



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 — 2 de junio de 2024
Tutores: Dr. César Ignacio García Osorio
y Dra. Ana Serrano Mamolar

Índice general

Índice general	i
Índice de figuras	iii
Índice de tablas	v
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	2
A.3. Estudio de viabilidad	11
Apéndice B Especificación de Requisitos	19
B.1. Introducción	19
B.2. Objetivos generales	19
B.3. Catálogo de requisitos	20
B.4. Especificación de requisitos	22
Apéndice C Especificación de diseño	41
C.1. Introducción	41
C.2. Diseño de datos	42
C.3. Diseño procedimental	42
C.4. Diseño arquitectónico	44
C.5. Diseño de interfaces	47
Apéndice D Documentación técnica de programación	49
D.1. Introducción	49
D.2. Estructura de directorios	49

D.3. Manual del programador	49
D.4. Compilación, instalación y ejecución del proyecto	49
D.5. Pruebas del sistema	49
Apéndice E Documentación de usuario	51
E.1. Introducción	51
E.2. Requisitos de usuarios	51
E.3. Instalación	52
E.4. Manual del usuario	52
Apéndice F Anexo de sostenibilización curricular	61
F.1. Introducción	61
Bibliografía	63

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. Diagrama de flujo	43
C.3. Patrón Modelo-Vista-Controlador	45
C.4. Inyección de dependencias	46
C.5. Patrón MVVM	46
C.6. Patrón Estrategia	47
C.7. Diseño Ventana Maestra	47
C.8. Diseño Ventana Detalle	48
C.9. Diseño Adaptativo	48
 E.1. Ventana Inicial	53
E.2. Visualización convocatorias	53
E.3. Paginación	54
E.4. Filtro Fecha Fin de Plazo	55
E.5. Aplicar filtros	55
E.6. Convocatoria específica, avisos y detalles	56

E.7. Suscripción correo	57
E.8. Correo enviado correctamente	57
E.9. Correo no válido	57
E.10. Correo enviado con archivo ICalendar adjunto	58
E.11. Evento en calendario personal	59
E.12. Ventana detalles	60
E.13. Convocatoria en la web de una universidad determinada	60

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 planificativas 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 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 analizará 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. Es por ello por lo que se han ido realizando desarrollos iterativos e incrementales los cuales se han ido subiendo a la plataforma de desarrollo GitHub cada 2 semanas. 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 GitHub las cuales se detallaban en el momento que se realizaba una subida a la plataforma con el nuevo incremento, de esta manera, había una cohesión entre el nuevo desarrollo del producto y las tareas llevadas a cabo para conseguir ese nuevo producto funcional.

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 cuales 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 del mismo.

Por otro lado, también se trató el tema de las herramientas que iban a ser utilizadas para la realización del proyecto, buscando unas herramientas las cuales 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 hacer el *web scraping*:** Para determinar todos los atributos que serían interesantes de recopilar para mostrarlos en la web de mi proyecto.
- **Determinar las librerías de Python que van a ser utilizadas para llevar a cabo el proceso de *web scraping***

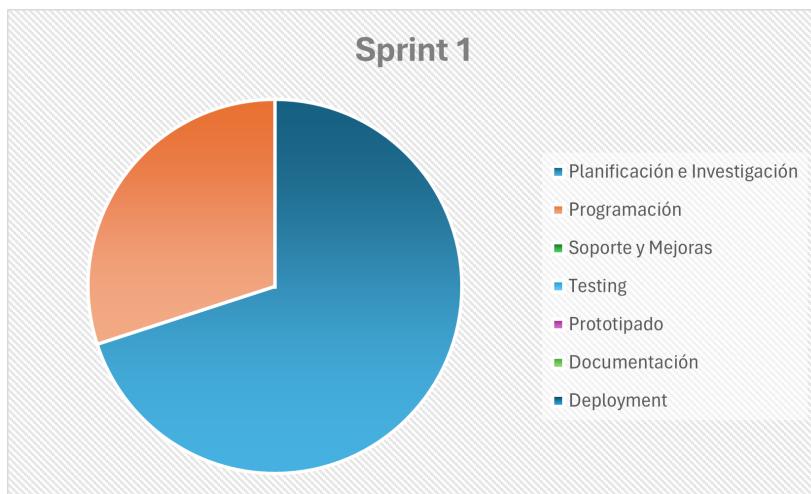


Figura A.1: Sprint 1.

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, Selenium y BeautifulSoup. 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 una limpieza de algunos 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*.

