



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**DeltaOffers**

**Portal de convocatorias de  
PDI Y PAS de las  
Universidades de Castilla y  
León**



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

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



---

# Índice general

---

<b>Índice general</b>	<b>i</b>
<b>Índice de figuras</b>	<b>iii</b>
<b>Índice de tablas</b>	<b>v</b>
<b>Apéndice A Plan de Proyecto Software</b>	<b>1</b>
A.1. Introducción . . . . .	1
A.2. Planificación temporal . . . . .	2
A.3. Estudio de viabilidad . . . . .	11
<b>Apéndice B Especificación de Requisitos</b>	<b>19</b>
B.1. Introducción . . . . .	19
B.2. Objetivos generales . . . . .	19
B.3. Catálogo de requisitos . . . . .	20
B.4. Especificación de requisitos . . . . .	22
<b>Apéndice C Especificación de diseño</b>	<b>41</b>
C.1. Introducción . . . . .	41
C.2. Diseño de datos . . . . .	42
C.3. Diseño procedimental . . . . .	43
C.4. Diseño arquitectónico . . . . .	45
C.5. Diseño de interfaces . . . . .	51
<b>Apéndice D Documentación técnica de programación</b>	<b>53</b>
D.1. Introducción . . . . .	53
D.2. Estructura de directorios . . . . .	53

D.3. Manual del programador . . . . .	56
D.4. Pruebas del sistema . . . . .	63
<b>Apéndice E Documentación de usuario</b>	<b>65</b>
E.1. Introducción . . . . .	65
E.2. Requisitos de usuarios . . . . .	65
E.3. Instalación . . . . .	66
E.4. Manual del usuario . . . . .	66
<b>Apéndice F Anexo de sostenibilización curricular</b>	<b>77</b>
F.1. Introducción . . . . .	77
<b>Bibliografía</b>	<b>79</b>

---

# Índice de figuras

---

A.1. Sprint 1 . . . . .	3
A.2. Sprint 2 . . . . .	4
A.3. Sprint 3 . . . . .	5
A.4. Sprint 4 . . . . .	6
A.5. Sprint 5 . . . . .	7
A.6. Sprint 6 . . . . .	8
A.7. Sprint 7 . . . . .	9
A.8. Sprint 8 . . . . .	10
A.9. Sprint 9 . . . . .	10
 B.1. Diagrama de casos de uso. . . . .	23
C.1. Entidad Convocatorias . . . . .	42
C.2. Entidad Suscripciones . . . . .	43
C.3. Diagrama de flujo . . . . .	44
C.4. Patrón Modelo-Vista-Controlador [24] . . . . .	46
C.5. Inyección de dependencias . . . . .	47
C.6. Patrón MVVM . . . . .	47
C.7. Patrón Estrategia . . . . .	48
C.8. Diagrama arquitectura general del sistema. . . . .	50
C.9. Diseño Ventana Maestra . . . . .	51
C.10. Diseño Ventana Detalle . . . . .	51
C.11. Diseño Adaptativo . . . . .	52
 D.1. Ejecución Git clone . . . . .	56
D.2. Directorio del proyecto . . . . .	56
D.3. Tabla Convocatorias . . . . .	57
D.4. Tabla Suscripciones . . . . .	58

D.5. Extensión Python Visual Studio Code . . . . .	58
D.6. Variables de Entorno Python . . . . .	59
D.7. Proyecto Python Visual Studio Code . . . . .	59
D.8. Instalación dependencias . . . . .	60
D.9. Conexión Python con la base de datos MySQL . . . . .	60
D.10. Convocatorias insertadas en la base de datos . . . . .	61
D.11. Cargas de trabajo Visual Studio Community . . . . .	62
D.12. Cadena de conexión .NET . . . . .	62
D.13. Aplicación en .NET ejecutada en local . . . . .	63
D.14. Ejecución Test . . . . .	64
E.1. Ventana Inicial . . . . .	67
E.2. Visualización convocatorias . . . . .	67
E.3. Paginación . . . . .	68
E.4. Filtro Fecha Fin de Plazo . . . . .	69
E.5. Aplicar filtros . . . . .	69
E.6. Convocatoria específica, avisos y detalles . . . . .	70
E.7. Suscripción correo . . . . .	71
E.8. Correo enviado correctamente . . . . .	71
E.9. Correo no válido . . . . .	71
E.10. Límite Avisos Mensuales . . . . .	72
E.11. Correo enviado con archivo ICalendar adjunto . . . . .	73
E.12. Evento en calendario personal . . . . .	73
E.13. Ventana detalles . . . . .	74
E.14. Convocatoria en la web de una universidad determinada . . . . .	75

---

# Índice de tablas

---

A.1. Costes humanos totales . . . . .	12
A.2. Costes hardware . . . . .	13
A.3. Otros Costes . . . . .	14
A.4. Costes Totales . . . . .	14
A.5. Dependencias junto con sus Licencias Python . . . . .	16
A.6. Dependencias junto con sus Licencias .NET . . . . .	17
B.1. CU-1 Recopilación de datos. . . . .	25
B.2. CU-2 Tratamiento de datos. . . . .	26
B.3. CU-3 Estructuración de datos. . . . .	27
B.4. CU-4 Almacenamiento de datos. . . . .	28
B.5. CU-5 Actualización de los datos. . . . .	29
B.6. CU-6 Visualización de las convocatorias. . . . .	30
B.7. CU-7 Mostrar listado maestro de las convocatorias. . . . .	31
B.8. CU-8 Aplicación de filtros. . . . .	32
B.9. CU-9 Mostrar detalles. . . . .	33
B.10.CU-10 Acceso al portal de convocatorias de las universidades. . . . .	34
B.11.CU-11 Activar Avisos. . . . .	35
B.12.CU-12 Suscripción a avisos por correo. . . . .	36
B.13.CU-13 Comprobación correo valido. . . . .	37
B.14.CU-14 Recibir archivos ICalendar adjuntos. . . . .	38
B.15.CU-15 Compatibilidad ICalendar. . . . .	39



## *Apéndice A*

---

# **Plan de Proyecto Software**

---

## **A.1. Introducción**

En los años actuales, se viene dando gran prioridad a dedicar tiempo y recursos para realizar una planificación del proyecto de calidad. Esto se debe a que en todo proyecto es muy importante asentar unas bases sólidas de planificación y de estructura antes de ponernos a trabajar en los aspectos más técnicos.

Este plan de proyecto nos indicará los pasos a seguir para que el resultado de un proyecto sea satisfactorio y por lo tanto se consiga alcanzar el éxito.

Esta fase de planificación va a estar fragmentada en dos partes:

- **Planificación temporal:** Donde se detallarán los plazos que se han seguido en el proyecto y las distintas tareas y actividades realizadas en estos mismos.
- **Estudio de viabilidad:** Donde se realizará un estudio económico con su respectivo análisis de costes y beneficios y, por otro lado, un estudio legal en el que se analizarán todas las leyes que pueden llegar a estar involucradas en el proyecto.

Ambas partes deben de tenerse en cuenta a la hora de realizar un proyecto y si se realizan correctamente, estaremos en una situación ventajosa para llevar a cabo el proyecto con el menor número de contratiempos y problemas posibles.

## A.2. Planificación temporal

La gestión de este proyecto se ha llevado a cabo siguiendo una metodología Scrum [26], metodología basada en *sprints*. Durante estos *sprints* de dos semanas de duración, se acordaba una reunión retrospectiva con los tutores en la cual se hacían revisiones de las tareas realizadas durante el *sprint* y se planteaban nuevas tareas para *sprints* futuros.

Todas estas tareas abordadas, han sido recogidas dentro del apartado *Issues* en la plataforma de desarrollo colaborativo con control de versiones basado en Git (Github). Además, durante estos *sprints*, también se iban subiendo a esta misma plataforma, los distintos desarrollos realizados.

Por otro lado, cabe destacar que la metodología ágil Scrum no se ha podido realizar en su totalidad dado que es una metodología diseñada para grupos de trabajo y este proyecto ha sido un proyecto individual.

Finalmente, se van a detallar cuáles han sido los *sprints* realizados con algunas descripciones de las tareas y actividades realizadas. Además, se mostrarán una serie de gráficas con el porcentaje de tiempo dedicado en el sprint a las distintas actividades realizadas.

### Sprint 1 (08/02/2024 - 21/02/2024)

Este *sprint* comenzó con una reunión inicial con mis tutores César Ignacio García Osorio y Ana Serrano Mamolar tras haber acordado su mentorización para este proyecto previamente. En esta reunión se determinaron los objetivos del proyecto realizando una estructuración de este.

Por otro lado, también se trató el tema de las herramientas que iban a ser utilizadas para la realización del proyecto, buscando unas herramientas que se adaptasen a las tareas que se iban a realizar y al objetivo final que se quería conseguir.

Finalmente, se me dieron algunos consejos en función de las dudas que tenía sobre como estructurar y organizar el proyecto en cuanto a tiempos y a las actividades a realizar.

El trabajo realizado estas dos semanas consistió en:

- **Investigación de las webs de las que voy a extraer información mediante *web scraping*:** Para determinar todos los atributos candidatos a recopilar para mostrarlos en la web de mi proyecto.

- Determinar las bibliotecas de Python que van a ser utilizadas para llevar a cabo el proceso de *web scraping*

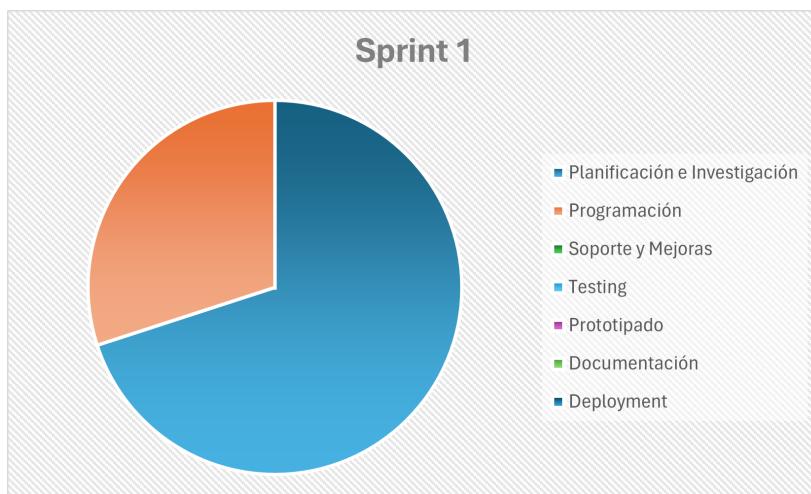


Figura A.1: Sprint 1.

## Sprint 2 (22/02/2024 - 06/03/2024)

Durante esta segunda iteración se realizó el proceso de *web scraping* para las universidades públicas de Castilla y León. Esta fue una primera toma de contacto con el *scraping* en la que se obtuvieron los atributos de cada una de las convocatorias y para cada una de las universidades.

Para este proceso se utilizaron herramientas como Requests [20], Selenium [14] y BeautifulSoup [2]. Durante este *sprint* se fue aprendiendo la utilización de estas herramientas mediante la lectura de documentación y su aplicación práctica.

Además, dado que la calidad de algunos datos cuando se realizó el *scraping* no eran buenos, se decidió realizar la limpieza de algunos de ellos ya pensando en la calidad de mi web en el futuro.

Por último, se realizó una subida al repositorio del proyecto en GitHub con los desarrollos realizados durante este *sprint*.

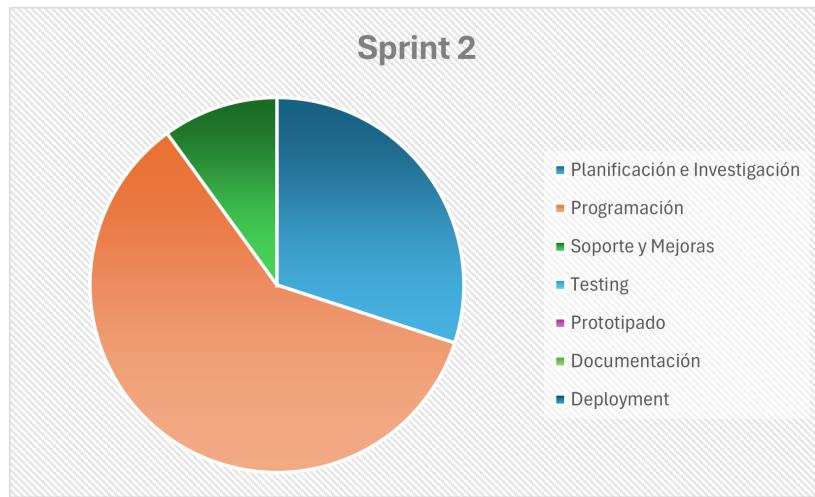


Figura A.2: Sprint 2.

### Sprint 3 (07/03/2024 - 20/03/2024)

Por un lado, este periodo consistió en añadir funcionalidad nueva al proceso de *web scraping* mediante la explotación de nuevos atributos los cuales no habían sido contemplados anteriormente.

Sin embargo, también se buscó mucho mejorar la calidad del código realizado hasta esta fecha. Para ello se decidió pasar el desarrollo realizado a programación orientada a objetos teniendo así una estructura mucho más definida y favoreciendo de esta manera la eliminación de duplicidad de código. De esta manera, se estructuró el proyecto con una clase por cada una de las universidades de las que se trajeron los datos.

Por último, se aplicó la guía de estilos de *PEP8* para mejorar la legibilidad y calidad del código. Esto se hizo con una herramienta disponible para Python llamada Autopep8 [13] la cual nos aplicaba la guía de estilos al guardar los cambios en el desarrollo realizado.

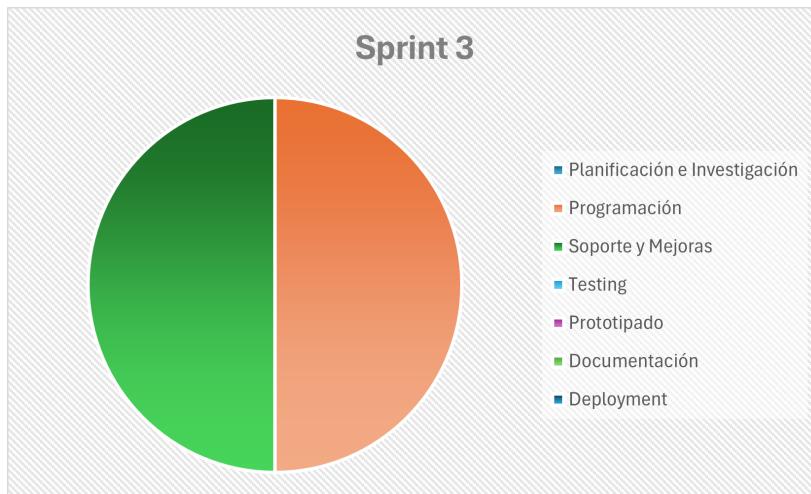


Figura A.3: Sprint 3.

### Sprint 4 (21/03/2024 - 03/04/2024)

En esta iteración, se realizó una limpieza de los datos recopilados en cuanto a su formato y se comenzó a utilizar herencia mediante la creación de una clase abstracta con los métodos comunes de las clases concretas. Este aspecto mejora enormemente la extensibilidad y facilidad de mantenimiento del código para situaciones futuras.

Gran parte del tiempo del sprint, se dedicó a la implantación de *testing*. Esto se debe a que los test unitarios en los proyectos de desarrollo de software tienen un valor crucial en la detección de errores y mejora de mantenimiento. Además, aseguran que el software tenga una cierta calidad. Para este proceso se utilizaron herramientas como MagicMock y UnitTest [17].

Por otro lado, se creó un entorno virtual en Python con *virtualenv* para la gestión de dependencias y de esta manera aislar el entorno en el que se está desarrollando el proyecto [15].

Finalmente, se creó un directorio en el proyecto con un fichero en el que se realizaban las llamadas a las correspondientes clases en las que se desarrolló el *web scraping* de cada una de las universidades para recopilar la información. Todos estos datos fueron recopilados y se realizó una conexión con la base de datos para que fuesen almacenados ahí de forma segura.

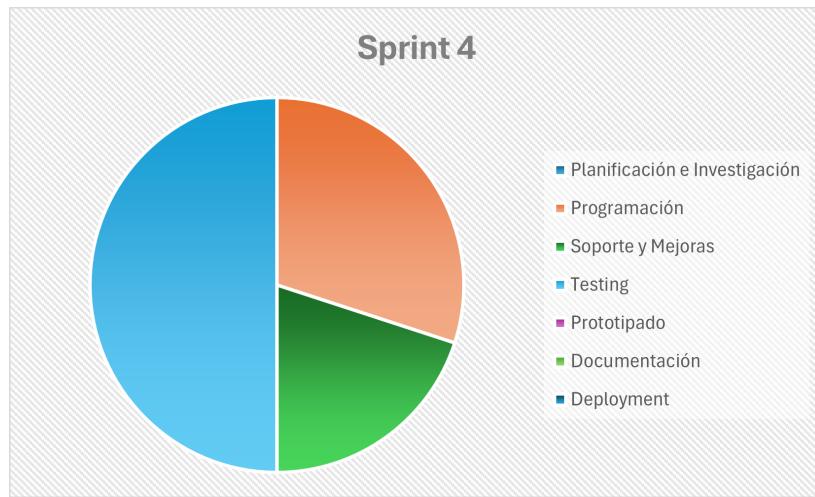


Figura A.4: Sprint 4.

