenga un navegador web instalado en el sistema. En el proyecto, el navegador utilizado fue Firefox dado que esta herramienta fue utilizada para hacer *web scraping* de la UVA cuya web recomendaba que fuese abierta con dicho navegador.
- **Driver:** Es el encargado de manejar las peticiones del usuario y el que nos permite interactuar con las páginas web.

Se optó por la utilización de esta herramienta dado que el contenido de la página web de la Universidad de Valladolid se cargaba mediante JavaScript dinámico. Por esta razón, pese a tener un mayor dominio de la herramienta BeautifulSoup, se realizó el *web scraping* de la web de la universidad mencionada con Selenium.

4.6. MySQL

MySQL es un sistema de gestión de bases de datos de código abierto desarrollado por Oracle el cual utiliza el lenguaje SQL [1].

Se decidió utilizar MySQL como herramienta de gestión de bases de datos debido a su rendimiento y fiabilidad de uso. Además, gracias a que la base de datos es relacional, tendremos una estructura del dato mucho más definida.

Quizás, el acceso a datos sea más rápido y de menor coste en cuanto a rendimiento mediante la utilización de una base de datos no relacional, pero, dado que la aplicación realizada no requiere de una carga de datos masiva, esta diferencia en cuanto a tiempos es despreciable.

En las primeras etapas del proyecto, instalé MySQL Server en mi máquina local. Esto me permitió realizar las pruebas de carga de datos en la base de

datos utilizada. Posteriormente, una vez que tenía la aplicación preparada para el despliegue, decidí migrar la base de datos de mi entorno local a un servidor MySQL Azure.

4.7. ASP.NET Core MVC

ASP.NET Core MVC es un *framework* utilizado para construir aplicaciones web y API mediante la utilización del patrón de diseño Modelo-Vista-Controlador [15]. El lenguaje de programación utilizado en este *framework* es principalmente C# aunque también contiene soporte para HTML, CSS y Javascript.

Las principales ventajas que nos proporciona este *framework* son las siguientes:

- Framework Multiplataforma
- Proporciona gran rendimiento y escalabilidad
- Modular
- Soporte para la inyección de dependencias

Esta herramienta ha sido la utilizada para la aplicación web del proyecto y se han seguido con ella la siguientes pasos para el desarrollo de la misma:

- En primer lugar, se definían los modelos y el contexto en función de la base de datos MySQL diseñada previamente.
- Posteriormente, se realizaba el acceso a datos e implementación de la lógica desde los controladores. Una vez la lógica estaba implementada, se pasaban los datos a la vista.
- La vista será la encargada de mostrar los datos y la interacción con el usuario final.

Una vez entendido cómo funciona este *framework*, vamos a mencionar cuales han sido las extensiones utilizadas en la aplicación.

X.PagedList.Mvc.Core

Esta librería permite aplicar paginación a listas en aplicaciones web ASP.NET Core MVC [11]. Esto mejora enormemente la usabilidad de la aplicación dado que las convocatorias del PDI Y PAS estarán divididas en distintas páginas permitiendo navegar fácilmente entre ellas.

Pomelo.EntityFrameworkCore.MySql

Esta librería es la que nos ha permitido conectar de manera eficiente nuestra aplicación en .NET con la base de datos MySQL.

Pomelo.EntityFrameworkCore.MySql es un proveedor de Entity Framework Core construido sobre MySqlConnection que permite el uso de Entity Framework Core ORM con MySQL. Esto nos ha permitido la interacción con la base de datos mediante objetos en vez de utilizar consultas SQL [9].

MailKit

Esta biblioteca de .NET tiene como propósito el envío, recepción y manipulación de correos electrónicos y suele ser frecuentemente usada en aplicaciones web que utilizan la automatización de correos electrónicos [20].

En el proyecto realizado, ha sido utilizada para el envío de correos mediante la utilización del protocolo SMTP [14]. El protocolo simple de transferencia de correo, es un protocolo TCP/IP utilizado para el envío y la recepción de correos electrónicos.

Esta biblioteca y este protocolo han sido utilizados para la implementación de un sistema de avisos basado en correo electrónico.

4.8. GitHub Actions

Esta herramienta, permite la personalización, automatización y ejecución de flujos de trabajo en los repositorios de GitHub.

Este es un servicio de integración continua y entrega continua que permite la automatización de compilaciones, pruebas y despliegue.

Estos flujos de trabajo aparecen en el directorio *.github/workflows* dentro de un repositorio de GitHub. Este directorio contendrá uno o varios flujos de trabajo los cuales realizarán unas tareas [10].

En el caso del proyecto, se ha utilizado esta herramienta para la realización de actualizaciones periódicas a la base de datos. Para ello, se han definido dos tareas programadas que ejecutarán dos archivos Python en función de la fecha y hora definida en el *cron*. Esta tarea programada al igual que el resto de flujos de trabajo, ha sido definida en un archivo de extensión *.yml*.