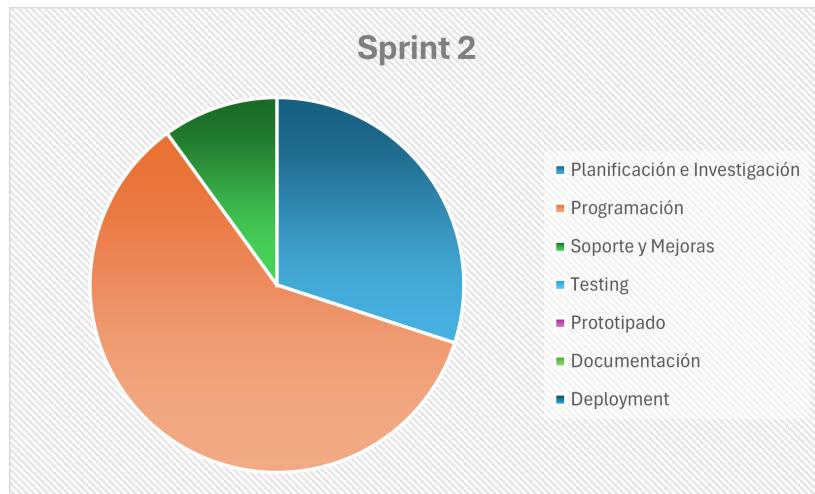


Figura A.2: Sprint 2.

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 esa 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 la cual nos aplicaba la guía de estilos al guardar los cambios en el desarrollo realizado.

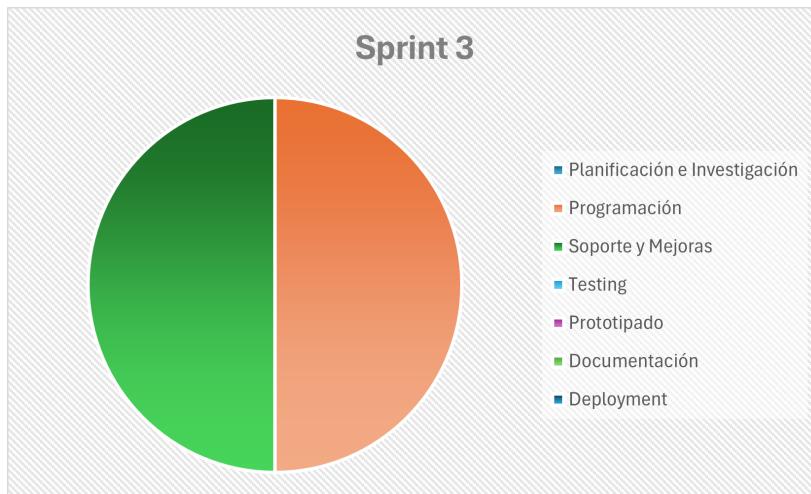


Figura A.3: Sprint 3.

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.

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.

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.

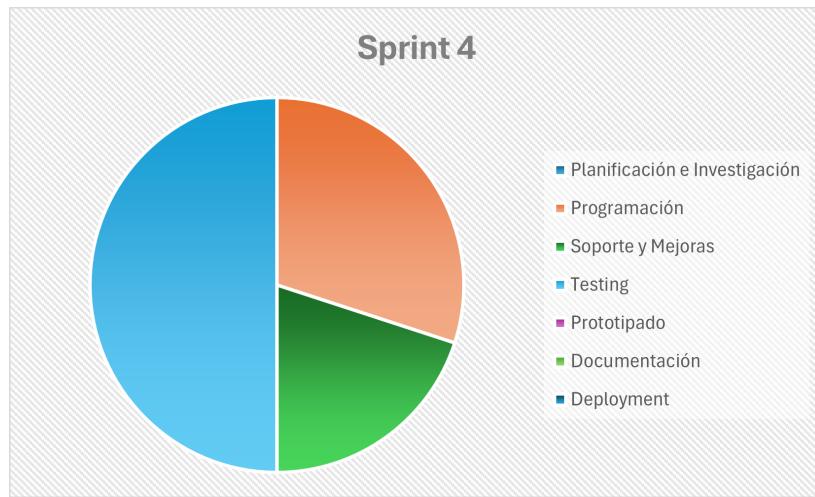


Figura A.4: Sprint 4.

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

Los primeros días de este periodo, consistieron en corregir algunos errores que se percibieron en 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. Esta base de datos puede ser considerado un almacen de datos o *Data Warehouse* en el que aparecian 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.

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 la misma.

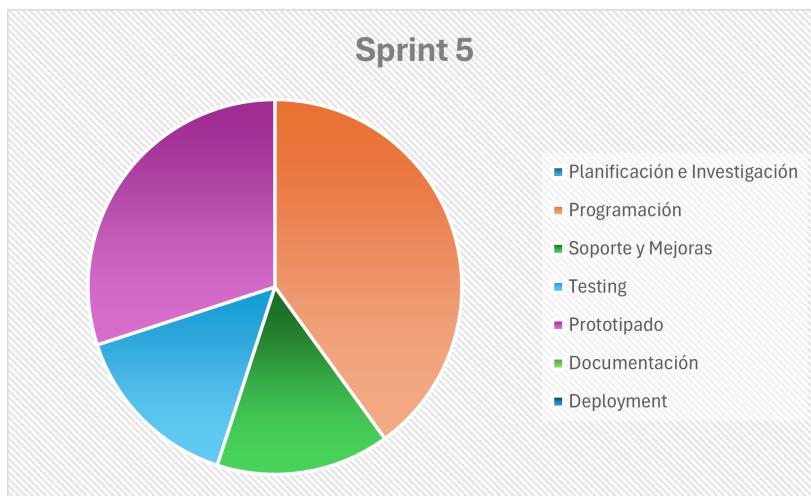


Figura A.5: Sprint 5.