## Sprint 5 (04/04/2024 - 17/04/2024)

Los primeros días de este periodo, consistieron en corregir algunos errores que se percibieron al ejecutar los test de aspectos que no se estaban realizando del todo bien en el proceso de *web scraping*. Una vez ya realizada esta tarea, se realizaron las inserciones de los datos recopilados con un formato adecuado a una base de datos MySQL [12]. Esta base de datos puede ser considerado un almacén de datos o *Data Warehouse* en el que aparecían los atributos comunes y no comunes entre las distintas webs de las distintas universidades junto con una recopilación de todas las convocatorias.

Una vez realizado esto, se comenzó con el desarrollo de la web que iba a recoger todas estas convocatorias del PDI y PAS. Para realizar esta web se decidió optar por un *framework* como Asp .NET Core MVC el cual es utilizado para realizar aplicaciones web siguiendo el patrón Modelo-Vista-Controlador [9].

Por otro lado, también se realizaron algunas tareas de *mockup* pensando en diseños atractivos para la aplicación web a realizar. En el momento que se encontraron una serie de prototipos acordes con la idea principal, se comenzó a trabajar con HTML y CSS para la realización de las interfaces.

Finalmente, se utilizaron *media queries* para que la web tuviera un diseño *responsive* y, de esta manera, que se pudiese utilizar en cualquier tipo de pantalla sin importar las dimensiones de esta.

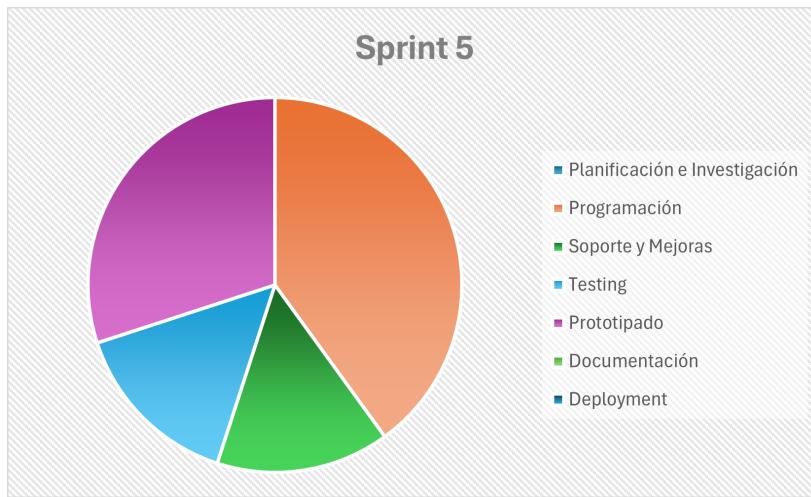


Figura A.5: Sprint 5.

## Sprint 6 (18/04/2024 - 01/05/2024)

Durante este *sprint*, se realizaron con diseño adaptativo las ventanas en las que aparecen los detalles de cada una de las convocatorias.

Posteriormente, para añadir funcionalidad extra a mi aplicación web, se decidió integrar el inicio de sesión con Google mediante autenticación basada en OAuth 2.0 [8]. Esto se realizó para que se pudiese permitir la integración con Google Calendar desde la web y de esta manera aportar un valor extra a la aplicación.

La integración con Google Calendar también se realizó en este sprint [6]. De esta manera, los usuarios que iniciasen sesión con la aplicación podrán añadir a su calendario la fecha en la que caduca esa convocatoria y así poder acordarse de realizar solicitudes en el caso de que estén interesados.

Con la finalización de este *sprint* se consideraba que la funcionalidad de la aplicación se había acabado de desarrollar en su totalidad.

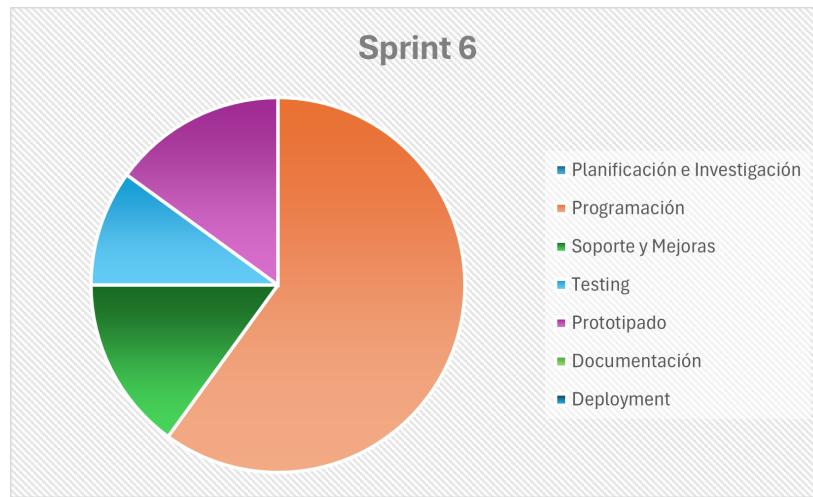


Figura A.6: Sprint 6.

### Sprint 7 (02/05/2024 - 15/05/2024)

Esta iteración, fue una de las finales del proyecto y como consecuencia, se dedicó gran parte del tiempo para realizar la documentación. Para ello, se recopilaron todas las anotaciones, datos y tareas de los anteriores *sprints*.

Por otro lado, se ha comenzado a dedicar gran parte del tiempo a la investigación sobre las distintas posibilidades que hay para lanzar la aplicación de manera gratuita para que pueda ser probada por cualquier usuario.

Por último, dado que la integración con Google Calendar solo era viable realizarla con usuarios de *testing*, se decidió eliminar esta integración e incorporar a la web un sistema de avisos por correo en la que se enviaban adjuntos archivos ICalendar para que cada usuario pudiese añadir sus convocatorias de interés como eventos a sus calendarios personales [11].

En este sprint, al igual que en la mayoría de los anteriores, también se dedicó parte de tiempo a corrección de errores y mejoras.

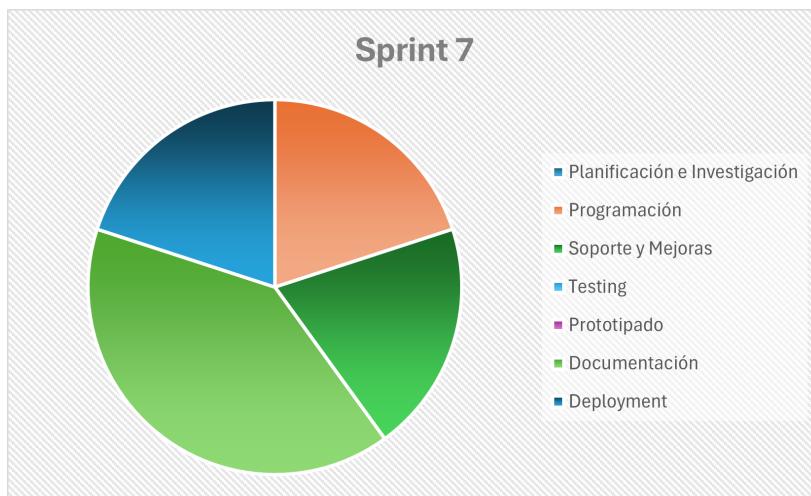


Figura A.7: Sprint 7.

## Sprint 8 (16/05/2024 - 30/05/2024)

En esta etapa, se continuó trabajando con la documentación del proyecto. Además, pese a tener algunos problemas con algunas herramientas utilizadas para hacer el *deployment*, se acabó realizando satisfactoriamente mediante la utilización de un plan gratuito de Azure Services [7].

Por otro lado, se realizó la implementación de un *cron* para mantener la base de datos actualizada. Esto lo pude hacer gracias a GitHub Actions [5].

Finalmente, también se realizó alguna mejora en el código. Hasta ahora, para la importación de paquetes se estaba trabajando con rutas relativas y para un mejor despliegue, se comenzaron a utilizar rutas absolutas. Este es uno de los cambios más destacables en cuanto a código en esta iteración.

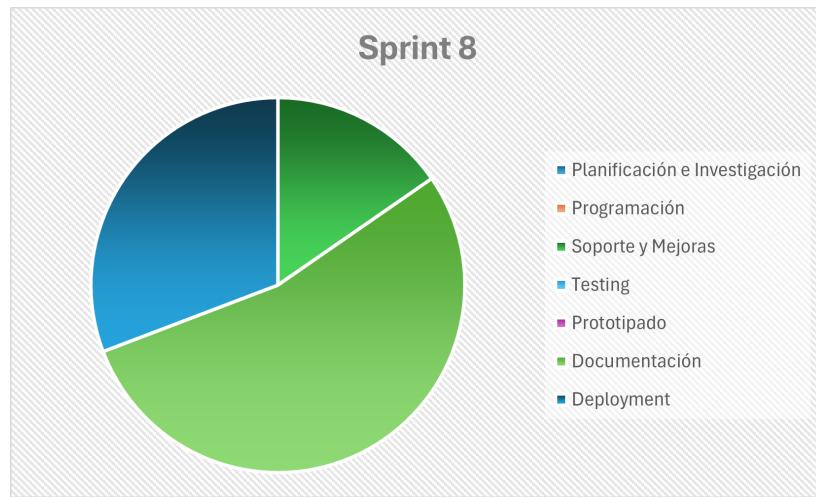


Figura A.8: Sprint 8.

### Sprint 9 (31/05/2024 - 12/06/2024)

Con esta iteración se pondría broche final al proyecto. Ya con el despliegue de la aplicación realizado, se dedicó la mayoría del tiempo a la preparación de la documentación y últimos detalles del proyecto para entregarlo al tribunal.

Se realizaron las últimas mejoras en el código, y se realizó una carga del proyecto en una máquina virtual. Esta máquina virtual se almacenará en un USB para su posterior entrega al tribunal.

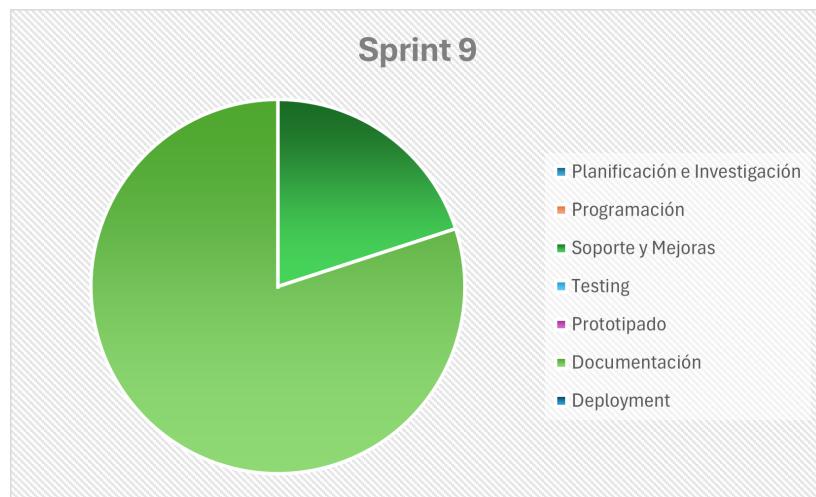


Figura A.9: Sprint 9.

## Resumen

Una vez finalizados todos los *sprints* se han podido obtener una serie de conclusiones sobre cómo ha sido la planificación del proyecto y el tiempo que se ha invertido en cada una de las distintas actividades y tareas, estas conclusiones se van a comentar a continuación:

- En las primeras iteraciones, como es lógico, se invirtió gran parte del tiempo a la planificación y estructuración del proyecto.
- En casi todos los *sprints* se ha dedicado un porcentaje de tiempo a la programación y a la realización de nuevos desarrollos.
- El soporte y las mejoras de la aplicación, fue algo frecuente en gran parte de las iteraciones gracias a la implementación de *testing* y realización de test manuales mediante el uso de la aplicación, se pudieron corregir los errores.
- Una vez que acabaron los desarrollos en cuanto a la extracción de datos, se dedicó más tiempo al prototipado y al desarrollo de las interfaces de la aplicación.
- Los últimos tramos del proyecto fueron dedicados a la documentación de la aplicación y a su lanzamiento.

## A.3. Estudio de viabilidad

### Viabilidad económica

#### Introducción

En este apartado se va a realizar un análisis de los costes y los beneficios que tendría este proyecto. Estos factores serán muy importantes en el caso de que este proyecto se ejecutase en un entorno real. De esta manera podríamos examinar su viabilidad financiera e incluso apreciar cual sería el impacto que tienen los costes y beneficios a la hora de tomar decisiones.

Por otro lado, dado que el proyecto no se ha llevado a cabo en un entorno real, podría haber otros factores que afecten a la viabilidad económica del mismo y que no hayan sido detallados en este apartado.

## Análisis de los Costes

En este apartado, podemos identificar tres categorías de costes que casi todo proyecto software se encuentra en el camino: los costes humanos, los costes de hardware y los costes de software.

### **Costes humanos:**

Este proyecto ha sido realizado por una sola persona. Considerando que ese empleado es un Ingeniero Informático cuyo país de residencia es España, según el portal *Talent.com* al que puedes acceder desde el siguiente [enlace](#), el salario promedio por hora es de 13,86 € [19]. Contando que se han dedicado 300 horas, podemos deducir lo siguiente:

$$13,86 \text{ €} \times 300 \text{ horas} = 4158 \text{ € brutos totales}$$

Esta cantidad la tenemos que dividir entre los 4 meses en los que se ha realizado en proyecto y por lo tanto:

$$4158 \text{ €}/4 \text{ meses} = 1039,5 \text{ € brutos mensuales}$$

Una vez analizado lo que va a recibir el trabajador, vamos a analizar cuáles serían, como consecuencia, los costes totales de personal que tendrá la empresa [4]:

Concepto	Coste
Salario mensual bruto	1039,5€
Contingencias comunes (23,6 %)	245,32€
Tipo general de desempleo (5,5 %)	57,17€
Contingencias profesionales (3,5 %)	36,38€
Fondo de garantía social (0,2 %)	2,08€
Formación profesional (0,6 %)	6,24€
Salario mensual total	1386,69€
<b>Total costes humanos</b>	<b>5546,76€</b>

Tabla A.1: Costes humanos totales

### Costes hardware:

Para el desarrollo del proyecto, tan solo es necesario poseer de un ordenador portátil. En este caso, suponemos que el ordenador ha sido comprado nuevo por la empresa para la realización de este nuevo proyecto. Contando con que el proceso de *web scraping* es costoso en cuanto a rendimiento, se necesitará un ordenador con unas características medianamente potentes, por lo tanto, el coste hardware será el siguiente:

Concepto	Coste
Ordenador Portátil	879,99€
<b>Total costes hardware</b>	<b>879,99€</b>

Tabla A.2: Costes hardware

### Costes software:

En cuanto a los costes software, tan solo podríamos recalcar el sistema operativo utilizado, pero en este caso, la licencia venía con el ordenador portátil de fábrica. Este sistema operativo era el Windows 10 Pro.

Por otro lado, me gustaría destacar que todas las herramientas software utilizadas son gratuitas. Esto es un factor muy a tener en cuenta a la hora de realizar un estudio económico del proyecto dado que reduce enormemente los costes en comparación con otros proyectos.

**Otros costes:** Aunque este tipo de costes no estén encasillados en una categoría como las definidas previamente. Un proyecto de este estilo suele necesitar de otros productos o servicios que también son imprescindibles para que el proyecto tenga éxito.

Los productos y servicios mencionados previamente se consideran los siguientes:

- **Infraestructura de alojamiento y dominio:** Dado que para este tipo de proyectos es necesario disponer de instancias de computación en la nube y un dominio personalizado.
- **Conexión a internet:** Debido a que a pesar de que el software sea gratuito, se necesita en ocasiones conexión con la red para poder ejecutarlo.

- **Documentación impresa y material para la presentación:** Da do que es necesario por si se quieren tener reuniones con posibles compradores del producto o para presentarlo dentro de la empresa en la que se ha realizado el proyecto. En este caso esto se ha realizado para poder presentarlo ante el tribunal.

Concepto	Coste
Infraestructura y Dominio	65,55€
Internet	87,60€
Documentación y material para la presentación	50€
<b>Total otros costes</b>	<b>203,15€</b>

Tabla A.3: Otros Costes

#### **Costes totales:**

Una vez analizados todos los costes anteriores, podemos calcular cuales han sido los costes totales:

Concepto	Coste
Costes Humanos	5546,76€
Costes Software	0€
Costes Hardware	879,99€
Otros costes	203,15€
<b>Costes Totales</b>	<b>6629,9€</b>

Tabla A.4: Costes Totales

#### **Beneficios:**

El objetivo de esta aplicación ha sido aportar un servicio a las universidades de Castilla y León mediante una aplicación en la que se pudiesen ver recopiladas todas las convocatorias que estas universidades publican en sus webs.

Por esta razón, en caso de que la aplicación pasase a desarrollo, esta misma se distribuirá de manera gratuita para un beneficio de la comunidad universitaria.

Por otro lado, si se quisiese obtener un beneficio económico de la misma, se podría plantear la implantación de anuncios y publicidad. Otra opción

válida como modelo de negocio, sería el cobro de un importe para aquellos usuarios que hayan accedido a una convocatoria gracias a esta web y hayan sido elegidos para el puesto solicitado.