La utilización de GitHub Actions ha sido de tremenda ayuda a la hora del despliegue del proyecto dado que ha permitido la ejecución de tareas programadas sobre el código sin necesidad de realizar estas tareas de manera local o con un coste mayor.

4.9. Microsoft Azure

Azure es una plataforma de Microsoft que ofrece servicios en la nube, con ella, se pueden crear, probar y desplegar aplicaciones y servicios mediante la utilización de los servidores proporcionados por Microsoft. Esta plataforma supone una enorme ventaja a la hora de tener tus proyectos en la nube sin que sea necesario tener una infraestructura física propia [31].

Los servicios Azure utilizados en este proyecto, han sido dos:

- **Servidor de Azure Database para MySQL** [17]: Proporciona MySQL totalmente administrado de gran rentabilidad, seguridad y flexibilidad. Este servidor se ha utilizado para desplegar la base de datos MySQL utilizada en el proyecto con las convocatorias correspondientes del PDI Y PAS.
- **Azure App Services** [16]: Este es un servicio basado en HTTP que permite hospedar aplicaciones web en la nube. Este servicio permitió desplegar la aplicación con facilidad, de manera segura y sin tener preocupación por la infraestructura del sistema. Otra ventaja de este servicio, es su compatibilidad con diferentes *frameworks* y lenguajes de programación.

Por otro lado, Azure App Service permite la integración con otros servicios de Azure. Esto nos permitió realizar la conexión del servidor MySQL con el que se desplegó la base de datos de manera sencilla.

4.10. Diseño adaptativo

El diseño adaptativo es una rama del diseño web que tiene como objetivo la adaptación de las ventanas de las páginas web a cualquier tipo de dispositivo independientemente del ancho de pantalla [28]. Esto quiere decir que una página web adaptativa, debe poder utilizarse en ordenadores, *tablets* o *smartphones*.

Para conseguir el objetivo de que la página web sea adaptativa, se han utilizado las *media queries* [7], estas permiten la aplicación de estilos CSS según un ancho determinado de la pantalla.

Para ello, se iban indicando los correspondientes anchos de pantalla e interiormente se van definiendo los nuevos estilos CSS. A continuación se muestra en ejemplo de una de las *medias queries* realizadas en el proyecto.

```
@media screen and (max-width: 1263px) {  
  .filtros {  
    width: 90%;  
    font-size: medium;  
    padding: 20px;  
  }  
}
```

Figura 4.2: Aplicación media queries en CSS

Finalmente, se consiguió que nuestra web tuviese un diseño adaptativo. A continuación, se va a mostrar como quedaría una ventana de la aplicación desarrollada para distintos dispositivos.



Figura 4.3: Visualización convocatorias en pantalla de ordenador

The image shows a tablet screen with a dark header containing a logo 'D' and the text 'DELTA OFFERS'. Below the header is a filter section with three dropdown menus: 'Universidad:' (set to 'Todas'), 'Categoría:' (set to 'Todas'), and 'Plazo:' (set to 'Seleccionar plazo'). A 'Filtrar' button is to the right of the 'Plazo' dropdown. Below the filters are four job listings, each in a white card with a green gradient at the bottom. Each card contains the reference number, title, category, deadline, university, and two buttons: 'Recibir Avisos' and 'Ver Detalles'.

Universidad: Todas
Categoría: Todas
Plazo: Seleccionar plazo **Filtrar**

Ref.: 2009/00006/003/117 Convocatoria de ayudas para el PDI de la UBU para el diseño y ejecución de SAP (Short Advanced Programmes) en el marco del Proyecto RUN-EU 2.0

CATEGORIA: PDI **Recibir Avisos**
PLAZO: Desde 26/04/2024 hasta 31/12/2024 **Ver Detalles**
UNIVERSIDAD: Universidad de Burgos

EN PLAZO

CONVOCATORIA DE 6 DE JUNIO DE 2024. PLAZAS DE PROFESOR AYUDANTE DOCTOR.

CATEGORIA: GENERAL **Recibir Avisos**
PLAZO: Desde 06/06/2024 hasta 28/06/2024 **Ver Detalles**
UNIVERSIDAD: Universidad de Leon

EN PLAZO

Convocatoria proceso selectivo acceso libre Auxiliares de Servicios

CATEGORIA: PAS **Recibir Avisos**
PLAZO: Hasta 18/06/2024 **Ver Detalles**
UNIVERSIDAD: Universidad de Burgos

EN PLAZO

PI AUXILIARES DE ARCHIVOS, BIBLIOTECAS Y MUSEOS: RR 10/05/2024 por la que se convocan pruebas selectivas, por el turno de promoción interna.