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. 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. 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.

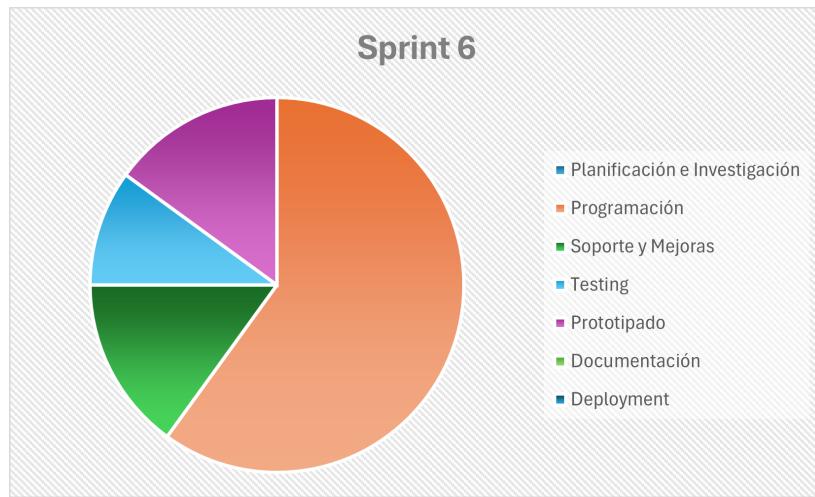


Figura A.6: Sprint 6.

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.

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

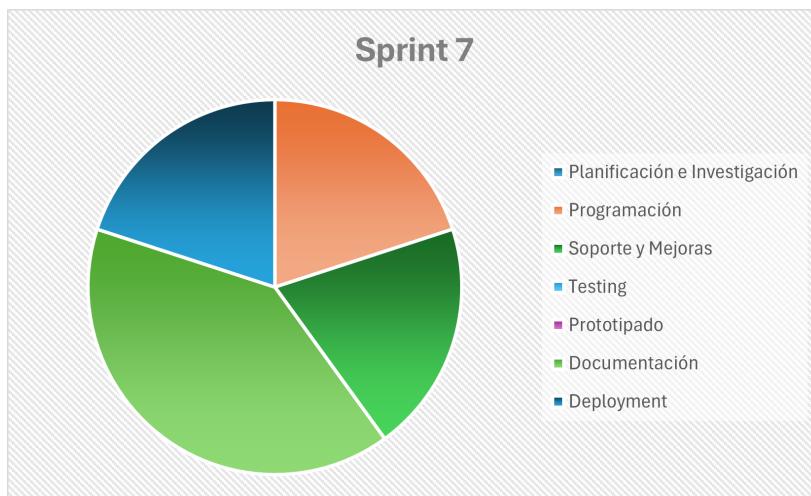


Figura A.7: Sprint 7.

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

En esta etapa, se continuo 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.

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

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.

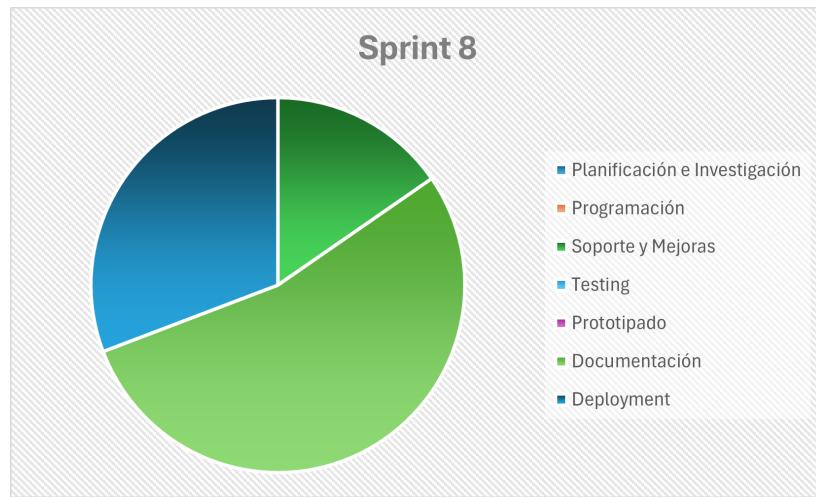


Figura A.8: Sprint 8.

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 dedico la mayoría del tiempo a la preparación de la documentación y ultimos detalles del proyecto para entregarlo al tribunal.

Se realizarón las ultimas mejoras en el código y se realizó una carga del proyecto en una maquina virtual. Esta maquina virtual se almacenará en un USB para su posterior entrega al tribunal.