## Viabilidad legal

Al igual que la viabilidad económica, la viabilidad legal también es un aspecto muy importante para considerar a la hora de planificar un proyecto software. Este apartado va a consistir en el análisis del cumplimiento de leyes por parte del proyecto y las licencias de software.

En primer lugar, en cuanto al contenido multimedia utilizado, ha sido obtenido en distintas webs cuyos contenidos están libres de *copyright* bajo la licencia **CC0 (Creative Commons Zero)** [18]. Esto implica que ese contenido multimedia puede ser utilizado sin tener permisos debido a que no están protegidas por derechos de autor. En este caso, se ha dado alguna excepción a la hora de obtener los logos de las distintas universidades involucradas en el proyecto.

Estas imágenes han sido obtenidas a través de las webs de las propias universidades. En principio, dado que esto es un proyecto educativo, estas imágenes pueden ser utilizadas sin ningún inconveniente. Además, en el caso de que el proyecto pasase a desarrollo, no se prevé tener ningún inconveniente en este aspecto legal con las mismas dado que los beneficiarios de que esta aplicación pasase a desarrollo serían las propias universidades.

Posteriormente, se van a analizar las licencias de las dependencias utilizadas en el proyecto. Este proceso lo vamos a dividir en 2 partes. En primer lugar se van a analizar las dependencias utilizadas en cuanto al proceso de *web scraping* realizado con Python. Estas dependencias se han aparecerán listadas en el fichero *requirements.txt* del proyecto y serán las siguientes:

<b>Dependencia</b>	<b>Versión</b>	<b>Descripción</b>	<b>Licencia</b>
autopep8	2.0.4	Guía de estilos para Python	MIT
beautifulsoup4	4.12.3	Biblioteca para la extracción de datos en HTML	MIT
mysql-connector-python	8.3.0	Herramienta para conexión con la base de datos	GPL
pillow	10.3.0	Biblioteca para el tratamiento de imágenes	BSD
pytest	8.1.1	Herramienta para implementación de testing	MIT
requests	2.31.0	Biblioteca para hacer peticiones HTTP	Apache2
selenium	4.18.1	Herramienta automatización sitio web	Apache2

Tabla A.5: Dependencias junto con sus Licencias Python

Posteriormente, veremos los paquetes que han sido instalados en .NET para la realización de la web junto con sus respectivas licencias:

Dependencia	Versión	Descripción	Licencia
X PagedList Mvc Core	9.1.2	Herramienta para aplicar paginación	MIT
Pomelo Entity Framework Core MySql	8.0.2	Herramienta para la conexión con la base de datos	MIT
MySql Entity Framework Core	8.0.2	Herramienta para conexión con la base de datos	MIT
Microsoft Entity Framework Core	8.0.3	Herramienta para acceso a bases de datos	Apache2
Microsoft Entity Framework Core Tools	8.0.3	Herramienta para el trabajo con bases de datos mediante objetos	Apache2
Mailkit	4.6.0	Herramienta envío de emails	Apache2

Tabla A.6: Dependencias junto con sus Licencias .NET

Tal y como hemos podido apreciar anteriormente, la mayoría de las dependencias de este proyecto tienen una licencia de software MIT o Apache v2.0 ambas licencias son bastante permisivas y disponen de gran disponibilidad en cuanto a su uso.

Una vez evaluado lo anterior, se ha decidido que el código fuente sea público para que pueda ser modificado, estudiado y utilizado por cualquier persona sin restricciones de propiedad intelectual o de licencia. Por lo tanto, la licencia otorgada al proyecto será una licencia **MIT** esta licencia permite a los usuarios utilizar, modificar y distribuir el software sin restricciones, siempre y cuando se proporcione atribución al autor original y se incluya un aviso de derechos de autor en todas las copias del software [23].

Además, otro aspecto importante a considerar es que esta licencia nos permite compatibilidad con Apache v2.0. Algo que puede ser ventajoso de cara al futuro y nuevas implementaciones.

Mediante la elección de esta licencia se contribuye enormemente con la comunidad dado que el producto se podrá utilizar y compartir con total libertad.



## *Apéndice B*

---

# **Especificación de Requisitos**

---

## **B.1. Introducción**

La especificación de requisitos en un proyecto software consiste en describir cómo se va a comportar el sistema que ha sido desarrollado. En este apartado, se detallarán los distintos casos de uso los cuales describen las distintas acciones o actividades que va a realizar un actor para que finalmente se lleve a cabo algún proceso [22].

Tanto los requisitos como los casos de uso son muy importantes en este tipo de proyectos dado que se detallan las funciones que va a realizar el sistema de manera concisa. Además, esto mejora enormemente la comunicación entre el equipo que ha desarrollado el software y el cliente o usuarios finales. Por último, la especificación de estos requisitos cuando se inicia un proyecto ayuda a localizar los posibles inconvenientes con los que se puede encontrar el equipo de desarrollo a lo largo del mismo.

## **B.2. Objetivos generales**

En este apartado se van a comentar cuales han sido los objetivos del proyecto, algunos de ellos fueron propuestos en las fases iniciales del proyecto, sin embargo, muchos otros se han añadido después para mejorar la funcionalidad de la aplicación web y que esta sea más completa y de mejor calidad. Los objetivos en su totalidad han sido los siguientes:

- Recopilación de convocatorias de Personal Docente Investigador y Personal de Administración y Servicios de las Universidades públicas de Castilla y León de sus respectivas páginas web.
- Interpretar los datos obtenidos, tratamiento de estos y almacenarlos de manera segura.
- Proporcionar a las personas interesadas en este tipo de ofertas, una web unificada que listase todas estas convocatorias con un formato unificado. Además, esta web mejorará en usabilidad y en funcionalidad a las webs en las que las universidades publican sus convocatorias.
- Proporcionar el acceso a información más detallada de estas convocatorias junto con un enlace que te redirige a la web original.
- Permitir que las personas puedan recibir correos con un archivo adjunto para que puedan añadir las convocatorias a sus respectivos calendarios.

### B.3. Catálogo de requisitos

Una vez mencionados los objetivos, se enumerarán los requisitos funcionales y no funcionales del proyecto:

#### Requisitos funcionales:

- **RF-1 Recopilación de datos:** La aplicación tiene que ser capaz de recopilar datos de las distintas webs mediante *web scraping*.
  - **RF-1.1 Tratamiento de datos:** La aplicación debe ser capaz de tratar los datos e interpretarlos.
    - **RF-1.1.1 Estructuración de datos:** La aplicación debe estructurar los datos obtenidos para que estos tengan un formato y estructura uniforme.
  - **RF-1.2 Almacenamiento de datos:** La aplicación debe almacenar los datos recopilados de forma segura en una base de datos.
  - **RF-1.3 Actualización de los datos:** Los datos mostrados en la aplicación deben de ser actualizados periódicamente con las últimas actualizaciones en las webs de origen.
- **RF-2 Visualización de las convocatorias:** La aplicación tiene que ser capaz de mostrar la información de las distintas convocatorias.

- **RF-2.1 Mostrar listado maestro de las convocatorias:** La aplicación debe de listar en una página *maestra* la información esencial y descriptiva de cada convocatoria, esto se realizará de manera paginada.
- **RF-2.2 Aplicación de filtros:** La web tiene que permitir la aplicación de filtros para que los usuarios puedan acceder más rápido a las convocatorias de interés.
- **RF-2.3 Mostrar Detalles:** La aplicación debe de tener páginas *detalle* con los detalles de las convocatorias que el usuario selecciona.
- **RF-3 Acceso al portal de convocatorias de las universidades:** La aplicación tiene que permitir al usuario acceder mediante un enlace en la página de detalles a las páginas webs propias de cada universidad.
- **RF-4 Activar avisos:** El usuario debe poder activar avisos en las convocatorias que le interesen.
  - **RF-4.1 Suscripción a avisos por correo:** El usuario debe poder suscribirse a recibir avisos por medio del correo electrónico.
    - **RF-4.1.1 Comprobación correo válido:** La aplicación debe comprobar si el correo con el que el usuario se ha suscrito es válido.
  - **RF-4.2 Recibir avisos:** La aplicación debe enviar avisos por correo de las convocatorias que el usuario desea y el usuario deberá recibir estos.
  - **RF-4.3 Recibir archivos ICalendar adjuntos:** La aplicación debe generar un archivo ICalendar con la fecha fin de las convocatorias y adjuntarlo en el correo enviado al usuario.
    - **RF-4.3.1 Compatibilidad ICalendar con principales servicios de calendario:** La aplicación debe proporcionar un archivo de extensión ".ics" que debe de ser compatible con Outlook, Google Calendar y otros.
  - **RF-4.4 Acceso al portal de convocatorias de las universidades:** En el correo electrónico que recibe el usuario se debe permitir acceder mediante un enlace en la página de detalles a las páginas webs propias de cada universidad.
- **RF-5 Comienzo de uso de la web:** La aplicación debe mostrar una ventana principal con información de la temática de la web junto con un acceso para comenzar su uso.

**Requisitos no funcionales:**

- **RNF-1 Rendimiento:** Los tiempos de respuesta son aceptables contando que se está utilizando un servidor web gratuito. La aplicación siempre responde ante las peticiones correctamente, aunque el tiempo de respuesta en ocasiones no sea inmediato.
- **RNF-2 Seguridad:** Los datos obtenidos serán almacenados de manera segura sin que se pueda acceder de manera no autorizada.
- **RNF-3 Usabilidad:** La experiencia del usuario al interactuar con el sistema es buena. La web es intuitiva y fácil de utilizar. La curva de aprendizaje es muy pequeña.
- **RNF-4 Disponibilidad:** La aplicación web está disponible para los usuarios siempre y cuando tengan conexión a internet.
- **RNF-5 Mantenibilidad:** La aplicación web será fácil de mantener siempre y cuando la estructura HTML de las webs de las que se extraen los datos sea la misma. En el caso de que no, la web necesitará de soporte. La implementación de test unitarios facilita enormemente este aspecto.
- **RNF-6 Escalabilidad:** La estructura del proyecto está diseñada para que la aplicación sea fácilmente escalable gracias a la aplicación de buenas prácticas y patrones de diseño.
- **RNF-7 Portabilidad:** La página web puede ser accedida desde cualquier navegador. Además, tiene un diseño adaptativo para que sea compatible con multitud de dispositivos independientemente del ancho de la pantalla.

## B.4. Especificación de requisitos

En este apartado se mostrará un diagrama de los casos de uso basándonos en los requisitos previamente mencionados. Posteriormente, se detallarán cada uno de estos.

## Diagrama de casos de uso

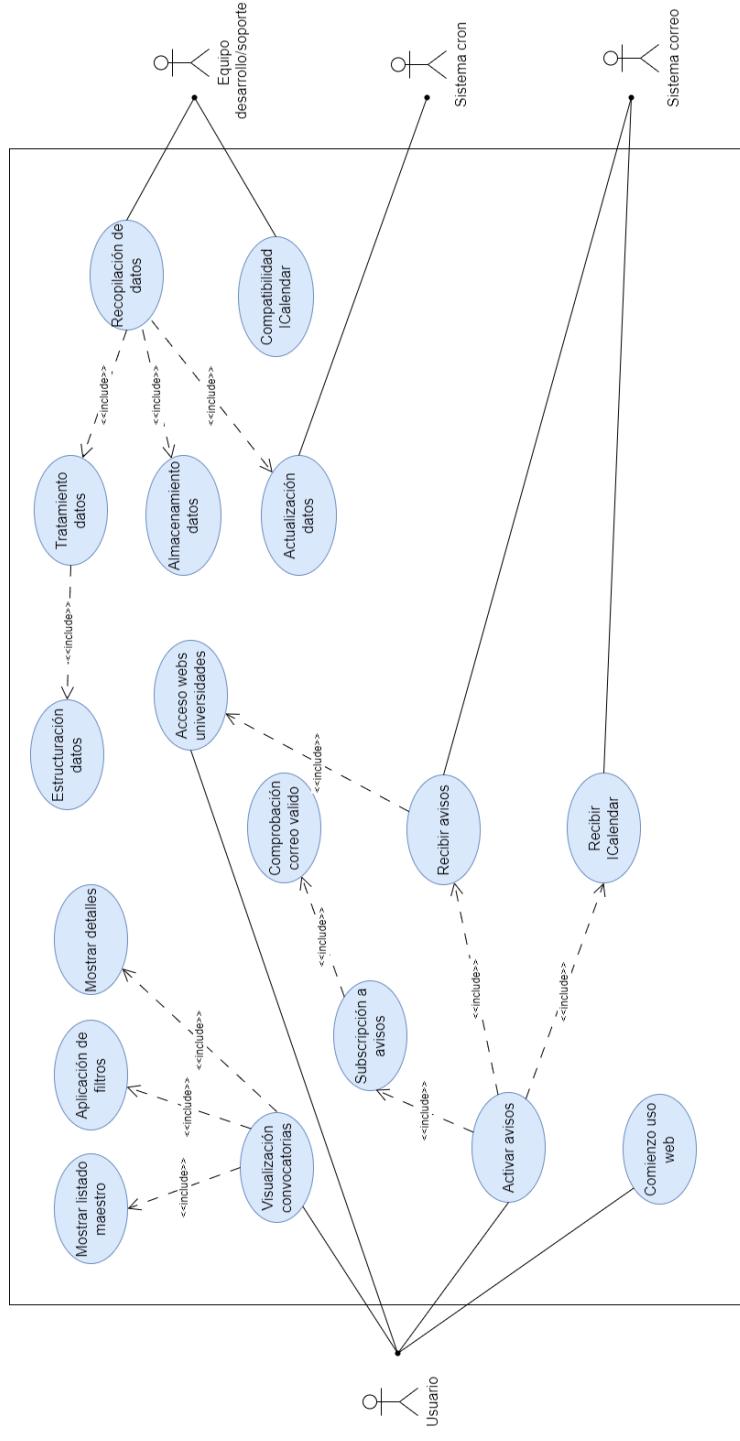


Figura B.1: Diagrama de casos de uso.

## Actores

Los actores en un diagrama de casos de uso, son aquellos que desempeñan alguna función dentro del sistema. Los actores no tienen por qué ser una sola persona sino también un equipo o un sistema externo. En el caso de nuestro sistema, hemos encontrado gran diversidad en cuanto a los actores:

- **Usuario:** Son aquellos que van a utilizar la aplicación y dispondrán de toda la funcionalidad de esta. La aplicación está diseñada para que el usuario pueda interactuar con ella de una manera sencilla y con una buena experiencia de usuario.
- **Equipo desarrollo/sopporte:** En este proyecto, los desarrolladores y el equipo de soporte, también realizan acciones que impactan en el sistema. Esto se debe a que, dado que nuestra web está constantemente recopilando datos de webs externas, en el caso de que nuevas convocatorias de nuevas universidades se quieran mostrar en la web o que cambie la estructura de la actual web de origen, el equipo de desarrollo y soporte tendrán que realizar labores de tratamiento de datos, estructuración y almacenamiento.
- **Sistema cron:** Este sistema ha sido el encargado de actualizar la base de datos periódicamente según se iban actualizando las convocatorias en las webs de las propias universidades. De esta manera, la información en la web del proyecto, estaba actualizada constantemente.
- **Sistema de correo:** Por último, otro de los actores involucrados ha sido el sistema de correo. Este ha permitido el envío de avisos a los usuarios que van a utilizar la web. En este caso, el sistema utilizado ha sido el servidor de correo electrónico SMTP, el cual permite el envío de emails de manera automatizada.

<b>CU-1</b>	<b>Recopilación de datos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Miguel Ubierna Gutiérrez
<b>Requisitos asociados</b>	RF-1, RF-1.1, RF-1.1.1, RF-1.2, RF-1.3,
<b>Descripción</b>	La aplicación deberá disponer de las convocatorias que las universidades muestran en sus propias webs.
<b>Precondición</b>	Las webs de las correspondientes universidades deberán estar disponibles.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El equipo de desarrollo examinará la estructura de las webs para su extracción de datos.</li> <li>2. Recopilará los datos más interesantes.</li> <li>3. Se tratarán los datos recopilados y se almacenarán los mismos.</li> <li>4. Los datos se mostrarán de manera actualizada en la web mediante el sistema <i>cron</i>.</li> </ol>
<b>Postcondición</b>	La base de datos será actualizada.
<b>Excepciones</b>	Fallo en las solicitudes HTTP a las correspondientes webs.
<b>Importancia</b>	Alta.

Tabla B.1: CU-1 Recopilación de datos.

<b>CU-2</b>	<b>Tratamiento de datos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Miguel Ubierna Gutiérrez
<b>Requisitos asociados</b>	RF-1.1, RF-1.1.1
<b>Descripción</b>	La aplicación debe ser capaz de tratar los datos, para que solo sean mostrados aquellos que sean descriptivos y de interés.
<b>Precondición</b>	Que se hayan podido recopilar los datos correctamente.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. Se obtienen todos los datos.</li> <li>2. Se detectan los datos no válidos o erróneos</li> <li>3. Se detectan los datos que no son de interés.</li> <li>4. Se trabaja con los datos de interés</li> </ol>
<b>Postcondición</b>	Ninguna.
<b>Excepciones</b>	Ninguna.
<b>Importancia</b>	Alta.

Tabla B.2: CU-2 Tratamiento de datos.