CATEGORIA: PAS **Recibir Avisos**
PLAZO: Desde 21/05/2024 hasta 17/06/2024

Figura 4.4: Visualización Convocatorias Pantalla Tablet



Figura 4.5: Visualización Convocatorias Pantalla Smartphone

4.11. Testing

La implementación de test unitarios, fue realizada en la aplicación de *web scraping*. De esta manera se comprobaban si las convocatorias y campos obtenidos eran los correctos.

Las herramientas de *testing* utilizadas han sido las siguientes:

- **Unit Test:** [21] Esta es una biblioteca utilizada en Python para la creación y ejecución pruebas de manera automática. Unit Test permite la realización de pruebas unitarias de una manera organizada y con gran variedad de métodos de aserción. La utilización de esta herramienta aplicada a programas en Python es clave para que los desarrollos realizados sean de calidad.

- **MagicMock:** Esta biblioteca de pruebas utilizada en Python, permite el reemplazamiento de partes del sistema a cambio de objetos simulados. Esto ayuda a reducir la cantidad de código auxiliar que muchas veces podemos encontrar en los entornos de pruebas. De esta manera evitamos la utilización de los objetos para la realización de test unitarios de manera directa.

4.12. Guía de estilos de código

Para asegurarnos de disponer de un código bien estructurado y de calidad, se decidió utilizar la guía de estilos PEP 8. Esta guía de estilos fue aplicada gracias a la biblioteca Autopep8 proporcionada y a su fácil integración con el entorno de desarrollo Visual Studio Code.

Esta guía define conceptos como la longitud de las líneas, la indentación, las líneas en blanco y la guía de importación de otros paquetes [24].

4.13. Documentación

L^AT_EX ha sido la herramienta utilizada para la documentación del proyecto. Esta, permite la creación de textos orientados a documentos escritos de gran calidad tipográfica [30].

Esta herramienta suele ser utilizada para la redacción de libros y artículos científicos, esto se debe a la gran facilidad que presenta para la escritura de fórmulas y expresiones matemáticas. Quizás no se utilice todo lo que debería dado que comparada con otros procesadores de texto, esta tiene una curva de aprendizaje alta.

4.14. Lenguajes de Programación

En esta sección se van a comentar los lenguajes de programación utilizados para el desarrollo del proyecto y la finalidad con la que se ha utilizado cada uno de ellos.

- **Python:** Este lenguaje ha sido utilizado para la realización del *web scraping*. Además, una vez recopilados los datos, se realizó un tratamiento y limpieza de los mismos.

Finalmente, este lenguaje también fue utilizado para la actualización de la base de datos del sistema.

- **CSS:** Lenguaje que se encarga de la apariencia y diseño de la página web realizada. Otro aspecto importante a considerar, serían las *media queries* aplicadas para la realización de la aplicación con diseño adaptativo.
- **C#:** Utilizado para el desarrollo de la aplicación web en ASP .NET Core MVC dentro de los modelos, los controladores e incluso en las vistas de manera embebida.
- **HTML:** Lenguaje utilizado para mostrar la estructura y contenido de la página web.
- **JavaScript:** Utilizado para aportar funcionalidades en el lado del cliente. Principalmente utilizado para validaciones, formularios y animaciones en la aplicación web.

GitHub proporciona una gráfica en la que se indica qué porcentaje de cada lenguaje de programación hay en mi repositorio. La gráfica es la siguiente:

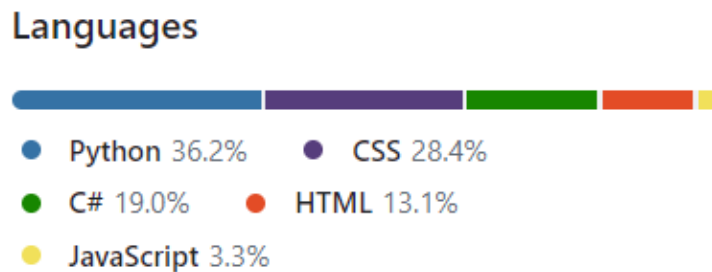


Figura 4.6: Porcentaje de cada lenguaje de programación en el proyecto

5. Aspectos relevantes del desarrollo del proyecto

En este apartado, se van a detallar los aspectos más interesantes del desarrollo del proyecto y los desafíos a los que me he tenido que enfrentar a lo largo del mismo.

Además, es importante detallar las decisiones que se tomaron en esos momentos de dificultad e indicar de qué manera se solventaron los problemas.

Por otro lado, también voy a indicar en este apartado cual es la arquitectura general del sistema diseñado dado que me parece que es un aspecto interesante dentro del proyecto.