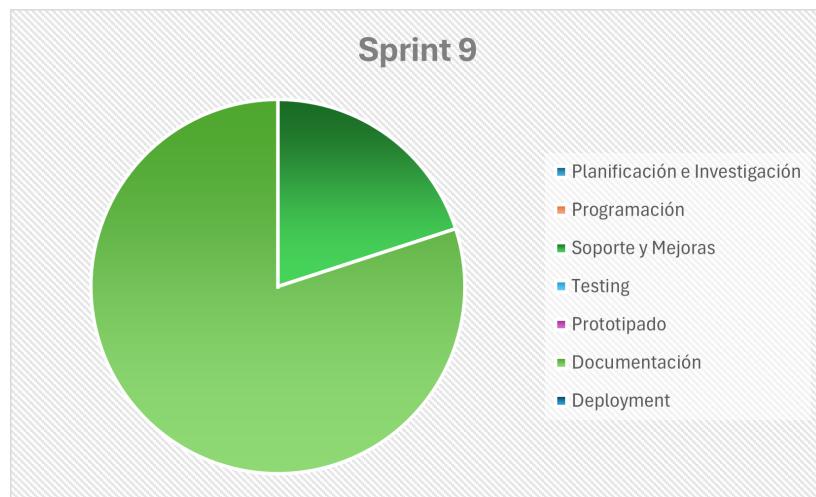


Figura A.9: Sprint 9.

Resumen

Una vez finalizados todos los *sprints* se han podido obtener una serie de conclusiones sobre como 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 *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 dedico 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 € la hora. 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 cual sería, como consecuencia, los costes totales de personal que tendrá la empresa:

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€
Total costes humanos	5546,76€

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 el caso en el 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€
Total costes hardware	879,99€

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 fabrica. 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€
Total otros costes	203,15€

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€
Costes Totales	6629,9€

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 a 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)**. 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 prevee 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 gestionado mediante el entorno *virtualenv* para python y son las siguientes:

Dependencia	Versión	Descripción	Licencia
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	Librería 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 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.

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

Los 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.

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:

- En las primeras iteraciones, como es lógico, se invirtió gran parte del tiempo a la planificación y estructuración del proyecto.

- 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 los mismos y almacenarlos de manera segura.
- Proporcionar a los usuarios interesados 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 redirecciona a la web original.
- Permitir que los usuarios 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 pagina *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 paginas 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 valido.
 - **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 de extension ".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 electronico que recibe el usuario se debe permitir acceder mediante un enlace en la página de detalles a las paginas webs propias de cada universidad.
- **RF-5 Comienzo de uso de la web:** La aplicación debe mostrar una ventana principal con información de la tematica 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 basados en los requisitos previamente mencionados. Posteriormente, se detallarán cada uno de estos.