<b>CU-3</b>	<b>Estructuración de datos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Miguel Ubierna Gutiérrez
<b>Requisitos asociados</b>	RF-1.1.1
<b>Descripción</b>	La aplicación debe ser capaz de tratar los datos, para que después estos sean mostrados con una estructura y formato correctos.
<b>Precondición</b>	Se debe disponer de los datos de interés.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. Se obtienen los datos de interés.</li> <li>2. Se estructuran los mismos por categorías o importancia.</li> <li>3. Se corrigen las divergencias en cuanto a los formatos.</li> <li>4. Se almacenan los mismos de manera correcta.</li> </ol>
<b>Postcondición</b>	La base de datos debe estar disponible.
<b>Excepciones</b>	Ninguna.
<b>Importancia</b>	Media.

Tabla B.3: CU-3 Estructuración de datos.

<b>CU-4</b>	<b>Almacenamiento de datos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Miguel Ubierna Gutiérrez
<b>Requisitos asociados</b>	RF-1.2
<b>Descripción</b>	La aplicación debe ser capaz de almacenar los datos de interés que ya tienen el formato adecuado.
<b>Precondición</b>	Disponer de una base de datos.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. Se configura un servidor para almacenar la base de datos.</li> <li>2. Se crean las bases de datos en el servidor.</li> <li>3. La aplicación se conecta con el servidor para transferir sus datos al mismo.</li> <li>4. Se almacenan los datos de manera correcta.</li> </ol>
<b>Postcondición</b>	El servidor debe estar disponible.
<b>Excepciones</b>	El servidor no está disponible, no se cargarán las convocatorias.
<b>Importancia</b>	Alta.

Tabla B.4: CU-4 Almacenamiento de datos.

<b>CU-5</b>	<b>Actualización de los datos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Miguel Ubierna Gutiérrez
<b>Requisitos asociados</b>	RF-1.3
<b>Descripción</b>	La aplicación debe ser capaz de mostrar las ofertas de manera actualizada basándose en las webs sobre las que se extraen los datos.
<b>Precondición</b>	Disponer conexión con el servidor de base de datos.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. Se configura un sistema <i>cron</i> para que los datos se actualicen de manera periódica.</li> <li>2. El sistema <i>cron</i> ejecuta determinados <i>scripts</i> de la aplicación para que la web este actualizada.</li> <li>3. Se almacenan los datos actualizados en la base de datos.</li> </ol>
<b>Postcondición</b>	Las ofertas estarán actualizadas.
<b>Excepciones</b>	El servidor no está disponible, no se cargarán las nuevas convocatorias.
<b>Importancia</b>	Alta.

Tabla B.5: CU-5 Actualización de los datos.

<b>CU-6</b>	<b>Visualización de las convocatorias</b>
<b>Versión</b>	1.0
<b>Autor</b>	Miguel Ubierna Gutiérrez
<b>Requisitos asociados</b>	RF-2, RF-2.1, RF-2.2, RF-2.3,
<b>Descripción</b>	Permite que el usuario pueda visualizar las convocatorias recopiladas de las distintas webs de las universidades.
<b>Precondición</b>	Disponer conexión con el servidor de base de datos.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. Se cargan las respectivas convocatorias de las bases de datos.</li> <li>2. El usuario empieza con el uso de la aplicación.</li> <li>3. El usuario podrá ver las convocatorias.</li> </ol>
<b>Postcondición</b>	Ninguna.
<b>Excepciones</b>	El servidor que aloja la base de datos no está disponible, no se cargarán las convocatorias.
<b>Importancia</b>	Alta.

Tabla B.6: CU-6 Visualización de las convocatorias.

<b>CU-7</b>	<b>Mostrar listado maestro de las convocatorias</b>
<b>Versión</b>	1.0
<b>Autor</b>	Miguel Ubierna Gutiérrez
<b>Requisitos asociados</b>	RF-2.1
<b>Descripción</b>	Permite al usuario visualizar un listado maestro de las convocatorias con los atributos más descriptivos de las mismas.
<b>Precondición</b>	Disponer de conexión con el servidor de base de datos.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la página maestra de la aplicación.</li> <li>2. Se cargan las respectivas convocatorias de las bases de datos.</li> <li>3. Se mostrará un listado paginado de las convocatorias con los atributos más descriptivos de las mismas como la categoría, plazo y universidad.</li> </ol>
<b>Postcondición</b>	Ninguna.
<b>Excepciones</b>	El servidor que aloja la base de datos no está disponible, no se cargarán las convocatorias.
<b>Importancia</b>	Alta.

Tabla B.7: CU-7 Mostrar listado maestro de las convocatorias.

<b>CU-8</b>	<b>Aplicación de filtros</b>
<b>Versión</b>	1.0
<b>Autor</b>	Miguel Ubierna Gutiérrez
<b>Requisitos asociados</b>	RF-2.2
<b>Descripción</b>	Permite al usuario aplicar los filtros que este desea para que encuentre de manera rápida las convocatorias de su interés.
<b>Precondición</b>	Las convocatorias deben de cargar correctamente.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. Se mostrará un listado paginado de las convocatorias con los atributos más descriptivos de las mismas como la categoría, plazo y universidad.</li> <li>2. El usuario podrá filtrar por estos campos.</li> <li>3. Una vez se pulse el botón 'Filtrar' se aplicarán los filtros.</li> <li>4. Se mostrarán las convocatorias filtradas y paginadas.</li> </ol>
<b>Postcondición</b>	Las convocatorias devueltas deben de ser solo las que cumplen el filtro.
<b>Excepciones</b>	Ninguna.
<b>Importancia</b>	Media.

Tabla B.8: CU-8 Aplicación de filtros.

<b>CU-9</b>	<b>Mostrar detalles</b>
<b>Versión</b>	1.0
<b>Autor</b>	Miguel Ubierna Gutiérrez
<b>Requisitos asociados</b>	RF-2.3
<b>Descripción</b>	Permite al usuario ver detalles de las convocatorias por las que está interesado con información adicional a la que se mostraba en el listado maestro.
<b>Precondición</b>	Las convocatorias deben de cargar correctamente.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario detecta en el listado una convocatoria que es de su interés.</li> <li>2. Accede a los detalles de la misma.</li> <li>3. Se mostrará información más detallada de la convocatoria junto con los atributos mostrados en el listado.</li> </ol>
<b>Postcondición</b>	Se deberán mostrar los detalles de la convocatoria que ha seleccionado el usuario.
<b>Excepciones</b>	Ninguna
<b>Importancia</b>	Media

Tabla B.9: CU-9 Mostrar detalles.

<b>CU-10</b>	<b>Acceso al portal de convocatorias de las universidades</b>
<b>Versión</b>	1.0
<b>Autor</b>	Miguel Ubierna Gutiérrez
<b>Requisitos asociados</b>	RF-3, RF-4.4
<b>Descripción</b>	Permite a los usuarios acceder mediante un enlace a las secciones de las webs donde las propias universidades suben sus convocatorias.
<b>Precondición</b>	Los detalles de las convocatorias deben haber cargado correctamente.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a los detalles de una convocatoria desde el listado general.</li> <li>2. Dentro de la ventana detalles habrá un enlace que redirigirá al usuario a la propia web de la universidad correspondiente.</li> <li>3. El usuario visualizará la convocatoria correspondiente y podrá ofrecerse como candidato realizando los trámites requeridos por la universidad.</li> <li>4. El usuario también podrá acceder desde el correo cuando realiza una suscripción para recibir avisos.</li> </ol>
<b>Postcondición</b>	Ninguna
<b>Excepciones</b>	Ninguna
<b>Importancia</b>	Alta

Tabla B.10: CU-10 Acceso al portal de convocatorias de las universidades.

<b>CU-11</b>	<b>Activar Avisos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Miguel Ubierna Gutiérrez
<b>Requisitos asociados</b>	RF-4, RF-4.1, RF-4.1.1, RF-4.2, RF-4.3, RF-4.3.1, RF-4.4
<b>Descripción</b>	Permite que el sistema de correo pueda enviar avisos sobre las convocatorias a los usuarios.
<b>Precondición</b>	El listado de las convocatorias ha debido de cargar correctamente.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza el listado maestro de las convocatorias.</li> <li>2. Selecciona la opción de obtener avisos en la convocatoria en la que está interesado.</li> <li>3. Si la validación es correcta, se enviará el aviso correspondiente.</li> </ol>
<b>Postcondición</b>	El email debe ser enviado al correo electrónico proporcionado.
<b>Excepciones</b>	Ventana de error en el caso de que el correo no sea correcto.
<b>Importancia</b>	Baja.

Tabla B.11: CU-11 Activar Avisos.

<b>CU-12</b>	<b>Suscripción a avisos por correo</b>
<b>Versión</b>	1.0
<b>Autor</b>	Miguel Ubierna Gutiérrez
<b>Requisitos asociados</b>	RF-4.1, RF-4.1.1
<b>Descripción</b>	Permite que el usuario se suscriba al sistema de avisos de la <i>app</i> para recibir avisos de las convocatorias que el desee.
<b>Precondición</b>	Ninguna.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario indica que quiere recibir avisos de una determinada convocatoria.</li> <li>2. El usuario deberá llenar un formulario con su email y aceptar recibir emails.</li> <li>3. Una vez realizado esto, el usuario ya estará suscrito para recibir avisos.</li> </ol>
<b>Postcondición</b>	Ninguna.
<b>Excepciones</b>	Ventana error en el caso de suscripción incorrecta.
<b>Importancia</b>	Baja.

Tabla B.12: CU-12 Suscripción a avisos por correo.

<b>CU-13</b>	<b>Comprobación correo valido</b>
<b>Versión</b>	1.0
<b>Autor</b>	Miguel Ubierna Gutiérrez
<b>Requisitos asociados</b>	RF-4.1.1
<b>Descripción</b>	La aplicación dispone de un sistema de validación de correos para asegurarse de que el usuario completa correctamente el formulario.
<b>Precondición</b>	Ninguna.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario acepta la recepción de emails para ser avisado de una determinada convocatoria.</li> <li>2. Si el email es válido, se enviará el aviso.</li> <li>3. Se mostrará una alerta en la que indica que el aviso ha sido enviado correctamente.</li> </ol>
<b>Postcondición</b>	Ninguna.
<b>Excepciones</b>	Ventana error en el caso de que el correo no sea válido.
<b>Importancia</b>	Baja.

Tabla B.13: CU-13 Comprobación correo valido.

<b>CU-14</b>	<b>Recibir archivos ICalendar adjuntos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Miguel Ubierna Gutiérrez
<b>Requisitos asociados</b>	RF-4.3
<b>Descripción</b>	El sistema de correo enviará archivos ICalendar adjuntos con la fecha de fin de las convocatorias para que los usuarios puedan añadirlo a sus calendarios personales.
<b>Precondición</b>	Ninguna.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario se suscribe a los avisos correctamente.</li> <li>2. El usuario recibe un email con el aviso y el archivo ICalendar.</li> <li>3. El archivo ICalendar puede ser abierto por el usuario y añadirlo a su calendario personal.</li> </ol>
<b>Postcondición</b>	Deberá aparecer un nuevo evento en el calendario personal del usuario.
<b>Excepciones</b>	Ninguna.
<b>Importancia</b>	Baja.

Tabla B.14: CU-14 Recibir archivos ICalendar adjuntos.

<b>CU-15</b>	<b>Compatibilidad ICalendar</b>
<b>Versión</b>	1.0
<b>Autor</b>	Miguel Ubierna Gutiérrez
<b>Requisitos asociados</b>	RF-4.3.1
<b>Descripción</b>	El archivo de extensión .ics deberá ser compatible con los principales servicios de calendario.
<b>Precondición</b>	El archivo ICalendar se ha debido de generar correctamente.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario abre el archivo con su correo.</li> <li>2. El evento se añadirá a los servicios de calendario más populares como Google Calendar o Outlook.</li> </ol>
<b>Postcondición</b>	Deberá aparecer un nuevo evento en el calendario personal del usuario.
<b>Excepciones</b>	Si el formato no es compatible, le aparecerá una excepción al usuario.
<b>Importancia</b>	Baja.

Tabla B.15: CU-15 Compatibilidad ICalendar.



## *Apéndice C*

---

# Especificación de diseño

---

## C.1. Introducción

En este apartado, se van a constatar las especificaciones de diseño de este proyecto. Estas especificaciones de diseño van acorde a los objetivos propuestos al principio del proyecto y a los requisitos funcionales y no funcionales mencionados previamente. El diseño del sistema, lo podemos dividir en tres partes:

- **Diseño de datos:** Define como se estructuran y almacenan los datos del sistema. Este proceso es muy importante para entender cómo se relacionan los datos y cómo definir una estructura ventajosa para el programa y sus correspondientes procedimientos.
- **Diseño de procedimental:** Indica los algoritmos y procedimientos utilizados en el sistema. En este proceso, suelen utilizarse los diagramas de flujo como representación gráfica de los algoritmos utilizados.
- **Diseño arquitectónico:** Que el sistema posea de una buena arquitectura, es enormemente ventajoso para que sea robusto y eficiente. En este apartado se especificará cual es la arquitectura del sistema y sus principales componentes. Además, se explicarán los patrones de diseño aplicados en el proyecto.

## C.2. Diseño de datos

Esta aplicación tan solo cuenta con dos entidades, las cuales son 'Convocatorias' y 'Suscripciones'. Se tomó esa decisión dado que con tan solo la creación de dos entidades se podían satisfacer los objetivos que tenía la aplicación.

Los atributos e información de interés de las correspondientes convocatorias se pueden encapsular en una sola tabla sin necesidad de relaciones con otras entidades lo cual podría suponer una pérdida de eficiencia cuando se acceden a los datos. A continuación, se va a mostrar la entidad junto con los atributos que tiene.

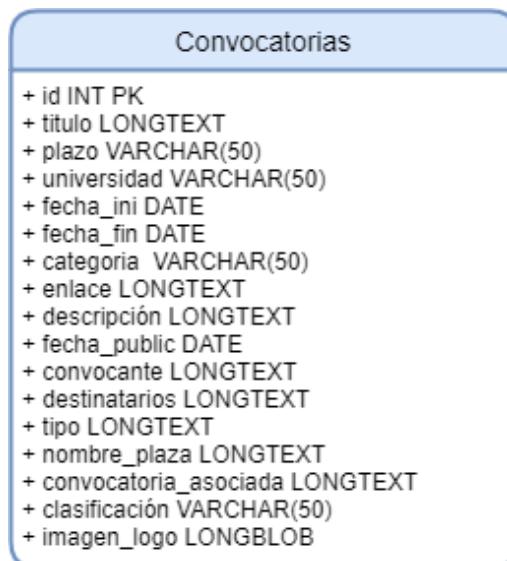


Figura C.1: Entidad Convocatorias.

Por otro lado, para evitar que la aplicación se utilizase como herramienta de spam, se limitó el número de avisos que se podían recibir por correo mensualmente. Para gestionar esto, se generó otra tabla con los emails registrados y el número de avisos que se habían enviado a ese determinado email. A continuación, se muestra gráficamente la entidad.



Figura C.2: Entidad Suscripciones.

### C.3. Diseño procedimental

En este apartado se van a detallar los procedimientos que va a realizar un usuario al utilizar la aplicación. Para ello, se va a realizar un diagrama de flujo para indicar de manera esquematizada los distintos procedimientos y flujos que el usuario puede seguir al utilizar la aplicación.

## Diagrama de flujo

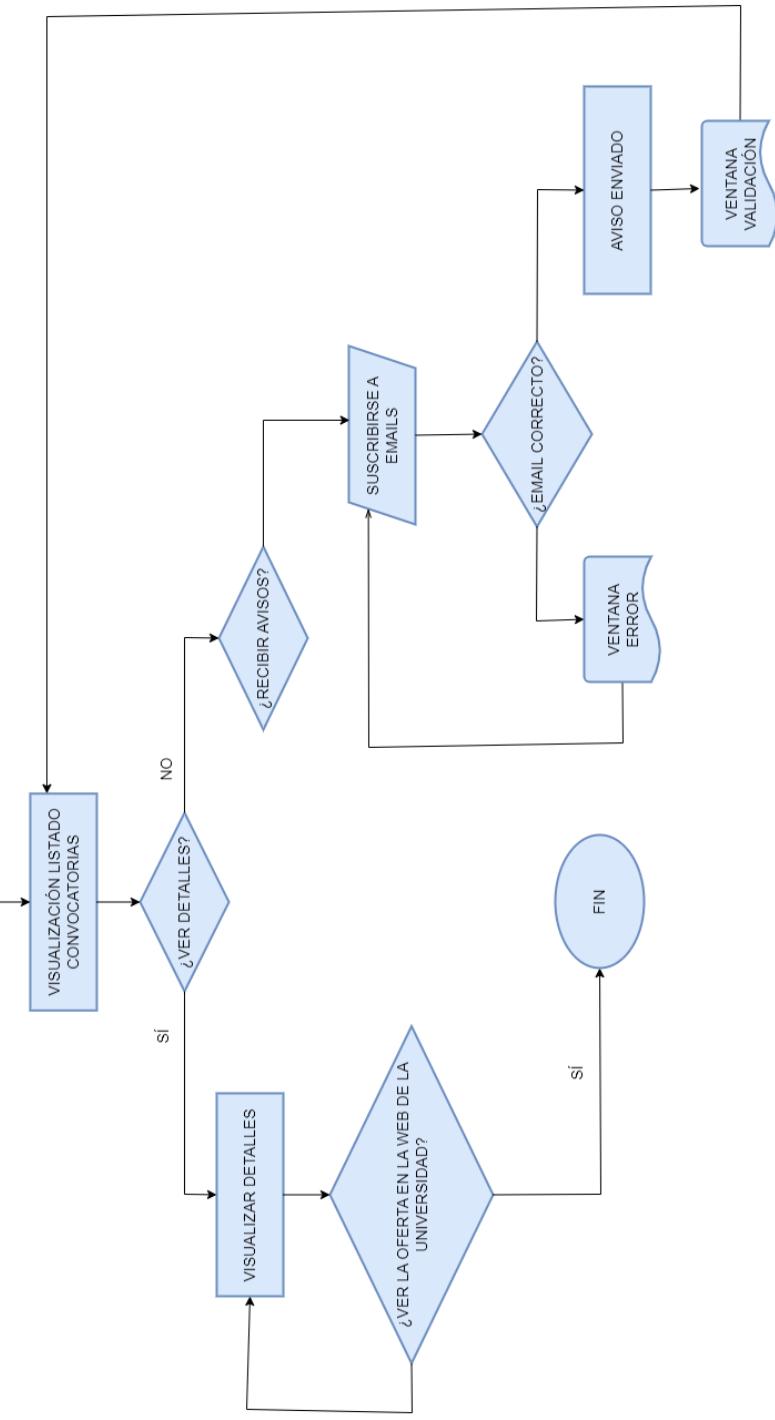


Figura C.3: Diagrama de flujo.