5.1. Análisis de las webs de extracción de datos

En las primeras fases del proyecto, se realizó un análisis de las webs de las Universidades públicas de Castilla y León para estudiar cuales podrían ser los atributos de interés para mostrar en la web del proyecto. Durante este proceso, surgieron varios problemas para la obtención de información. Esto se debió a que cada universidad subía las convocatorias a su web con un formato particular y con unos datos e informaciones particulares. Esto es un problema a la hora de querer montar una web de convocatorias centralizada dado que uno de los aspectos más importantes de una web es la uniformidad de la misma. A continuación, se van a comentar algunos de los desafíos que fueron encontrados durante esta fase.

Universidad de Salamanca

Debido a mi poco conocimiento en cuanto a las convocatorias PDI y PAS, en primer lugar se realizó *web scraping* de un portal de la USAL en el que se publicaban todas las convocatorias pero en el que no se tenía conocimiento si una convocatoria estaba en plazo o no. Esto se debe a que las fechas que comprendían el plazo no estaban publicadas en páginas web sino en documentos PDF.

Se planteó la posibilidad de realizar PDF scraping de las convocatorias de esta universidad pero tras tiempo de investigación, se descubrió que estos ficheros no seguían un formato unificado. Finalmente, debido a la imposibilidad de recopilación de datos de la web de la USAL [5] debido a la falta de datos necesarios para la web a desarrollar, se decidió excluir esta universidad del proyecto.

Universidad de León

El realizar el raspado de datos de esta universidad, también fue un reto en el proyecto. Esta universidad mostraba algunas convocatorias en plazo mientras que en verdad ya no estaban abiertas a candidatos (Por ejemplo, ya se habían publicado los listados de admitidos) [4]. Para determinar si una oferta estaba en plazo o no, la única manera era mediante la examinación de los títulos de los PDF adjuntos a una determinada convocatoria. Para ello, se realizó una estructura de datos con las palabras no deseadas con aquellas palabras que, en el caso de localizar alguna en un título ya sabíamos que esa convocatoria no estaba en plazo.

Esto llevo largos tiempos de análisis para finalmente tan solo acabar detectando las convocatorias que estaban en plazo verdaderamente. Es posible, que en casos futuros, vuelvan a surgir convocatorias fuera de plazo que están detectadas como que si lo son. Para ello, es necesario que si se continúa con el desarrollo de la aplicación, se tenga muy en cuenta este aspecto.

A continuación se va a mostrar un ejemplo en el que se da este caso. En la siguiente imagen, se va a exponer una convocatoria que actualmente está en plazo:

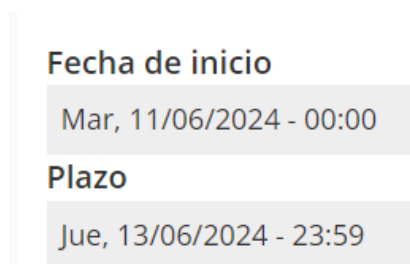


Figura 5.1: Convocatoria en plazo ULE

Sin embargo, si nos fijamos en el archivo adjunto a esa convocatoria, aparecerá una resolución de aprobados, lo cual quiere decir que la convocatoria no está en plazo.

Adjunto


 [resolucion provisional aprobados segundo ejercicio.report.pdf](#)

Figura 5.2: Fichero Adjunto a Convocatoria

Con la implementación de la estructura de datos de las palabras no deseadas, esta convocatoria no sería elegida.

Universidad de Valladolid

Al realizar *web scraping* de las páginas web en las que se muestran las convocatorias PDI Y PAS de la Universidad de Valladolid [6], también surgieron una serie de imprevistos. Esto se debe a que había convocatorias que aparecen en esta web cuyo estado indicaba que estaba 'EN PLAZO' pero si nos fijamos en las fechas de las mismas en verdad no lo están.

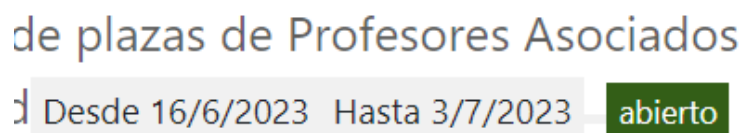


Figura 5.3: Estado Convocatoria Incorrecto

Esto fue algo que no percibí en la primera etapa del proyecto, posteriormente, decidí arreglarlo para que las ofertas se mostrasen en plazo con exactitud mediante la comparación de las fechas del plazo con la fecha actual.

Otro de los problemas, fue que, en numerosas convocatorias de la Universidad de Valladolid, había campos vacíos, esto hizo que en algunas convocatorias que por ejemplo no indicaban cuales eran su fecha de inicio de plazo y su fecha de fin, se tuviesen que determinar por cerradas.