Diagrama de casos de uso

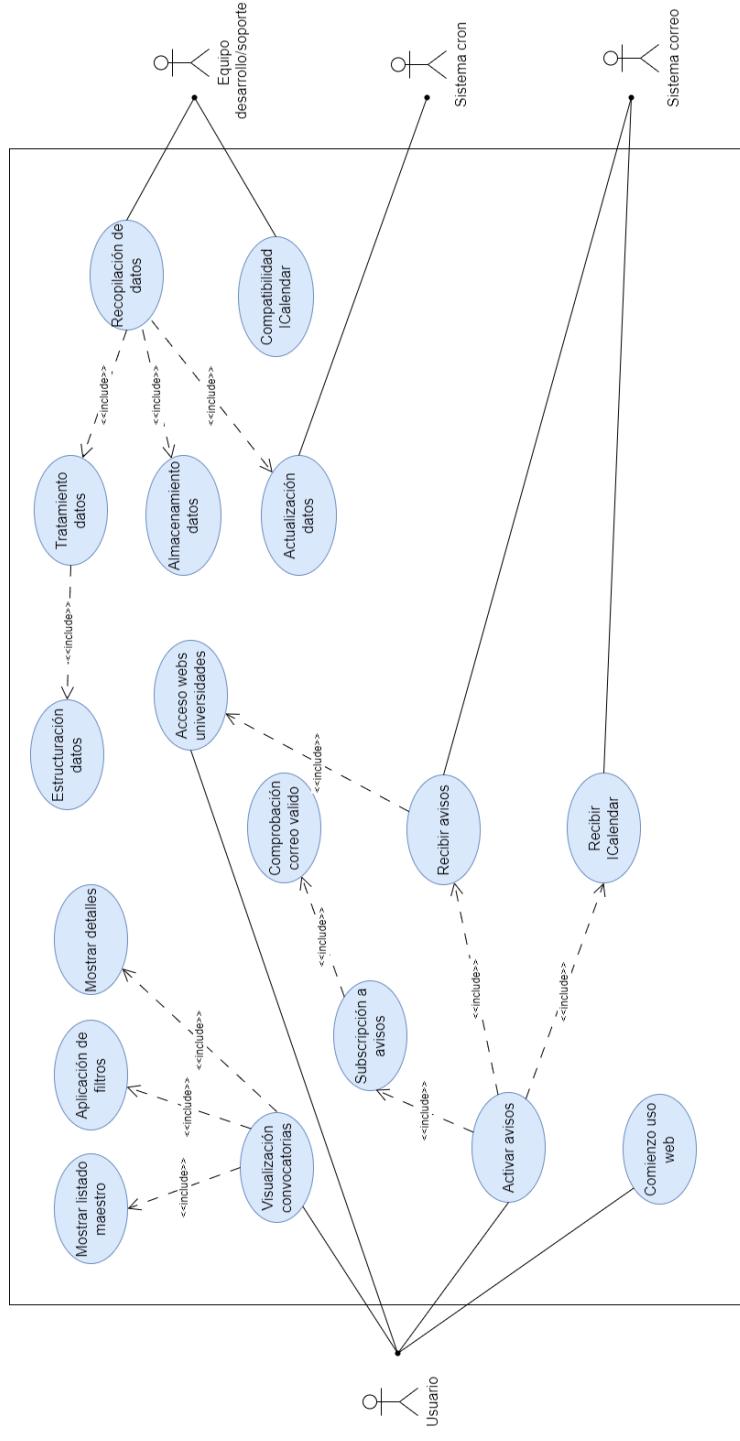


Figura B.1: Diagrama de casos de uso.

Actores

Los actores en un diagrama de casos de uso, son aquellos que desempeñan alguna función dentro del sistema. Los actores no tienen porque 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 esta diseñada para que el usuario pueda interactuar con ella de una manera sencilla y con una buena experiencia de usuario.
- **Equipo desarrollo/sopor:** 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 esta 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 periodicamente 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 correo:** Por último, otro de los actores involucrados ha sido el sistema de correo. Este ha permitido el envio 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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Tabla B.9: CU-9 Mostrar detalles.

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

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

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

Tabla B.11: CU-11 Activar Avisos.

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

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

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

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

CU-14	Recibir archivos ICalendar adjuntos
Versión	1.0
Autor	Miguel Ubierna Gutiérrez
Requisitos asociados	RF-4.3
Descripción	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.
Precondición	Ninguna.
Acciones	<ol style="list-style-type: none"> 1. El usuario se suscribe a los avisos correctamente. 2. El usuario recibe un email con el aviso y el archivo ICalendar. 3. El archivo ICalendar puede ser abierto por el usuario y añadirlo a su calendario personal.
Postcondición	Deberá aparecer un nuevo evento en el calendario personal del usuario.
Excepciones	Ninguna.
Importancia	Baja.

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

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

Tabla B.15: CU-15 Compatibilidad ICalendar.

Apéndice C

Especificación de diseño

C.1. Introducción

En esta apartado, se va 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 de 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

La aplicación solo cuenta con una entidad 'Convocatorias' , se tomo la decisión de solo tener una entidad dado que solo con una se pueden cumplir los objetivos de 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 pesar de que para completar la funcionalidad de nuestra aplicación solo se necesita una entidad, a continuación se va a mostrar cual es y los atributos que tiene.

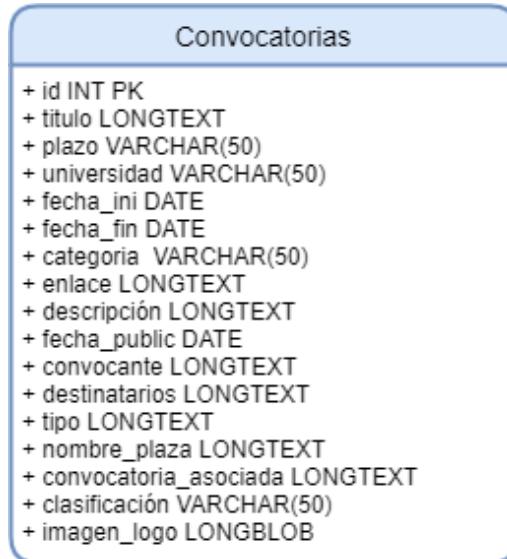


Figura C.1: Entidad Convocatorias.