## C.4. Diseño arquitectónico

La arquitectura software es la encargada de diseñar la estructura que va a tener el sistema. Tener una arquitectura consistente es fundamental en aspectos de eficiencia, mantenibilidad y seguridad.

En esta fase de estructuración del sistema, juegan un papel muy importante los patrones de diseño los cuales proporcionan una mayor eficiencia y calidad de software. Además, la utilización de estos patrones mejora la escalabilidad de los proyectos permitiendo que estos puedan crecer, pero continuando con una estructura sólida.

En este apartado, se va a comentar la arquitectura general del sistema y los patrones de diseño que han sido utilizados a lo largo del proyecto. Algunos de estos patrones han sido aplicados a la parte de *web scraping* mientras que otros han sido aplicados a la hora de realizar la web. Todos los patrones utilizados, se comentarán a continuación.

### Patrones de diseño y arquitectura

#### Patrón de Arquitectura MVC (Modelo-Vista-Controlador)

Este patrón [1] se utiliza para que la interfaz de usuario, los datos y la lógica de la aplicación, trabajen de manera desacoplada. Con este patrón utilizado para aplicaciones web, las solicitudes se enrutan a un controlador que es el que trabaja con el modelo de datos y, finalmente, elige la vista que se va a mostrar pasándole el modelo de datos. Más detalladamente, las capas de este patrón consisten en lo siguiente:

- **Modelo:** en esta capa se encontrará una representación de los datos del dominio y la lógica de negocio de la aplicación con las que podemos gestionar entidades. El modelo es el que nos proporciona los datos que utilizamos en la aplicación y nos permite almacenarlos correctamente.
- **Vista:** La vista es la que se encarga de que se visualicen las interfaces de nuestra aplicación. En la vista se podrán visualizar los datos del modelo que la vista ha recibido a través del controlador. Para esta parte de la vista, se han utilizado las *Razor Pages* dado que nos permiten incluir lógica en C# dentro del HTML.
- **Controlador:** Este actúa como intermediario entre el usuario y el sistema. Este es el encargado de enviar información desde el modelo a la vista y viceversa.

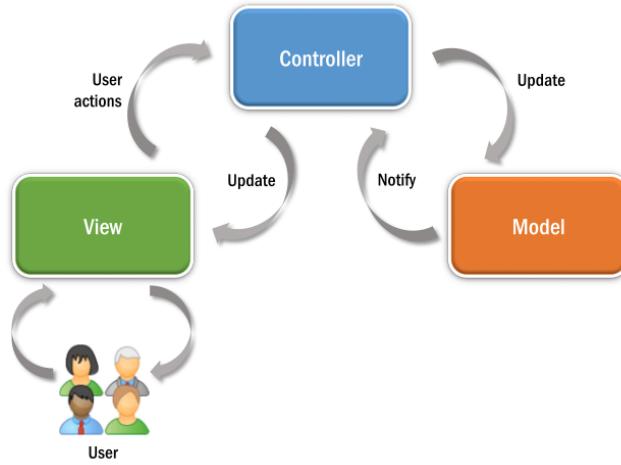


Figura C.4: Patrón Modelo-Vista-Controlador [24]

### Patrón Inyección de dependencias

Este es un patrón de diseño admitido por .NET en el que las dependencias de una determinada clase no necesitan ser creadas si no que se inyectan directamente [3]. Esto hace que las clases de alto nivel no dependan de las clases de bajo nivel.

Esto reduce el acoplamiento entre los distintos componentes de la aplicación y mejora la mantenibilidad del código.

Un ejemplo básico de utilización de este patrón en nuestra aplicación aparece cuando inyectamos el contexto de datos dentro de nuestros controladores. En lugar de generar una instancia en el propio controlador, se recibe una instancia de una determinada clase proporcionada por el contenedor de servicios donde se configuran todos los servicios que se van a utilizar en la aplicación.

```
private readonly ConvocatoriasdbContext _context;
0 references
public DetalleController(ConvocatoriasdbContext context)
{
    _context = context;
}
```

Figura C.5: Inyección de dependencias

### Patrón MVVM (Modelo-Modelo de Vista-Modelo)

Este patrón aísla la vista del modelo y el modelo de la vista. La función principal del modelo de vista en este caso es proporcionar una representación de los datos que se adapte a la vista [10]. Esto permite separar la lógica de presentación de la lógica de negocio y por lo tanto se evita que se realicen cambios importantes en el código del modelo.

En .NET este patrón se ha aplicado creando una serie de clases *ViewModel* en las cuales solo se incluían los atributos que fuesen necesarios para una vista en concreto. De esta manera se evitan realizar cambios en el modelo de datos.

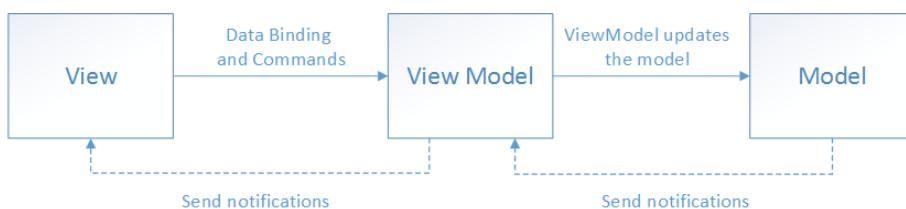


Figura C.6: Patrón MVVM

### Patrón Estrategia

Este patrón se aplicó en el proceso de web *scraping*, su principal propósito es permitir al objeto cliente elegir cuál de las estrategias le conviene.

En el caso del proyecto, se dispone una serie de algoritmos de *scraping* en función de cada universidad. Estos algoritmos estarán cada uno en clases separadas pero todos ellos implementarán una interfaz común. Esto hace que según la universidad de la que se quiera hacer el *scraping*, se selecciona su estrategia [25].

El patrón estrategia puede ser muy interesante de cara a futuro si se incluyen nuevas universidades en la aplicación dado que facilita enormemente las nuevas implementaciones y el testeo [27].

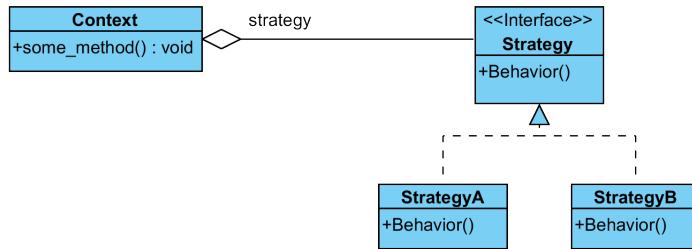


Figura C.7: Patrón Estrategia

## Arquitectura general del sistema

Para lograr el objetivo de desplegar la aplicación, se tuvo que diseñar una arquitectura en la que se utilizaron distintas herramientas para desplegar los diferentes componentes del sistema. En esta apartado, se van a detallar cada una de las partes de la arquitectura del sistema tal y como se muestra gráficamente en C.4.

En primer lugar, he de mencionar que el sistema desarrollado tiene dos partes. Por un lado, tenemos la parte del web *scraping* con Python con la que se recopilaban los datos y se actualizaba la base de datos. Por otro lado, tenemos la aplicación realizada con el *framework* .NET Core MVC, esta aplicación obtiene los datos de la base de datos actualizada mediante *web scraping* y los muestra en la web. Esta web es la que será utilizada por el usuario final.

Para el despliegue se ha desarrollado una arquitectura con tres componentes clave, las herramientas utilizadas se detallarán de manera más concreta en el capítulo 4 de la memoria. Aquí, se van a mencionar para un mejor entendimiento de la arquitectura.

- **Servidor de Azure Database para MySQL:** 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 despliegos sin realizar tareas de infraestructura.
- **GitHub Action:** 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:** 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.
- **Usuario:** Es el encargado de interactuar con la aplicación. Para ello, accederá a la URL proporcionada por el servicio de Azure.

## Diagrama arquitectura general del sistema

50

Apéndice C. Especificación de diseño

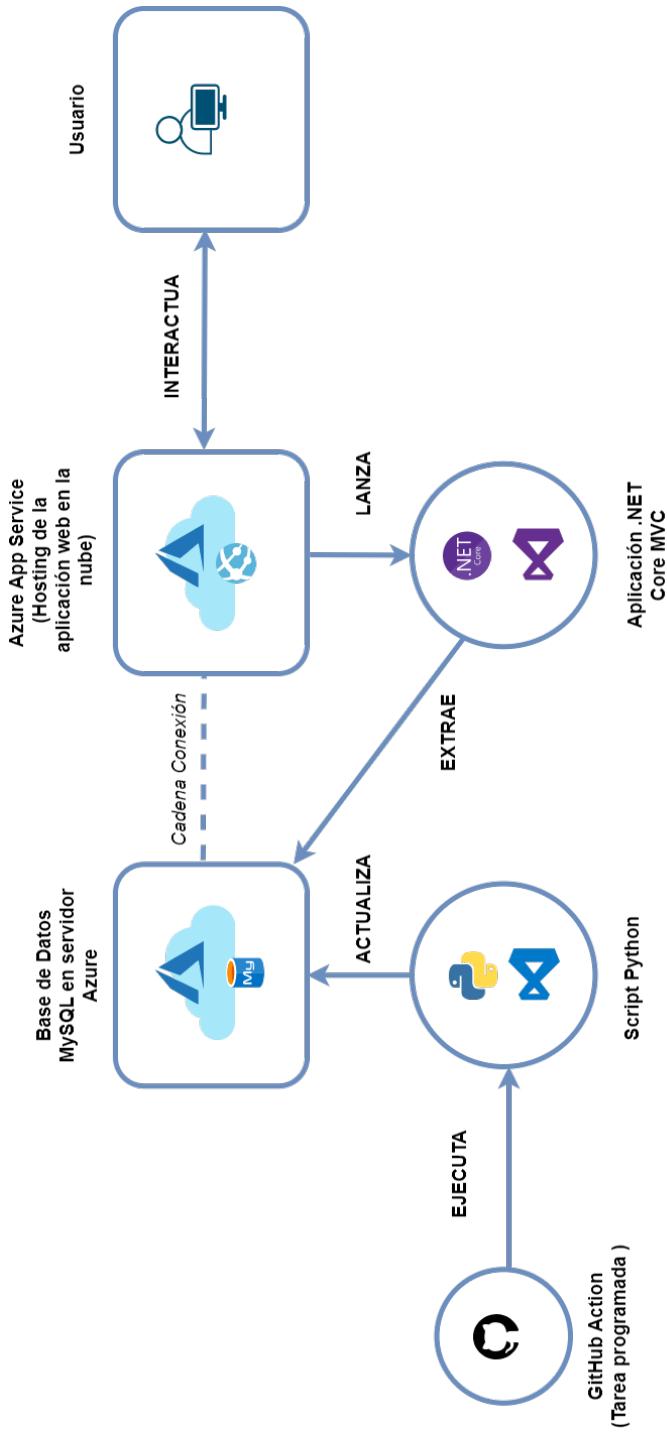


Figura C.8: Diagrama arquitectura general del sistema.

## C.5. Diseño de interfaces

Dado que no había utilizado ninguna herramienta de *mockup* recientemente y me parecía demasiado arcaico realizar diseños de interfaces a mano, decidí realizar los prototipos en sucio con PowerPoint. Esta decisión se tomó debido a la facilidad de uso que proporciona esa herramienta.

Estos prototipos se fueron creando mediante la utilización de formas, contornos, fuentes, etc. A pesar de que la herramienta utilizada no es específica para realizar los prototipos, el resultado fue satisfactorio.



Figura C.9: Diseño Ventana Maestra

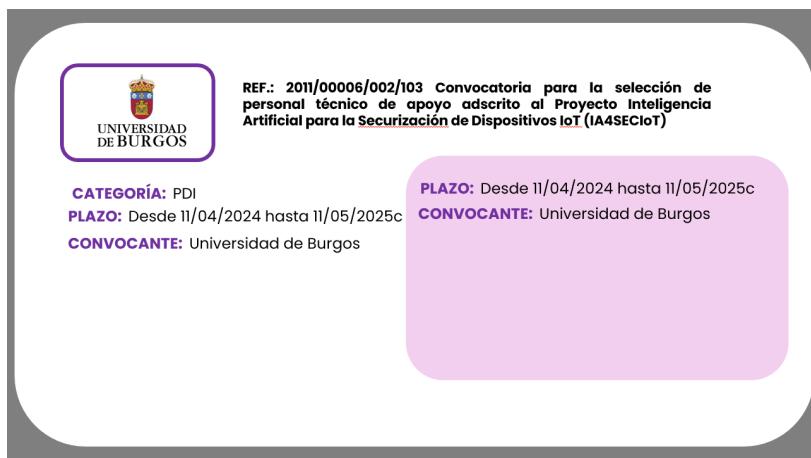


Figura C.10: Diseño Ventana Detalle

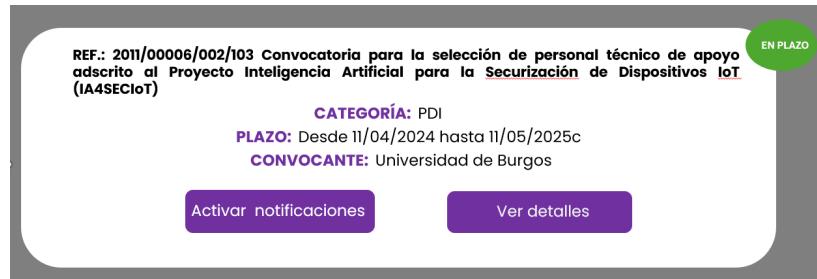


Figura C.11: Diseño Adaptativo

Estos prototipos surgieron como idea inicial. Las interfaces reales se han ido modificando a medida que se iban programando para que sean más atractivas para el usuario.

## *Apéndice D*

---

# Documentación técnica de programación

---

## D.1. Introducción

Este capítulo es fundamental para una posible evolución y mantenimiento del proyecto. En este se va a detallar la estructura de directorios del proyecto y un manual para futuros programadores del proyecto el cual incluye las directrices para realizar la compilación, instalación y ejecución de este. Además, de expondrán cuales han sido los test implementados y de qué manera se han implementado los mismos.

## D.2. Estructura de directorios

El repositorio de proyecto en GitHub, al cual se puede acceder mediante el siguiente [enlace](#), tiene los siguientes principales directorios:

- **.github**: El cual tiene en su interior un directorio **workflows** en el que aparecerán los archivos de extensión *.yml* utilizados para definir flujos de trabajo en GitHub Actions. En este caso, estos serán los encargados de actualizar la base de datos del sistema de manera periódica. La estructura de este directorio, no se volverá a explicar previamente dado que ha sido explicada en este punto.
- **DataCollections**: Este es el directorio que contiene el proyecto en Python. En él se realiza el *web scraping*, la actualización de la base de datos y las pruebas unitarias.

- **DeltaOffersWeb:** Incluye el proyecto .NET Core MVC en el que se desarrolla la aplicación web con la que el usuario podrá interaccionar y en la que se muestran las convocatorias de manera centralizada.
- **Doc:** En este directorio residirá la documentación del proyecto. Esta documentación ha sido realizada con Latex y contiene dos partes: la memoria y los anexos.

A continuación, se va a detallar cual es la estructura de directorios interna contenida dentro de los directorios principales mencionados anteriormente:

### **DataCollections:**

Estructura del directorio DataCollections:

- **Data:** Contiene los *scripts* que llaman a las clases que realizan el *scraping* y actualizan la base de datos.
- **Images:** Incluye imágenes de logos que van a ser introducidos en la base de datos.
- **Interfaces:** Directorio con interfaz de la que van a heredar las clases concretas en las que se realiza el *scraping* de cada universidad.
- **Scrapers:** Directorio en el que se encuentran las carpetas las cuales tienen en su interior las clases donde se implementa la lógica del *web scraping* de cada universidad.
  - **Scrapers/ScrapingUBU** : lógica web scraping UBU.
  - **Scrapers/ScrapingUVA** : lógica web scraping UVA.
  - **Scrapers/ScrapingULE** : lógica web scraping ULE.
- **Test:** Contiene los *scripts* de las pruebas unitarias realizadas para cada una de las universidades.