The image shows a recruitment notice for the University of Valladolid. The title is '2024DFCAD1 - Resolución Universidad de Valladolid, Cuerpos Docentes Univers'. Below the title, there are three fields: 'Tipo: Profesorado funcionario', 'Clasificación:', and 'Estado: Abierto'. The 'Clasificación:' field is empty, indicating a missing or blank value.

Figura 5.4: Convocatorias con campos vacíos

Estos han sido algunos inconvenientes a la hora de recopilar los datos de las distintas universidades. Todos ellos se han tratado de solventar de la manera más lógica posible intentando que afecten de la manera más mínima a la idea de aplicación que se tenía en el inicio.

5.2. Integración con Calendario Personal

Para añadir funcionalidad extra a la aplicación, se decidió realizar una integración con Google Calendar, para ello, en primer lugar, se tenía que dar la posibilidad a los usuarios de iniciar sesión con Google. Esto se implementó mediante la autenticación basada en OAuth 2.0 lo cual permitía a los usuarios de mi aplicación web iniciar sesión mediante la utilización de las cuentas de Google [19].

Posteriormente, cuando el proceso de autenticación estaba completado, se realizó un desarrollo para que los usuarios pudiesen añadir como evento,

las fechas de fin de las convocatorias a Google Calendar desde la aplicación y de esta manera disponer de un recordatorio previo antes de que las convocatorias expirasen. Esta integración se intentó realizar mediante Google Cloud Console que permitía a los desarrolladores gestionar sus servicios en Google Cloud.

Esta integración con Google y Google Calendar, se acabó llevando a cabo y era funcional para usuarios de test, sin embargo, si se quería publicar la aplicación y que la integración fuese funcional para cualquier usuario, Google requería lo siguiente para completar el proceso de verificación:

- Un enlace oficial a la Política de Privacidad de la aplicación
- Un vídeo de YouTube que muestra cómo se planea utilizar los datos de usuario de Google que obtiene de los ámbitos.
- Una explicación escrita que le indique a Google por qué necesita acceso a datos de usuario confidenciales y/o restringidos.
- Todos tus dominios verificados en Google Search Console

Estas restricciones, fueron consideradas demasiado duras como para un proyecto de este calibre. Además, quizás esta integración limitaba el disfrute de la funcionalidad de la aplicación para aquellos usuarios que no dispusiesen de cuenta de Google, por lo tanto se decidió cambiar esta funcionalidad y se implementó un nuevo sistema de avisos por correo.

Nuevo Sistema de Avisos por Correo

Para este sistema de avisos implementado, los usuarios no tenían necesariamente que iniciar sesión con su cuenta de correo electrónico de manera inicial como anteriormente.

Para la utilización de esta funcionalidad, bastará con que el usuario se suscriba al sistema de avisos de la convocatoria que el desee mediante un formulario en el que debe introducir su correo electrónico.

Una vez que se suscribe, le llegará al usuario un correo con información de la convocatoria solicitada y un archivo ICalendar definido por la aplicación. Estos archivos permiten añadir como evento a su calendario personal la fecha de fin de una convocatoria para que el usuario se acuerde del fin de plazo de la misma. Además, en este caso dará igual cual sea el servicio de correo electrónico utilizado (Google, Outlook...) lo cual mejora enormemente la implementación anterior.

5.3. Despliegue de la aplicación

La idea inicial para realizar el despliegue de la aplicación, era Docker [8]. Esta es una herramienta que permite a los desarrolladores a crear, compartir, ejecutar y verificar aplicaciones en cualquier lugar sin complejas configuraciones del entorno.

Sin embargo, cuando se utiliza Docker en Windows, es necesario tener habilitada alguna tecnología de virtualización, normalmente el Subsistema de Windows para Linux. Para ello, se necesita habilitar la característica WSL en mi ordenador lo cual era una operación que no acababa de completarse debido a errores en el camino.

Tras varias pruebas, cambios en archivos de configuración, etc. Se decidió investigar otras herramientas para el despliegue. Esta decisión tomada, es algo que ha dado un salto de calidad al proyecto debido a que actualmente la aplicación está desplegada de una manera muy profesional y novedosa.

Los componentes fundamentales para el despliegue final de la aplicación han sido los siguientes:

- **Servidor de Azure Database para MySQL:** [17] En este servidor en la nube, se ha desplegado la base de datos MySQL. Esta decisión se ha tomado de esta manera debido a la facilidad que nos proporciona Azure para realizar despliegues sin realizar tareas de infraestructura.
- **GitHub Action:** [10] Esta herramienta es utilizada para la automatización de flujos de trabajo. En el sistema, se ha utilizado esta herramienta para crear una tarea programada que ejecutará el proyecto en Python y por lo tanto hará que la base de datos se actualice de manera periódica.
- **Azure App Services:** [16] Este es un servicio basado en HTTP que permite hospedar aplicaciones web en la nube. En nuestro sistema, es el encargado de lanzar y alojar la aplicación desarrollada en .NET en la nube. Esto fue sencillo debido a que tanto Azure como .NET son propias de Microsoft por lo tanto el despliegue se pudo realizar con facilidad. Para que la información de la base de datos se mostrase correctamente en la web desplegada, se tuvieron que configurar las variables de entorno de este App Service e introducir la cadena de conexión de la base de datos desplegada en el servidor Azure Database. Este fue el último paso del montaje de la arquitectura del sistema y por lo tanto ya estaba la aplicación funcional disponible para utilizarse.