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

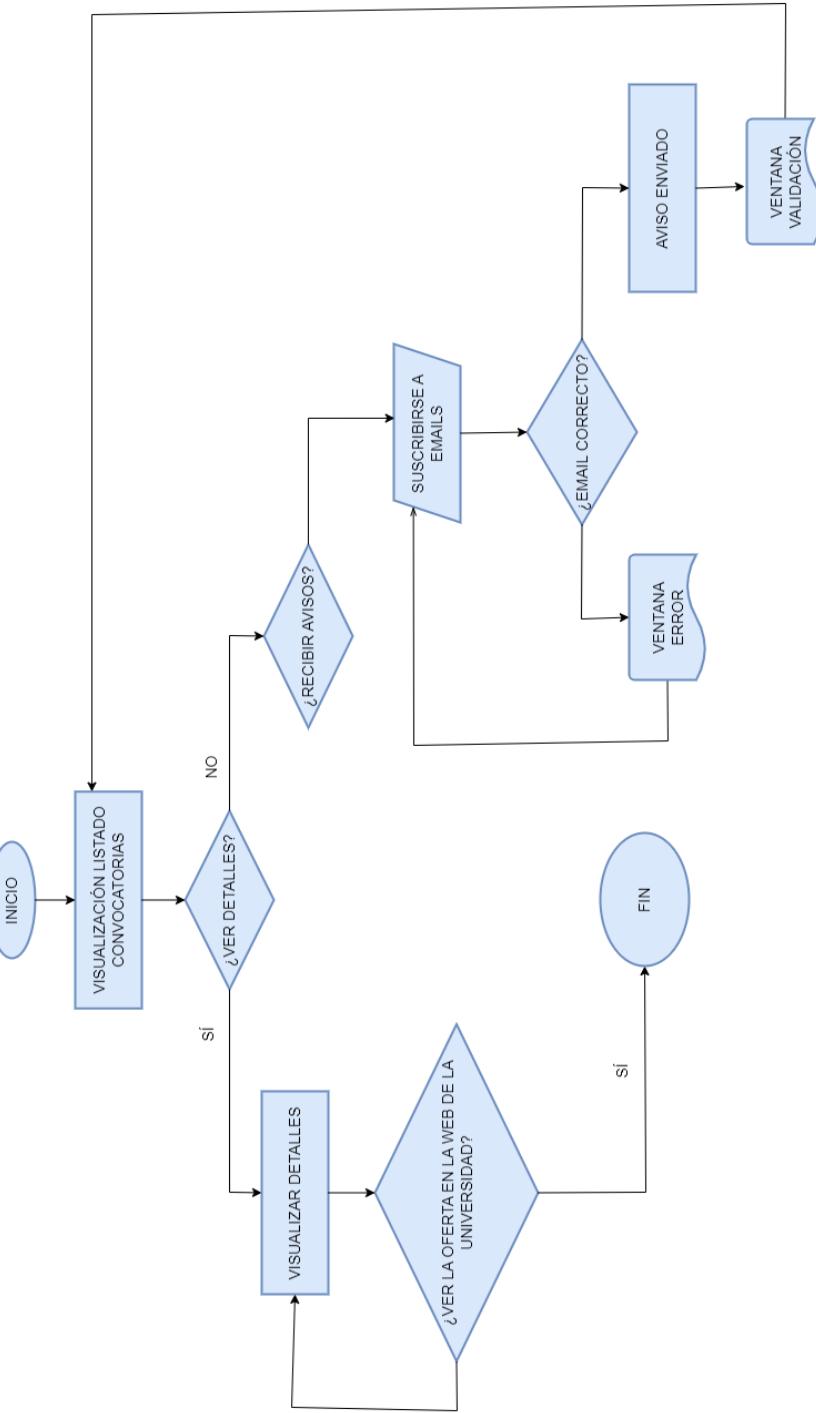


Figura C.2: Diagrama de flujo.

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 van a comentar 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 se utiliza que la interfaz de usuario, los datos y la logica 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 pasadole 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.

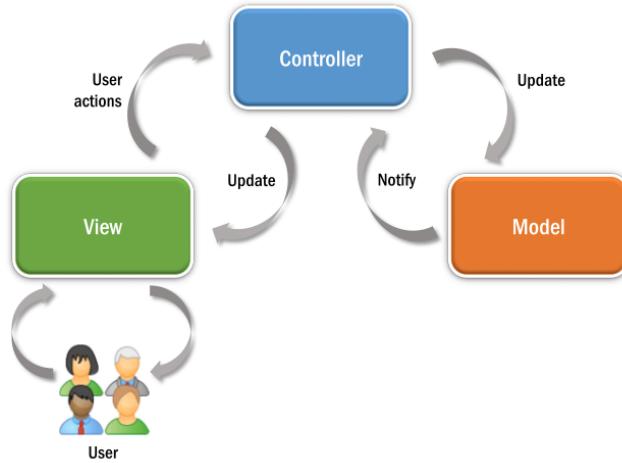


Figura C.3: Patrón Modelo-Vista-Controlador

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. 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.4: Inyección de dependencias

Patrón MVVM (Modelo-Modelo de Vista-Modelo)

Este patrón aisla 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. 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 fueran necesarios para una vista en concreto. De esta manera se evitan realizar cambios en el modelo de datos.

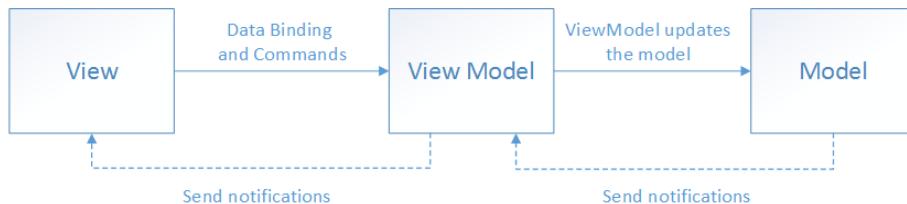


Figura C.5: Patrón MVVM

Patrón Estrategia

Este patrón se aplicó en el proceso de web scraping, su principal propósito es permitir al objeto cliente elegir cual 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.