### **DeltaOffersWeb:**

Estructura del directorio DeltaOffersWeb:

- **Dependencies:** Incluye todas las dependencias utilizadas en la aplicación.

- **Properties:** En él se indican los perfiles de lanzamiento de la aplicación.
- **wwwroot:** Contiene distintos directorios dentro con los archivos estáticos utilizados. En estos archivos estáticos podemos encontrar:
  - Archivos CSS para estilos del proyecto.
  - Archivos JavaScript que incluye lógica y animaciones utilizadas.
  - Imágenes utilizadas en la web
  - Otros como bootstrap, jquery, etc
- **Controllers:** Aparecen los controladores que gestionan la lógica de negocio y que pasan datos a la vista.
- **Models:** Incluye el contexto (capa de abstracción entre la base de datos y el código) y las clases utilizadas como acceso a los datos.
- **ViewModels:** Contiene las clases utilizadas para pasar datos específicos a la vista.
- **Views:** Contiene vistas con las que interactúa el usuario. Este directorio contiene los siguientes en su interior:
  - **Detalle:** Reside el código de las vistas para las ventanas de detalles.
  - **Home:** Contiene el código de la vista inicial
  - **Universidad:** Reside el código de las vistas para las ventanas de detalles.
  - **Shared:** Contiene el código de componentes compartidos que aparecen en todas las ventanas como el encabezado de la aplicación.

## Docs:

Estructura del directorio Docs:

- **img:** Contiene las imágenes incluidas en la documentación.
- **tex:** Incluye los archivos .tex con los correspondientes capítulos de la memoria y los anexos.

### D.3. Manual del programador

En este manual se van a dar las directivas necesarias para que nuevos desarrolladores puedan trabajar con la aplicación, aportar nuevas funcionalidades o mejorar su usabilidad. Para ello se van a describir cual es el proceso de montaje del proyecto desde la instalación de herramientas hasta la ejecución del mismo.

#### Git:

En primer lugar, dado que el código del proyecto está alojado en un repositorio de GitHub, los usuarios tendrán que proceder a la instalación de Git. Esta es una herramienta utilizada para el control de versiones y que permite que diferentes usuarios puedan bajarse el código a sus equipos para trabajar con él de manera local. Para realizar esta labor una vez que tenemos Git instalado deberemos colocarnos en el directorio en el que deseamos que se aloje nuestro proyecto en local y ejecutar desde la terminal lo siguiente:

```
git clone https://github.com/miguelUbierna/DeltaOffers
```

```
deltaoffers@DeltaOffersEnvi MINGW64 ~
$ git clone https://github.com/miguelUbierna/DeltaOffers
Cloning into 'DeltaOffers'...
remote: Enumerating objects: 5314, done.
remote: Counting objects: 100% (1190/1190), done.
remote: Compressing objects: 100% (666/666), done.
remote: Total 5314 (delta 442), reused 1081 (delta 387), pack-reused 4124
Receiving objects: 100% (5314/5314), 71.32 MiB | 1.84 MiB/s, done.
Resolving deltas: 100% (982/982), done.
Updating files: 100% (618/618), done.
```

Figura D.1: Ejecución Git clone

Tal y como se puede apreciar en la imagen, el comando *git clone* irá seguido de la URL en la cual se puede encontrar el repositorio de GitHub. Posteriormente, podemos comprobar que el directorio del proyecto ya lo tenemos en nuestro equipo para trabajar con él de manera local.

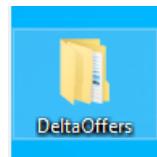


Figura D.2: Directorio del proyecto

## MySQL Server y MySQL Workbench

Este proyecto consiste en la recopilación de datos de páginas web de las universidades mediante *web scraping*. Estos datos recolectados, deberán almacenarse de manera segura y estructurada. Para lograr esto, se decidió utilizar una base de datos MySQL y por lo tanto se necesitaron instalar unas herramientas para el uso de este tipo de base de datos.

En primer lugar, el usuario debe instalar MySQL Server, esta herramienta se trata de un software gestor de bases de datos relacionales muy potente. Gracias a ella, nuestro equipo local funcionará como un servidor y gestor de bases de datos.

Posteriormente, se procede a la instalación de MySQL Workbench. Esta herramienta nos permitirá la creación de bases de datos, tablas y ejecución de consultas de una manera sencilla. Una vez instalada, se debe crear una nueva conexión con el servidor MySQL y crear la nueva base de datos junto con las siguientes tablas:

```

CREATE TABLE `convocatorias` (
  `id` int NOT NULL AUTO_INCREMENT,
  `titulo` longtext,
  `plazo` varchar(50) DEFAULT NULL,
  `universidad` varchar(50) DEFAULT NULL,
  `fecha_ini` date DEFAULT NULL,
  `fecha_fin` date DEFAULT NULL,
  `categoria` varchar(50) DEFAULT NULL,
  `enlace` longtext,
  `descripcion` longtext,
  `fecha_public` date DEFAULT NULL,
  `convocante` longtext,
  `destinatarios` longtext,
  `tipo` longtext,
  `nombre_plaza` longtext,
  `convocatoria_asociada` longtext,
  `clasificacion` varchar(50) DEFAULT NULL,
  `imagen_logo` longblob,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=286 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

```

Figura D.3: Tabla Convocatorias

```
CREATE TABLE `suscripciones` (
  `id` int NOT NULL AUTO_INCREMENT,
  `email` varchar(50) NOT NULL,
  `num_avisos` int DEFAULT '0',
  PRIMARY KEY (`id`),
  UNIQUE KEY `email` (`email`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Figura D.4: Tabla Suscripciones

Con la creación de tablas ya realizada. Tan solo se volverá a utilizar esta herramienta para comprobar que los datos recopilados mediante *web scraping* y cargados en la base de datos son los correctos.

## Instalación entorno proyecto Python

Para la realización del web scraping y la actualización de la base de datos creada anteriormente, se utilizó el lenguaje de programación Python. En esta subsección se van a detallar los pasos a seguir para poder ejecutar este proyecto y que se actualice la base de datos con los datos recopilados.

En primer lugar, es necesario un entorno de desarrollo. Como editor de código, en este caso fue utilizado Visual Studio Code y por lo tanto como paso inicial es fundamental la instalación de esta herramienta, lo cual se puede realizar a través del siguiente [enlace](#). Una vez instalado Visual Studio Code, realizaremos también la instalación de la extensión de Python disponible para este editor.

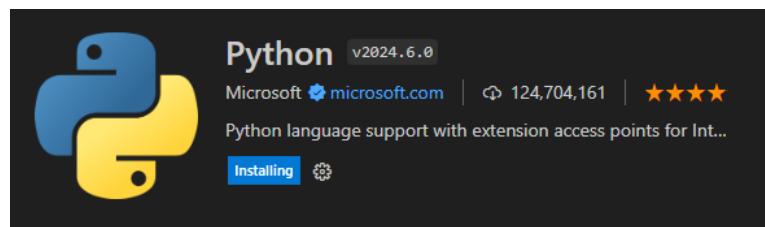


Figura D.5: Extensión Python Visual Studio Code

Posteriormente, se descargará Python desde la web oficial, concretamente la versión 3.12.2 que es la versión utilizada en el proyecto. Para la utilización de esta herramienta, será necesario modificar las variables de entorno del sistema añadiendo las siguientes rutas:

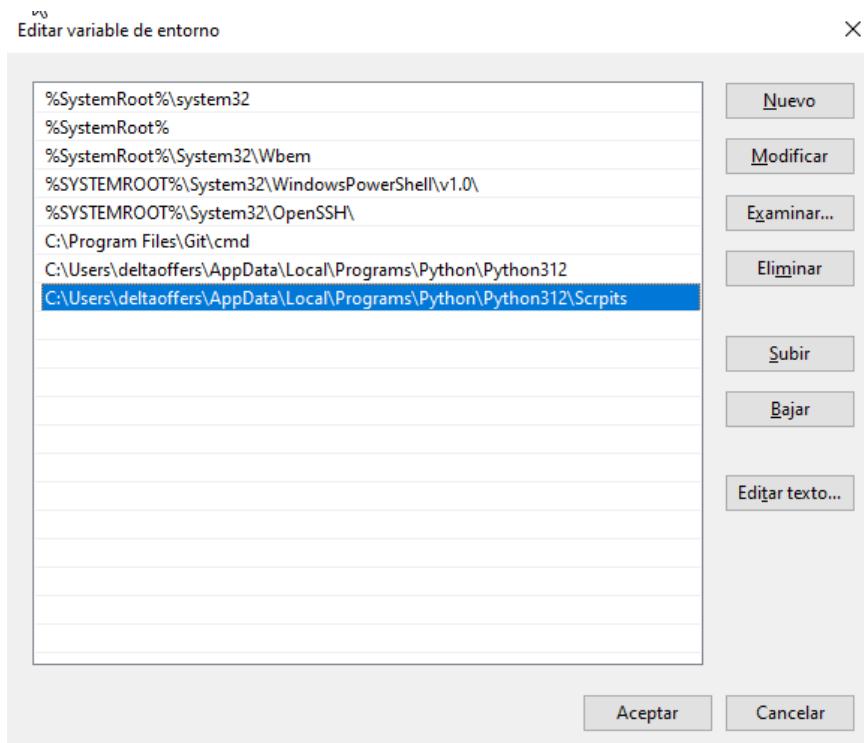


Figura D.6: Variables de Entorno Python

Finalizado este proceso, podemos abrir el proyecto en Python (el repositorio DataCollections) con esta herramienta para poder comenzar con el trabajo.

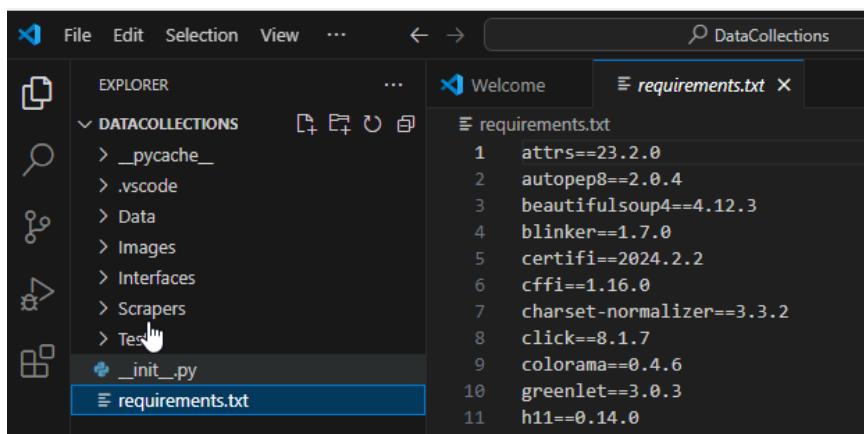


Figura D.7: Proyecto Python Visual Studio Code

Si intentamos ejecutar el proyecto, podemos apreciar que nos aparecerán errores debido a que no se han instalado los módulos necesarios. Todas estas dependencias, aparecerán en un fichero *requirements.txt* tal y como aparece en la imagen D.7, para instalarlas debemos ejecutar en la terminal de Visual Studio Code lo siguiente:

```
PS C:\Users\deltaoffers\Desktop\DeltaOffers\DataCollections> pip install -r requirements.txt
Collecting attrs==23.2.0 (from -r requirements.txt (line 1))
  Downloading attrs-23.2.0-py3-none-any.whl.metadata (9.5 kB)
Collecting autopep8==2.0.4 (from -r requirements.txt (line 2))
  Downloading autopep8-2.0.4-py3-none-any.whl.metadata (17 kB)
Collecting beautifulsoup4==4.12.3 (from -r requirements.txt (line 3))
```

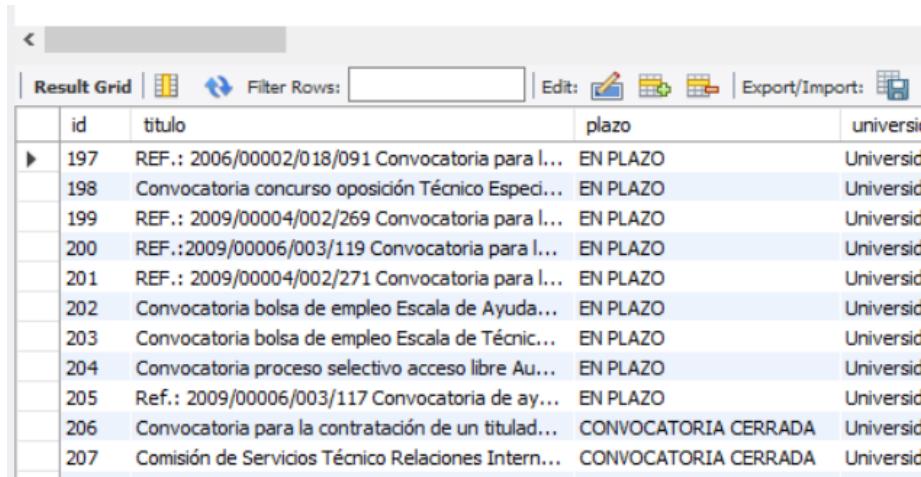
Figura D.8: Instalación dependencias

Para que al ejecutar el proyecto se actualice la base de datos, tenemos que conectarnos con la misma. Para ello, indicaremos en el código en Python, cual es el *host* al que queremos conectarnos, el usuario, la contraseña y el nombre de la base de datos.

```
conexion = mysql.connector.connect(
    host="localhost",
    user="root",
    password="tfg2024",
    database="convocatoriasdb")
```

Figura D.9: Conexión Python con la base de datos MySQL

Pare ejecutar el proyecto, iremos al fichero *data.py* haremos *click* derecho sobre él y elegiremos la opción '*Run Python/ Run Python File in Terminal*'. De esta manera se comenzará con la ejecución. Dado que el *web scraping* es un proceso de extracción de datos extenso en el que se tienen que hacer peticiones HTTP, este proceso puede tardar unos minutos. Finalizado ese tiempo podremos ver ya la base de datos MySQL actualizada con la recopilación de las correspondientes convocatorias.



The screenshot shows a database grid titled 'Result Grid' with columns: id, título, plazo, and universid. There are 10 rows of data, each representing a招ocatoria entry. The first few rows show titles like 'REF.: 2006/00002/018/091 Convocatoria para l...', 'Convocatoria concurso oposición Técnico Especi...', and 'REF.: 2009/00004/002/269 Convocatoria para l...'. The 'plazo' column contains values like 'EN PLAZO' and 'CONVOCATORIA CERRADA'. The 'universid' column consistently shows 'Universid'. Row 207 is highlighted with a blue background.

	id	título	plazo	universid
▶	197	REF.: 2006/00002/018/091 Convocatoria para l...	EN PLAZO	Universid
	198	Convocatoria concurso oposición Técnico Especi...	EN PLAZO	Universid
	199	REF.: 2009/00004/002/269 Convocatoria para l...	EN PLAZO	Universid
	200	REF.:2009/00006/003/119 Convocatoria para l...	EN PLAZO	Universid
	201	REF.: 2009/00004/002/271 Convocatoria para l...	EN PLAZO	Universid
	202	Convocatoria bolsa de empleo Escala de Ayuda...	EN PLAZO	Universid
	203	Convocatoria bolsa de empleo Escala de Técnic...	EN PLAZO	Universid
	204	Convocatoria proceso selectivo acceso libre Au...	EN PLAZO	Universid
	205	Ref.: 2009/00006/003/117 Convocatoria de ay...	EN PLAZO	Universid
	206	Convocatoria para la contratación de un titulad...	CONVOCATORIA CERRADA	Universid
	207	Comisión de Servicios Técnico Relaciones Intern...	CONVOCATORIA CERRADA	Universid

Figura D.10: Convocatorias insertadas en la base de datos

## Instalación entorno aplicación .NET

Este será el último paso que debemos de realizar para que nuestro proyecto funcione de manera local. En primer lugar, debemos realizar la instalación de Visual Studio Community, herramienta, la cual puede instalarse a través del siguiente [enlace](#).

Una vez realizado esto, en el instalador de Visual Studio, cargaremos las librerías necesarias para el desarrollo de ASP.NET y web. Para ello marcaremos la siguiente opción.

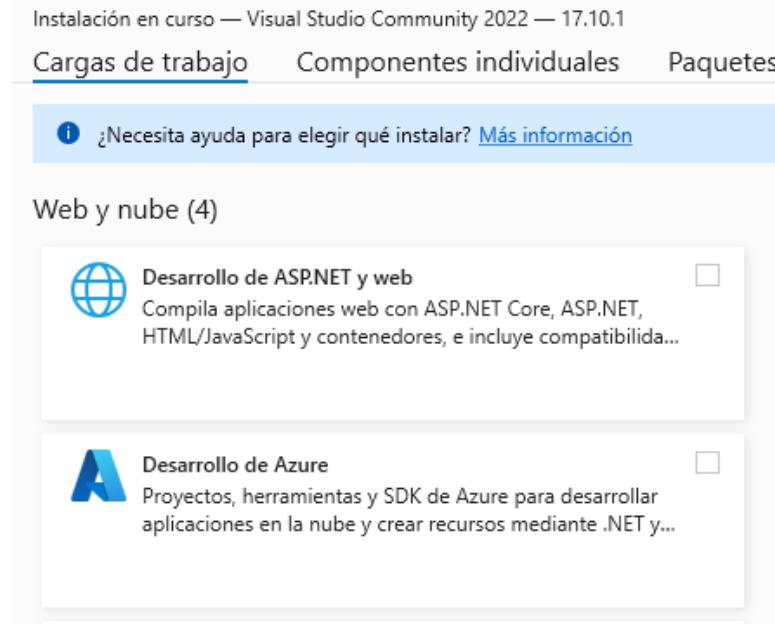


Figura D.11: Cargas de trabajo Visual Studio Community

Finalizado lo anterior, podemos abrir nuestra aplicación de Visual Studio y cargar el proyecto desarrollado en .NET Core el cual está alojado en el directorio 'DeltaOffersWeb' dentro del repositorio general del proyecto.

Una vez abierto, debemos configurar la cadena de conexión desde el fichero *AppSettings.json* para que nuestra aplicación obtenga la información de la base de datos MySQL preparada previamente. La cadena de conexión quedará definida de la siguiente manera:

```
"ConnectionStrings": {
    "MySqlConnection": "Server=localhost;Database=convocatoriasdb;User=root;Password=tfg2024;"}
```

Figura D.12: Cadena de conexión .NET

Habiendo completado los pasos anteriores, ya podremos ejecutar la aplicación. Para ello bastará con pulsar el botón '*Play*' en Visual Studio. Una vez realizado esto ya podemos ver la aplicación funcionando de manera local y estará disponible para su uso.

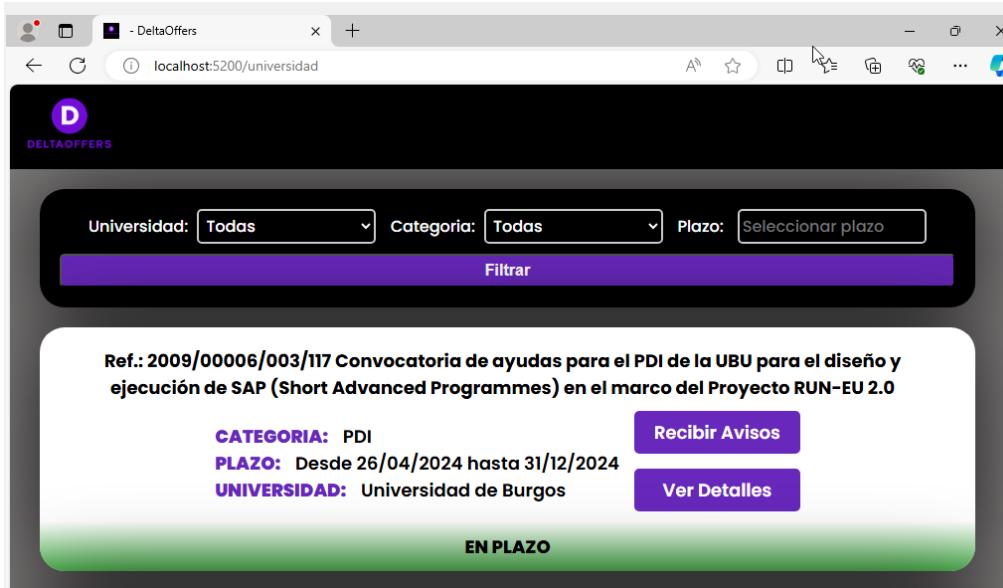


Figura D.13: Aplicación en .NET ejecutada en local

Para que el tribunal pueda probar el proyecto de manera local, se ha preparado una máquina virtual en la que se ha cargado el proyecto siguiendo los pasos previos. Esa máquina virtual será entregada al tribunal junto con la documentación y los directorios del proyecto en un USB.

## D.4. Pruebas del sistema

Para asegurarnos que el software realizado fuese de calidad, se decidieron implementar pruebas unitarias al proceso de *web scraping*. Gracias a la realización de test, pudimos percibir algunos errores que aparecían en el código de los cuales no era consciente a la hora de programar.

Estos test fueron desarrollados con Python. Para ello, utilizamos la herramienta *unittest* la cual permite la automatización de test, agregación de test en colecciones e independencia de los test de la estructura que los reporta.

Además, también se utilizó la herramienta *MagicMock* [16]. Esta es una biblioteca cuya función es simular el comportamiento de otros objetos. De esta manera evitamos la utilización de los objetos para la realización de test unitarios de manera directa.

Algunas de las pruebas que se han realizado mediante la implementación de los test son las siguientes:

- Comprobación de que el número de convocatorias obtenidas fuese el correcto.
- Comprobación de que el número de campos obtenido fuese correcto.
- Comprobación número mínimo de convocatorias
- Comprobación de que el número de convocatorias cerradas fuese correcto.
- Comprobación de que los campos tuviesen el valor y el formato deseado.

Para la ejecución de estas pruebas unitarias, en primer lugar, abriremos el proyecto en Python. Posteriormente, abriremos una nueva terminal en el proyecto y nos cambiaremos del directorio actual al directorio 'Test' . Finalmente, ejecutaremos lo siguiente:

```
PS C:\Users\deltaoffers\Desktop\DeltaOffers\DataCollections\Test> pytest -v
=====
test session starts =====
platform win32 -- Python 3.12.2, pytest-8.1.1, pluggy-1.4.0 -- C:\Users\deltaoffers\AppData\Local\Programs\Python\Python312\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\deltaoffers\Desktop\DeltaOffers\DataCollections\Test
collected 32 items
```

Figura D.14: Ejecución Test

Ejecutamos estos test con la opción `-v` para que estos se ejecuten en modo *verbose* y, por lo tanto, se nos proporcionará una mayor información de la prueba implementada y el resultado de la misma.

Por otro lado, para comprobar que la aplicación haya sido desarrollada correctamente, se implementaron pruebas manuales. Para la realización de estas pruebas, fui anotando los distintos escenarios o caminos que podría tener la web. Posteriormente, actué como '*tester*' y fui comprobando que los resultados obtenidos en las diferentes ventanas de la aplicación eran los esperados.

## *Apéndice E*

---

# **Documentación de usuario**

---

## **E.1. Introducción**

En este apartado se van a detallar cuales son los requisitos de los usuarios que utilizarán la aplicación web, las instalaciones necesarias para la utilización de la misma (que en este caso no hay ninguna) y un manual de usuario en el que se expondrán las funcionalidades que tiene la web en sus respectivas ventanas. De esta manera, el usuario no tendrá ningún inconveniente a la hora del uso de la web del proyecto.

## **E.2. Requisitos de usuarios**

Los requisitos que deben tener los usuarios para poder utilizar la web son los siguientes:

- Disponer de conexión a internet para poder acceder al dominio de la web.
- Disponer de un dispositivo de escritorio o móvil.
- Los dispositivos mencionados anteriormente deberán tener un navegador actualizado para garantizar una buena compatibilidad.
- Contar con una cuenta de correo electrónico activa para poder recibir los avisos. Este requisito no es imprescindible para el uso de la web pero si necesario si se quiere disponer de la funcionalidad de la aplicación al completo.

### E.3. Instalación

Dado que el proyecto ha consistido en realizar una aplicación web. Tan solo es necesario tener un navegador web para poder acceder a ella, este navegador web deberá estar actualizado para evitar problemas de compatibilidad.

Ninguna instalación es necesaria ya que la aplicación se encuentra desplegada en un servidor remoto y por lo tanto cualquier usuario podrá acceder a ella desde su dispositivo. El enlace completo para acceder a la web es el siguiente:

<https://deltaoffers.azurewebsites.net/>.

### E.4. Manual del usuario

En esta sección, se van a proporcionar instrucciones sobre el uso de la aplicación para que los usuarios puedan explotar sus funcionalidades al máximo.

Este manual de usuario se puede seguir en tiempo real a la vez que se utiliza la web, de esta forma el usuario entenderá el uso de la misma de manera más rápida y eficiente. La dirección de la web es aquella que se mostró en el apartado E.3.

#### Pantalla de inicio

En esta ventana inicial, el usuario será informado de cuál es el propósito de la web junto con el nombre de esta. En el momento en el que este desee empezar con la utilización de la misma, hará *click* en el botón 'Empieza ya'.

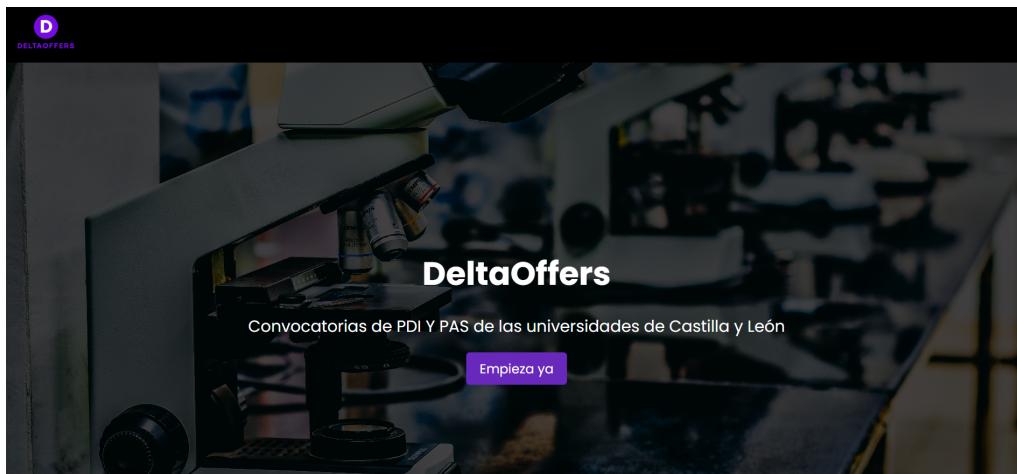
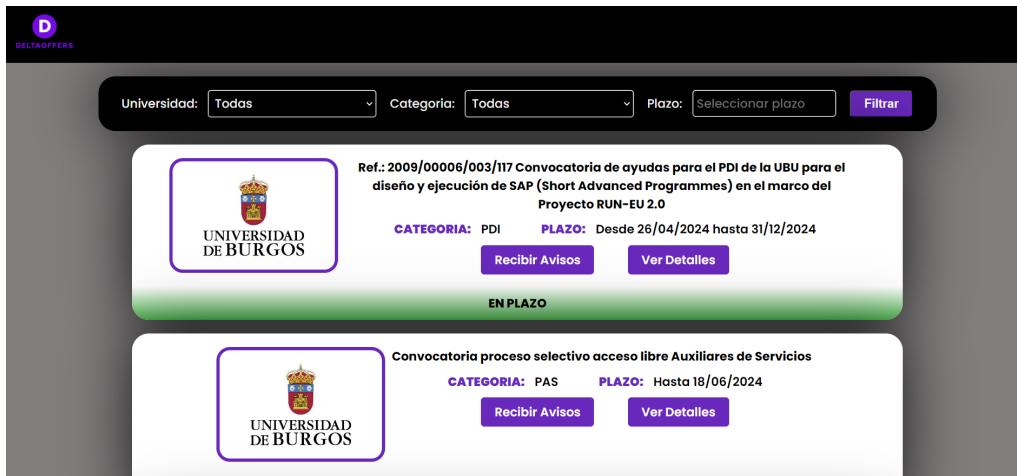


Figura E.1: Ventana Inicial

## Visualización convocatorias

Una vez pulsado el botón 'Empieza ya' el usuario visualizará un listado competo de las convocatorias. Estas convocatorias estarán ordenadas en orden descendiente por la fecha de fin de plazo. En este listado aparecerán las convocatorias en plazo actualmente y las convocatorias cerradas que se muestran más recientes en las webs de las propias universidades.



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

CATEGORÍA: PDI PLAZO: Desde 26/04/2024 hasta 31/12/2024

Recibir Avisos

Ver Detalles

EN PLAZO



Convocatoria proceso selectivo acceso libre Auxiliares de Servicios

CATEGORÍA: PAS PLAZO: Hasta 18/06/2024

Recibir Avisos

Ver Detalles

Figura E.2: Visualización convocatorias

Se podrán visualizar las correspondientes convocatorias haciendo *scroll* en la ventana de visualización y, para facilitar la navegación, se ha implementado que las convocatorias estén paginadas. El paso a la página siguiente se podrá realizar desde el final de la página actual.

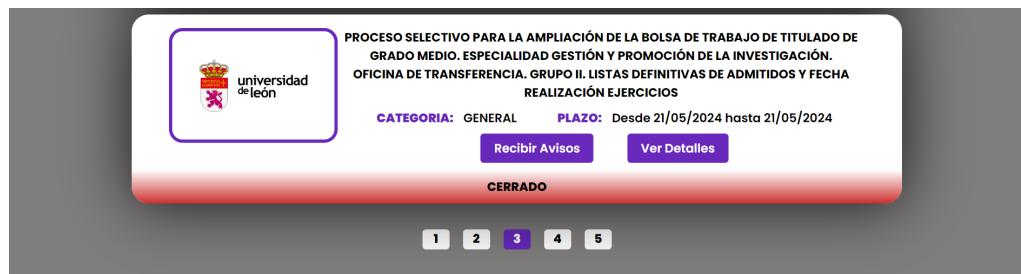


Figura E.3: Paginación

## Aplicar filtros

Cuando comenzamos con el uso de la aplicación, además de visualizarse las convocatorias, es posible aplicar filtros, de esta manera los usuarios podrán encontrar más rápido las convocatorias que sean de su interés. Los filtros aparecerán en un cuadro negro en la parte superior de la ventana.

Para aplicar los filtros se deben de seguir los siguientes pasos:

1. Elegimos la universidad por la que queremos filtrar. Si no queremos filtrar por universidad, dejamos la opción 'Todas'.
2. Elegimos la categoría deseada. En el caso de que no se quiera aplicar el filtro de categorías, dejamos marcada la opción de todas.
3. Elegimos la fecha de fin de plazo máxima, al aplicar ese filtro se nos abrirá un calendario para seleccionar la fecha seleccionada. En el caso de que no se quiera aplicar filtro de fecha, lo tenemos que dejar vacío ese *input*. Otra opción sería que hayamos filtrado por fecha y queremos eliminar el filtro, en ese caso, eliminaremos el texto que aparece en el *input* para filtrar por fecha.



Figura E.4: Filtro Fecha Fin de Plazo

4. Finalmente, debemos pulsar el botón 'Filtrar', por el contrario, si este botón no se pulsa, no se aplicarán los filtros y como consecuencia, no nos aparecerán las convocatorias que son de nuestro interés.

Convocatoria para la impartición de docencia en Xiangtan (China) dentro del Doble Grado de Diseño Mecánico, Manufactura y Automatización entre la Universidad de León y la Universidad de Xiangtan 2024-2025	
<b>CATEGORÍA:</b> GENERAL	<b>PLAZO:</b> Desde 28/05/2024 hasta 06/06/2024
<a href="#">Recibir Avisos</a>	<a href="#">Ver Detalles</a>

Convocatoria de proceso selectivo para la constitución de la bolsa de trabajo de Técnico Especialista de Laboratorio, Especialidad Mecánica (Grupo III).	
<b>CATEGORÍA:</b> GENERAL	<b>PLAZO:</b> Desde 27/05/2024 hasta 06/06/2024
<a href="#">Recibir Avisos</a>	<a href="#">Ver Detalles</a>

Figura E.5: Aplicar filtros

## Convocatoria específica

En cada convocatoria del listado, aparecerá la siguiente información:

- Título de la convocatoria.
- Universidad que oferta esa convocatoria.

- Categoría de la convocatoria.
- Plazo de la convocatoria.

Además, aparecerán dos botones los cuales nos permitirán realizar dos labores diferentes:

- Suscripción para recibir avisos de las correspondientes convocatorias.
- Ver más detalles de esa convocatoria en específico.

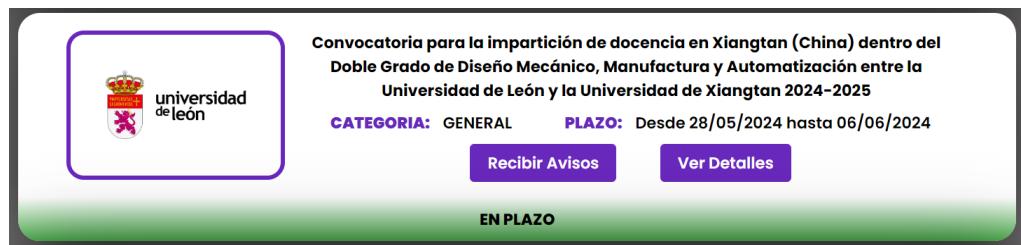


Figura E.6: Convocatoria específica, avisos y detalles

## Recibir Avisos

Una vez que se ha accedido a la sección de recibir avisos para poder recibir los mismos de una convocatoria específica, en primer lugar, nos aparecerá un formulario flotante en el que los usuarios deberán suscribirse con su correo electrónico.

Una vez se introduzca el correo de manera correcta, se pulsa el botón 'Recibir email' para recibir el aviso específico.

En el caso en el que nos aparezca el formulario flotante y queramos salir de él, tan solo tendremos que hacer *click* fuera del formulario y volveremos a la ventana de visualización de convocatorias.

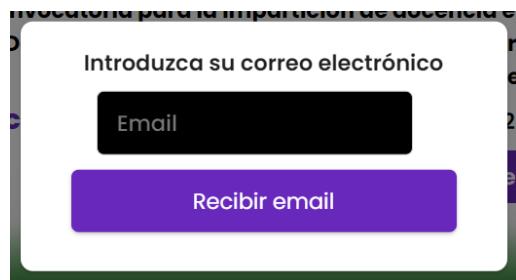


Figura E.7: Suscripción correo

En el caso de que la suscripción se haya realizado correctamente, el aviso se enviará al correo indicado en el formulario y aparecerá el siguiente mensaje de confirmación en la aplicación:

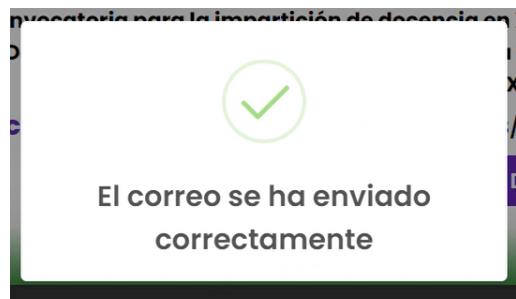


Figura E.8: Correo enviado correctamente

Por el contrario, en el caso de que no se haya enviado el correo debido a un error en la suscripción, aparecerá el siguiente error:

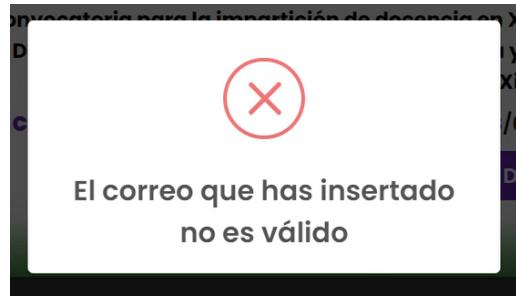


Figura E.9: Correo no válido

Al aparecer este error, se continuará solicitando llenar el formulario hasta que la suscripción sea válida y el correo se envíe correctamente o el usuario haga *click* fuera de la ventana flotante en la que está el formulario.

El número de suscripciones mensuales que se pueden realizar tiene un máximo de 5. Puede darse la situación en el que un usuario quiera exceder ese límite, en este caso se mostrará el siguiente error:



Figura E.10: Límite Avisos Mensuales

Al igual que en el caso anterior en el que nos aparecía el error de suscripción no válida, continuará apareciendo el formulario para introducir un correo hasta que se introduzca un correo válido. En el caso de que se quiera salir del envío de formulario, bastará con hacer *click* fuera de la ventana flotante.

## Añadir al calendario personal

Una vez aparece en la web la ventana en la que se indica que el correo se ha enviado [E.8](#), el usuario deberá iniciar sesión en su correo electrónico personal y verá en la bandeja de entrada un correo cuyo remitente es Delta Offers.

En ese correo, se le notificará al usuario cual es la correspondiente convocatoria de la que ha deseado recibir el aviso y se le adjuntará un archivo ICalendar para añadir un evento a su calendario personal con el fin de plazo de la convocatoria correspondiente.

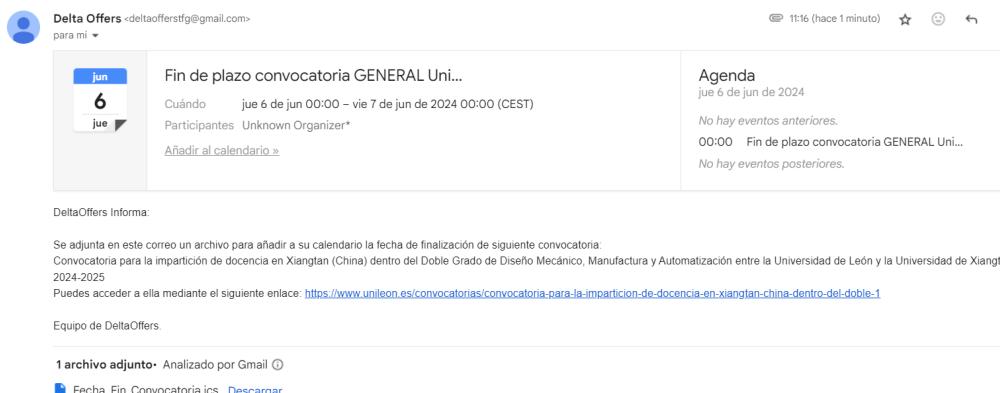


Figura E.11: Correo enviado con archivo ICalendar adjunto

Si el usuario finalmente desease añadir el fin de plazo de la convocatoria como evento en su calendario, deberá hacer *click* en el apartado en el que pone 'Añadir al calendario'. Finalizado todo este proceso, el evento en el calendario aparecería de la siguiente manera:

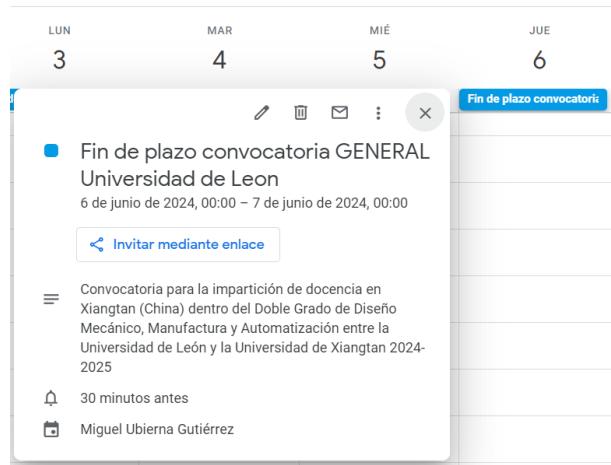


Figura E.12: Evento en calendario personal

## Ver detalles

Tal como se indicó en este apartado E.4, para una convocatoria específica, el usuario dispone de un botón 'Ver detalles' en la que se muestran detalles de la misma.

En este apartado de detalles, aparecerán otros datos e información de las convocatorias que no se mostraba en la ventana maestra en la que aparecía el listado. De esta manera, se proporcionará más información útil al usuario que esté interesado en una determinada convocatoria.

Los datos que podrás ver en esta ventana que anteriormente no se habían mostrado son:

- Descripción
- Convocante
- Fecha publicación
- Destinatarios
- Tipo
- Clasificación
- Nombre plaza
- Convocatoria asociada

*NOTA: Los campos mostrados en esta ventana, varían en función de la Universidad*



Figura E.13: Ventana detalles

Además, desde la ventana de detalles, se dispone de un enlace a la convocatoria publicada en la web de la correspondiente universidad. De esta manera, los usuarios a los que les interese una convocatoria y quieran postularse como candidatos, podrán hacerlo mediante el enlace proporcionado.

The screenshot shows the official website of the Universidad de Burgos. At the top, there is a navigation bar with links to 'Conócenos', 'Futuros estudiantes', 'Estudiantes', 'PDI', 'PTGAS', 'Empresas', and 'Alumni'. Below this is another row with 'SEDE ELECTRÓNICA', 'SECRETARÍA VIRTUAL', 'UBUVIRTUAL', 'BIBLIOTECA', and a search icon. The main content area features the university's logo and name ('UNIVERSIDAD DE BURGOS') along with the 'RUN' logo ('REGIONAL UNIVERSITY NETWORK EUROPEAN UNIVERSITY'). A sub-navigation menu includes 'ESTUDIOS', 'ADMISIÓN Y MATRÍCULA', 'INVESTIGACIÓN', 'INTERNACIONAL', and 'LA UNIVERSIDAD'. Below this, a breadcrumb trail shows 'Inicio > Te interesa > Ref.: 2009/00006/003/117...'. A red banner highlights the title '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'. The main text describes the call for tenders for PDI at the UBU for SAP projects under the RUN-EU 2.0 project. To the right, there are two boxes: 'DENTRO DE PLAZO' (highlighted in green) and 'INFORMACIÓN Y CONTACTO' (highlighted in blue). The 'DENTRO DE PLAZO' box contains the text 'Servicio de Gestión de la Investigación. Vic. de Investigación, Transferencia e Innovación' and 'Edificio de Administración y Servicios, C/ Don Juan de'. The 'INFORMACIÓN Y CONTACTO' box contains the text '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'.

Figura E.14: Convocatoria en la web de una universidad determinada



## *Apéndice F*

---

# **Anexo de sostenibilización curricular**

---

## **F.1. Introducción**

Los Objetivos de desarrollo sostenible [21] fueron implantados en 2015 por las naciones unidas. El principal propósito de estos objetivos es que para el año 2030 todas las personas disfruten de paz y prosperidad en este planeta.

Lo que se busca con la persecución de los mismos es lograr un equilibrio social, económico y ambiental en el planeta.

Para la realización de este apartado, se ha realizado una investigación sobre cuáles son los diferentes objetivos de desarrollo sostenible y cómo pueden estos verse vinculados con el proyecto realizado.

A continuación, se van a mencionar los aspectos en los que se ha considerado que el proyecto contribuye para alcanzar esos objetivos.

### **Objetivo 4 - Educación de calidad**

Este proyecto, facilita el acceso a convocatorias para el Personal Docente e Investigador y Personal de Administración y Servicios. Esto aporta un mayor abanico de oportunidades de empleo en el sector educativo y por lo tanto habrá más opciones que se tengan grupos docentes e investigadores de calidad.

Mediante la utilización de esta web centralizada, habrá un mayor número de candidatos para una determinada convocatoria y como consecuencia de

ello, nos aseguramos de que el profesional elegido, tendrá las condiciones idóneas para realizar esa determinada labor.

### **Objetivo 8 - Trabajo decente y crecimiento económico**

Gracias a web en la que se muestran las convocatorias de manera centralizada, los miembros de la comunidad autónoma incluso del resto del país tendrán un mejor accesibilidad a determinados empleos lo cual deriva en un crecimiento económico del país.

De esta manera, miembros de otras ciudades que no sean de la ciudad en la que está la propia universidad, podrán enterarse de puestos y plazas vacantes de los que anteriormente sin esta aplicación no tendrían conciencia de los mismos. Esto permite evitar que haya puestos que se queden sin ocupar y por lo tanto se promueve enormemente el empleo.

Por último, también es importante considerar que los empleos que ofrecen las universidades para PDI y PAS, son empleos con unas condiciones laborales más que óptimas. Esto contribuye a que los empleos sean decentes.

### **Objetivo 9 - Industria, innovación e infraestructura**

Este proyecto es un proyecto de desarrollo software cuyo código es *Open Source*. Esto hace que el código fuente del programa está disponible para que pueda ser visto, modificado o distribuido por cualquier persona.

Este aspecto contribuye enormemente con este ODS dado que el código abierto permite a otras personas innovar a partir de un proyecto con un código fuente ya desarrollado. De esta manera, una persona que quiera realizar una aplicación innovadora relacionada con esta temática, podrá contar con una base sólida correspondiente a este proyecto.

Por otro lado, la colaboración con la comunidad *Open Source* permite una mayor facilidad para que surjan nuevas empresas y *startups* relacionadas con el desarrollo software.

### **Objetivo 17 - Alianzas para lograr objetivos**

Debido a que la web del proyecto es una web que centraliza convocatorias de distintas universidades, esto potencia enormemente las alianzas entre las distintas universidades de Castilla y León pudiendo llegar a compartir recursos y estableciendo convenios entre las mismas en ámbitos docentes, investigadores y administrativos.

---

# Bibliografía

---

- [1] CampusMVP. Patrón modelo-vista-controlador. <https://www.campusmvp.es/recursos/post/que-es-el-patron-mvc-en-programacion-y-por-que-es-util.aspx>, 2024. [Internet; accedido 23-mayo-2024].
- [2] 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].
- [3] Developer. Inyección de dependencias en .net. <https://www.developerro.com/2023/06/28/dotnet-dependency-injection/>, 2024. [Internet; accedido 24-mayo-2024].
- [4] Endalia. Coste de un trabajador para la empresa. <https://es.talent.com/salary?job=ingeniero+inform%C3%A1tico>, 2024. [Internet; accedido 20-mayo-2024].
- [5] GitHub. Github actions documentation. <https://docs.github.com/es/actions>, 2024. [Internet; accedido 18-mayo-2024].
- [6] Medium. Crear eventos de google calendar en asp.net core. <https://medium.com/@mberrish/create-a-google-calendar-event-from-the-net-core-application-cf6d4a4f3365>, 2024. [Internet; accedido 16-mayo-2024].
- [7] Microsoft. Azure app services. <https://azure.microsoft.com/es-es/products/app-service>, 2024. [Internet; accedido 18-mayo-2024].
- [8] Microsoft. Configuración de inicio de sesión externo de google en asp.net core. <https://learn.microsoft.com/es>

- [es/aspnet/core/security/authentication/social/google-logins?view=aspnetcore-8.0](https://learn.microsoft.com/es-es/aspnet/core/security/authentication/social/google-logins?view=aspnetcore-8.0), 2024. [Internet; accedido 15-mayo-2024].
- [9] Microsoft. Información general asp .net core mvc. <https://learn.microsoft.com/es-es/aspnet/core/mvc/overview?view=aspnetcore-8.0>, 2024. [Internet; accedido 15-mayo-2024].
- [10] Microsoft. Modelo-vista-modelo de vista. <https://learn.microsoft.com/es-es/dotnet/architecture/maui/mvvm>, 2024. [Internet; accedido 25-mayo-2024].
- [11] Medium Multimedia. Guía completa para crear un archivo ics. <https://www.mediummultimedia.com/apps/como-se-crea-un-archivo-ics/>, 2024. [Internet; accedido 16-mayo-2024].
- [12] Tutoriales programación. Base de datos mysql desde python. [https://www.tutorialesprogramacionya.com/pythonya/detalleconcepto.php?punto=81&codigo=81&inicio=75#google\\_vignette](https://www.tutorialesprogramacionya.com/pythonya/detalleconcepto.php?punto=81&codigo=81&inicio=75#google_vignette), 2024. [Internet; accedido 12-mayo-2024].
- [13] PyPi. Autopep8. <https://pypi.org/project/autopep8/>, 2024. [Internet; accedido 11-mayo-2024].
- [14] Aprende Python. Selenium. <https://aprendepython.es/pypi/scraping/selenium/>, 2024. [Internet; accedido 5-junio-2024].
- [15] Docs Python. Creation of virtual environments. <https://docs.python.org/3/library/venv.html>, 2024. [Internet; accedido 12-mayo-2024].
- [16] Docs Python. Mock object library. <https://docs.python.org/3/library/unittest.mock.html>, 2024. [Internet; accedido 30-mayo-2024].
- [17] Docs Python. Unit test. <https://docs.python.org/es/3.9/library/unittest.html#assert-methods>, 2024. [Internet; accedido 11-mayo-2024].
- [18] Tecno Soluciones. Licencias de software de código abierto. <https://tecnosoluciones.com/licencias-de-software-de-codigo-aberto-open-source/>, 2024. [Internet; accedido 20-mayo-2024].

- [19] Talent. Salario medio para ingeniero informático en española. <https://es.talent.com/salary?job=ingeniero+inform%C3%A1tico>, 2024. [Internet; accedido 20-mayo-2024].
- [20] Read the Docs. Requests. <https://requests.readthedocs.io/en/latest/>, 2024. [Internet; accedido 5-junio-2024].
- [21] Undp. Objetivos de desarrollo sostenible. <https://docs.python.org/3/library/unittest.mock.html>, 2024. [Internet; accedido 5-junio-2024].
- [22] Wikipedia. Especificación de requisitos de software. [https://es.wikipedia.org/wiki/Especificaci%C3%B3n\\_de\\_requisitos\\_de\\_software](https://es.wikipedia.org/wiki/Especificaci%C3%B3n_de_requisitos_de_software), 2024. [Internet; accedido 22-mayo-2024].
- [23] Wikipedia. Licencia mit. <https://es.wikipedia.org/wiki/Licencia/MIT>, 2024. [Internet; accedido 20-mayo-2024].
- [24] Wikipedia. Model-view-controller architectural pattern. [https://commons.wikimedia.org/wiki/File:Model-View-Controller\\_architectural\\_pattern.svg](https://commons.wikimedia.org/wiki/File:Model-View-Controller_architectural_pattern.svg), 2024. [Internet; accedido 5-junio-2024].
- [25] Wikipedia. Patrón strategy. [https://es.wikipedia.org/wiki/Strategy\\_\(patr%C3%B3n\\_de\\_dise%C3%BAo\)](https://es.wikipedia.org/wiki/Strategy_(patr%C3%B3n_de_dise%C3%BAo)), 2024. [Internet; accedido 26-mayo-2024].
- [26] Wikipedia. Scrum (desarrollo de software). [https://es.wikipedia.org/wiki/Scrum\\_\(desarrollo\\_de\\_software\)](https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software)), 2024. [Internet; accedido 11-mayo-2024].
- [27] Wikipedia. Strategy pattern. [https://commons.wikimedia.org/wiki/File:Strategy\\_Pattern.png](https://commons.wikimedia.org/wiki/File:Strategy_Pattern.png), 2024. [Internet; accedido 5-junio-2024].