Diagrama de arquitectura general del sistema desplegado

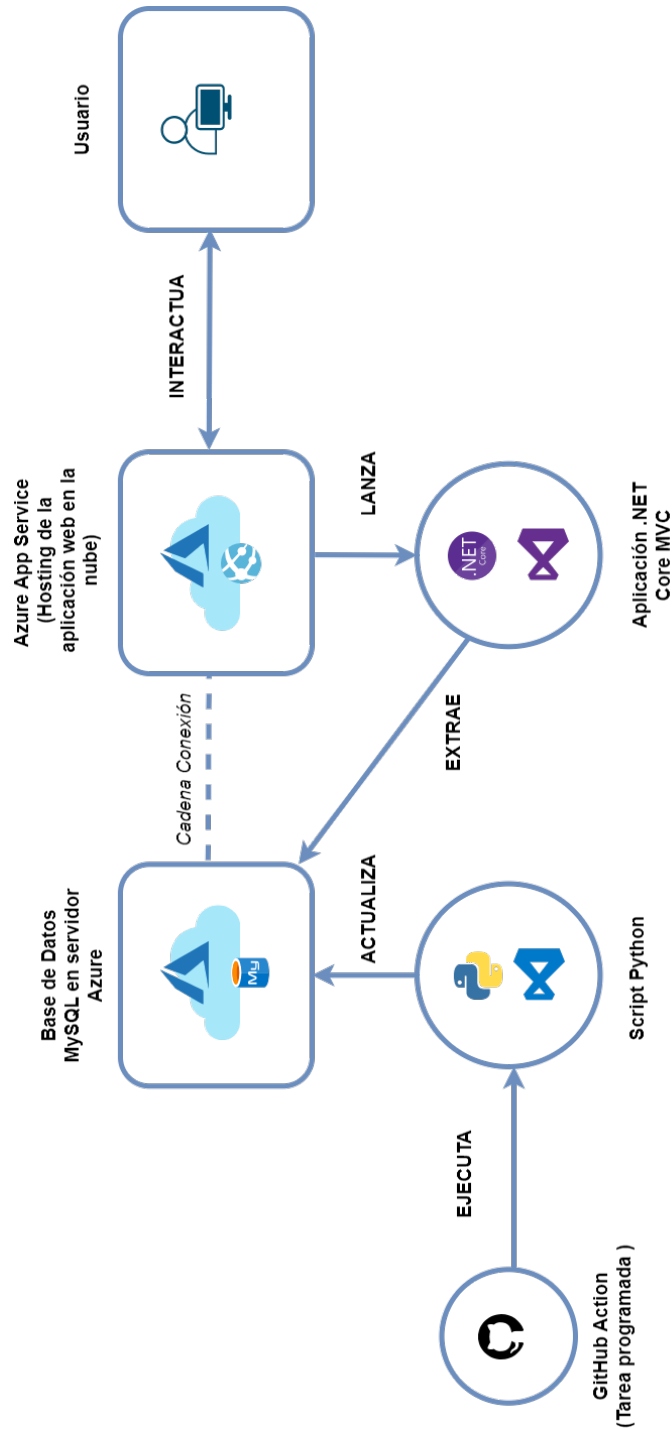


Figura 5.5: Diagrama arquitectura general del sistema.

6. Trabajos relacionados

Aunque no se disponga de información sobre si se han realizado proyectos previos sobre recopilación de convocatorias del PDI y PAS de manera centralizada y, tampoco se ha encontrado información en la web sobre su existencia. Voy a utilizar este apartado para exponer algunas aplicaciones web interesantes en las que se utiliza *web scraping* para la recopilación de datos y posteriormente mostrarlos en la web.

6.1. Indeed

Indeed es una plataforma que permite a los usuarios buscar empleos, guardarlos y solicitarlos directamente. En esta web se incluyen todas las ofertas de empleo de las principales bolsas, periódicos, asociaciones y páginas de empleo de las empresas. Todos los datos de estas ofertas son recopilados mediante *web scraping* a todas estas plataformas que se encargan de publicar empleos diariamente.