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.

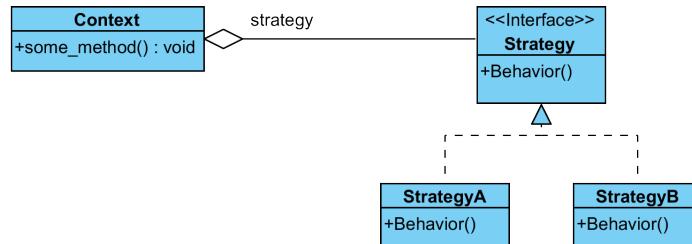


Figura C.6: Patrón Estrategia

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.7: Diseño Ventana Maestra

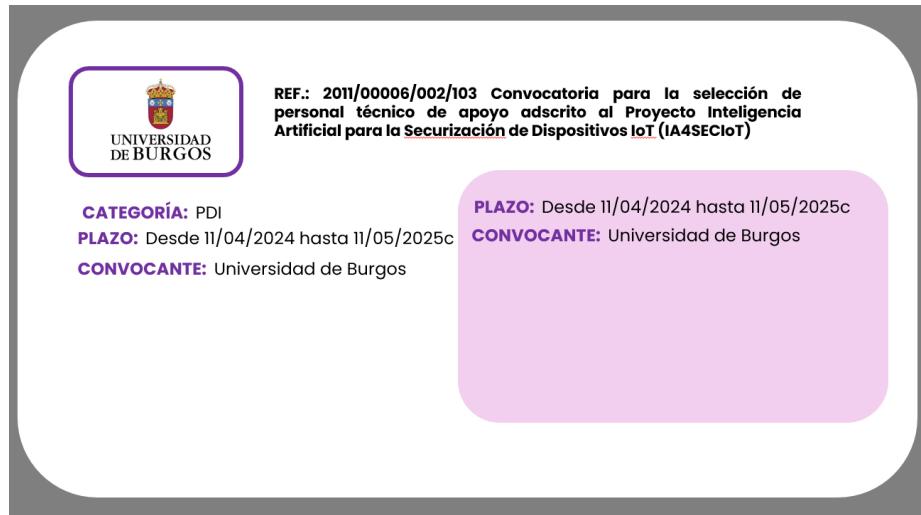


Figura C.8: Diseño Ventana Detalle



Figura C.9: Diseño Adaptativo

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

Apéndice D

Documentación técnica de programación

- D.1. Introducción**
- D.2. Estructura de directorios**
- D.3. Manual del programador**
- D.4. Compilación, instalación y ejecución
del proyecto**
- D.5. Pruebas del sistema**

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 gracias a que la aplicación ya se ha desplegado 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 manera el usuario entenderá el uso de la misma de manera mas 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 cual es el propósito de la web junto con el nombre de la misma. En el momento en el que este desee empezar con la utilización de la misma, hará *click* en el botón 'Empieza ya'.

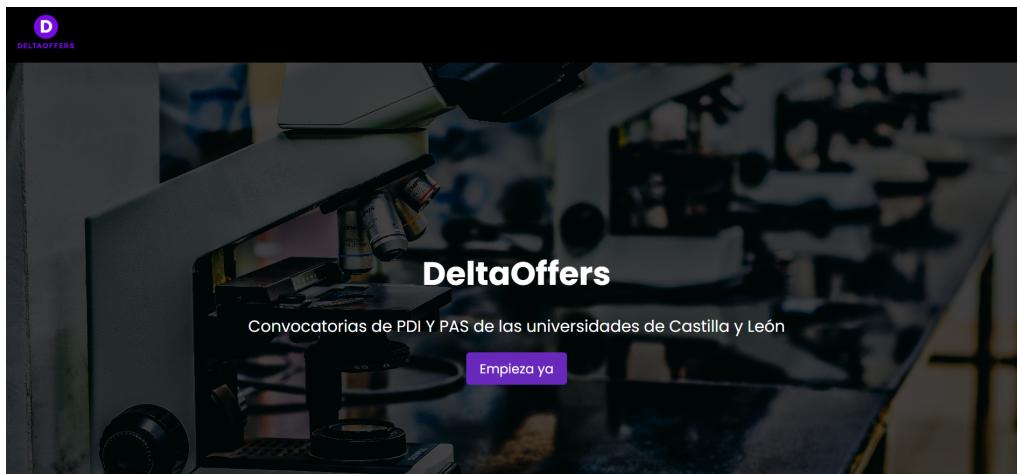


Figura E.1: Ventana Inicial

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.