Este modelo de aplicación es un modelo muy parecido al de la aplicación del proyecto dado que se tiene el mismo objetivo: Mostrar en una misma página web las ofertas de una manera centralizada.

La principal diferencia entre la aplicación web Indeed y DeltaOffers, es que Indeed está destinada a fines más comerciales mientras que DeltaOffers lo que pretende es servir a la comunidad universitaria.

6.2. Chollometro

Chollometro es una aplicación web que se encarga de la recopilación de ofertas, cupones y promociones disponibles en tiendas *online* y físicas.

Para la recopilación de todos estos descuentos en productos, esta aplicación también realiza *web scraping* de otras webs.

Esta aplicación, al igual que DeltaOffers, se encarga de recopilar datos (en este caso sobre descuentos) para mostrarlos a los usuarios de una manera centralizada.

6.3. Google News

Google News es una plataforma la cual muestra noticias de la actualidad. Estas noticias provienen de otros periódicos o fuentes de información y Google News las muestra de manera actualizada.

Al igual que en los casos anteriores, esta página web utiliza *web scraping* para la recopilación de noticias y las muestra en su propia web de manera unificada.

7. Conclusiones y Líneas de trabajo futuras

En este apartado de van a detallar cuales son las conclusiones a destacar una vez finalizado el proyecto. Además, se va a realizar un informe crítico sobre cómo es posible mejorar el proyecto y continuar en la línea de lo realizado hasta ahora.

7.1. Conclusiones

En cuanto al resultado final del proyecto, puedo destacar que estoy enormemente satisfecho por el cumplimiento de los objetivos propuestos inicialmente. Se ha conseguido desarrollar una aplicación que muestra las convocatorias de manera centralizada y actualizada y, además se ha proporcionado una mayor funcionalidad a la misma.

Durante el transcurso de este proyecto, he podido aprender gran variedad de técnicas y herramientas que me van a ser de provecho para el futuro. Actualmente, poseo muchos más conocimientos en comparación con mis inicios en el proyecto lo cual es algo que me llena de orgullo.

Además, dado que se ha trabajado de manera autodidacta, creo que este trabajo también me ha ayudado a convertirme en una persona mucho más resolutiva y eficaz en situaciones complicadas.

Por último, me gustaría destacar la motivación que este proyecto me ha aportado para continuar mi desarrollo en el mundo del software.

7.2. Líneas de trabajo futuras

En el caso de que este proyecto continúe adelante, se van a indicar en este apartado las líneas de trabajo próximas para los futuros desarrolladores del mismo:

- Para que la aplicación responda ante las interacciones del usuario de manera más rápida, podría ser interesante un cambio de despliegue a un servidor de pago. Esto se debe a que actualmente la aplicación está desplegada en servidores gratuitos los cuales tienen unas características mucho más limitadas.
- Para aumentar el alcance de la aplicación web, sería interesante comenzar a recopilar las convocatorias de más universidades de España y empezar a mostrar esas convocatorias en la web para llegar a un mayor número de usuarios.
- En el caso de que se quisiese realizar una aplicación más profesional en Castilla Y León, se podría intentar contactar con las universidades para contar con su apoyo frente al proyecto y que las convocatorias que se empiecen a subir a sus propias webs posean un formato sólido y estructura unificada.
- Por último, sería interesante continuar con la implementación de nuevas funcionalidades en la aplicación.

Bibliografía

- [1] Arsys. Mysql. <https://www.arsys.es/blog/mysql>, 2024. [Internet; accedido 6-junio-2024].
- [2] Axarnet. Web scraping. <https://axarnet.es/blog/web-scrapping>, 2024. [Internet; accedido 5-junio-2024].
- [3] Bright Data. Beautiful soup 4. <https://brightdata.es/blog/procedimientos/raspado-de-datos-web-usando-beautiful-soup-de-python>, 2024. [Internet; accedido 5-junio-2024].
- [4] Universidad de León. Convocatorias pdi. <https://www.unileon.es/actualidad/convocatorias>, 2024. [Internet; accedido 10-junio-2024].
- [5] Universidad de Salamanca. Convocatorias pdi. <https://www.usal.es/concursos-para-la-provision-de-plazas-de-pdi-contratado-laboral>, 2024. [Internet; accedido 10-junio-2024].
- [6] Universidad de Valladolid. Convocatorias pdi. <https://pdi.uva.es/1.convocatorias/index.html>, 2024. [Internet; accedido 10-junio-2024].
- [7] Mozilla Developer. Uso de media queries. https://developer.mozilla.org/es/docs/Web/CSS/CSS_media_queries/Using_media_queries, 2024. [Internet; accedido 7-junio-2024].
- [8] Docker. Docker. <https://www.docker.com/>, 2024. [Internet; accedido 10-junio-2024].
- [9] Entity Framework. Pomelo mysql provider. <https://entityframework-extensions.net/efcore-pomelo-mysql-provider>, 2024. [Internet; accedido 6-junio-2024].

- [10] GitHub. Github actions. <https://docs.github.com/es/actions/learn-github-actions/understanding-github-actions>, 2024. [Internet; accedido 7-junio-2024].
- [11] GitHub. X.pagedlist. <https://github.com/dncuug/X.PagedList>, 2024. [Internet; accedido 6-junio-2024].
- [12] Goyache. Personal administración y servicios. <https://goyache.es/ucm/funciones-personal-universitario-ucm-pas-pdi/>, 2024. [Internet; accedido 1-junio-2024].
- [13] Goyache. Personal docente e investigador. <https://goyache.es/diversidad/personal-docente-investigador-pdi-por-que-el-docente-debe-investigar/>, 2024. [Internet; accedido 1-junio-2024].
- [14] IBM. Smtip protocol. <https://www.ibm.com/docs/es/i/7.3?topic=information-smtp>, 2024. [Internet; accedido 6-junio-2024].
- [15] Microsoft. Asp .net core mvc. <https://learn.microsoft.com/es-es/aspnet/core/mvc/overview?view=aspnetcore-8.0>, 2024. [Internet; accedido 6-junio-2024].
- [16] Microsoft. Azure app services. <https://learn.microsoft.com/es-es/azure/app-service/overview>, 2024. [Internet; accedido 7-junio-2024].
- [17] Microsoft. Azure database for mysql. <https://azure.microsoft.com/es-es/products/mysql>, 2024. [Internet; accedido 7-junio-2024].
- [18] Microsoft. Git. <https://learn.microsoft.com/es-es/devops/develop/git/what-is-git>, 2024. [Internet; accedido 2-junio-2024].
- [19] Oauth2. What is oauth2. <https://auth0.com/es/intro-to-iam/what-is-oauth-2>, 2024. [Internet; accedido 10-junio-2024].
- [20] Oracle. Mailkit tool. <https://docs.oracle.com/es-ww/iaas/Content/Email/Reference/mailkit.htm>, 2024. [Internet; accedido 6-junio-2024].
- [21] Python Org. Unittest library. <https://docs.python.org/3/library/unittest.html>, 2024. [Internet; accedido 8-junio-2024].
- [22] PyPi. Beautiful soup 4. <https://pypi.org/project/beautifulsoup4/>, 2024. [Internet; accedido 5-junio-2024].

- [23] Aprende Python. Selenium. <https://aprendepython.es/pypi/scraping/selenium/>, 2024. [Internet; accedido 5-junio-2024].
- [24] Pythonista. Guía de estilos pep-8. <https://elpythonista.com/pep-8>, 2024. [Internet; accedido 8-junio-2024].
- [25] RAE. Personal administración y servicios. <https://dpej.rae.es/lema/personal-de-administraci%C3%B3n-y-servicios>, 2024. [Internet; accedido 1-junio-2024].
- [26] RAE. Personal docente e investigador. <https://dpej.rae.es/lema/personal-docente-e-investigador>, 2024. [Internet; accedido 1-junio-2024].
- [27] Read the Docs. Requests. <https://requests.readthedocs.io/en/latest/>, 2024. [Internet; accedido 5-junio-2024].
- [28] Wikipedia. Diseño adaptativo de páginas web. https://es.wikipedia.org/wiki/Dise%C3%B1o_web_adaptable, 2024. [Internet; accedido 7-junio-2024].
- [29] Wikipedia. Github. <https://es.wikipedia.org/wiki/GitHub>, 2024. [Internet; accedido 2-junio-2024].
- [30] Wikipedia. Latex. <https://es.wikipedia.org/wiki/LaTeX>, 2024. [Internet; accedido 8-junio-2024].
- [31] Wikipedia. Microsoft azure. https://es.wikipedia.org/wiki/Microsoft_Azure, 2024. [Internet; accedido 7-junio-2024].
- [32] Wikipedia. Microsoft visual studio. https://es.wikipedia.org/wiki/Microsoft_Visual_Studio, 2024. [Internet; accedido 4-junio-2024].
- [33] Wikipedia. Scrum. [https://es.wikipedia.org/wiki/Scrum_\(desarrollo_de_software\)](https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software)), 2024. [Internet; accedido 2-junio-2024].
- [34] Wikipedia. Visual studio code. https://es.wikipedia.org/wiki/Visual_Studio_Code, 2024. [Internet; accedido 4-junio-2024].
- [35] Wikipedia. Web scraping. https://es.wikipedia.org/wiki/Web_scraping#:~:text=Web%20scraping%20o%20raspado%20web,un%20navegador%20en%20una%20aplicaci%C3%B3n., 2024. [Internet; accedido 5-junio-2024].