The screenshot displays the search results for 'convocatorias'. At the top, there are filters for 'Universidad' (Todas), 'Categoria' (Todas), 'Plazo' (Seleccionar plazo), and a 'Filtrar' button. Below the filters, two call cards are shown. The first card is for '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'. It includes the 'UNIVERSIDAD DE BURGOS' logo, the category 'CATEGORIA: PDI', the deadline 'PLAZO: Desde 26/04/2024 hasta 31/12/2024', and buttons for 'Recibir Avisos' and 'Ver Detalles'. The second card is for 'Convocatoria proceso selectivo acceso libre Auxiliares de Servicios' and includes the 'UNIVERSIDAD DE BURGOS' logo, the category 'CATEGORIA: PAS', the deadline 'PLAZO: Hasta 18/06/2024', and buttons for 'Recibir Avisos' and '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 están 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 mas 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	
CATEGORÍA: GENERAL	PLAZO: Desde 28/05/2024 hasta 06/06/2024
Recibir Avisos	Ver Detalles

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

Figura E.5: Aplicar filtros

Convocatoria específica

En cada convocatoria del listado, aparecerá la siguiente información:

- Titulo 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 mas detalles de esa convocatoria en específico.

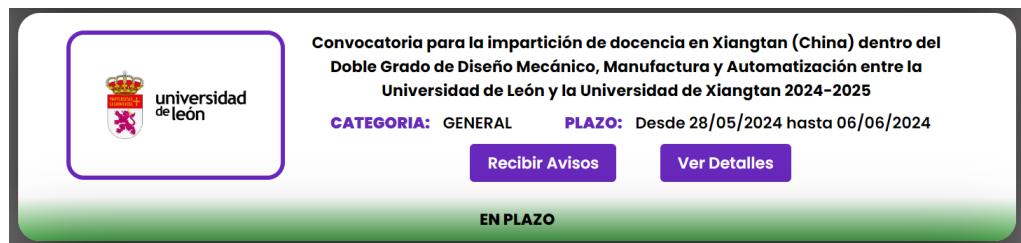


Figura E.6: Convocatoria específica, avisos y detalles

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 los 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 tendrémos que hacer *click* fuera del formulario y volveremos a la ventana de visualización de convocatorias.

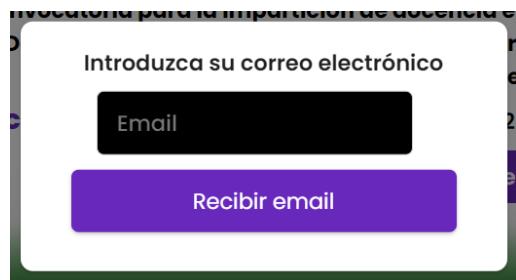


Figura E.7: Suscripción correo

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:

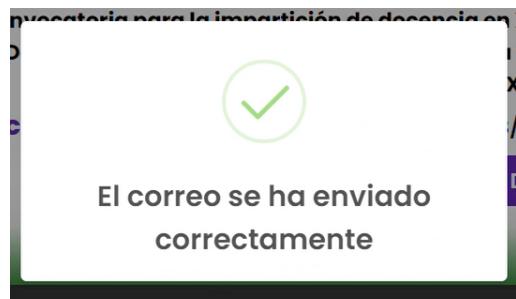


Figura E.8: Correo enviado correctamente

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:

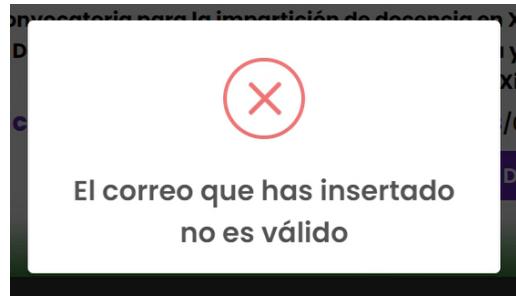


Figura E.9: Correo no válido

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.

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.10: 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:

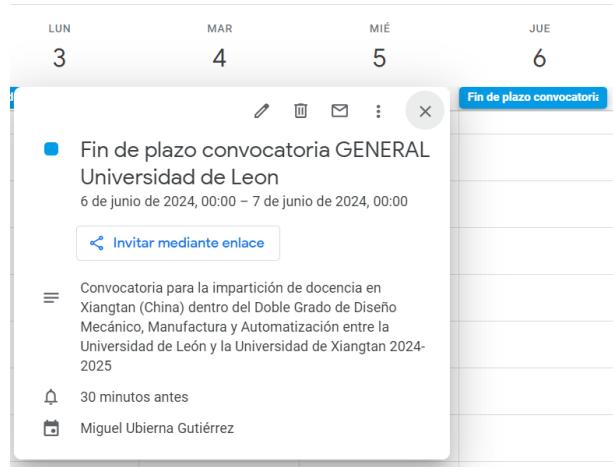


Figura E.11: Evento en calendario personal

Ver detalles

Tal como se indica en este apartado E.4, para una convocatorias 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 listas. 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.12: 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.

UNIVERSIDAD DE BURGOS REGIONAL UNIVERSITY NETWORK EUROPEAN UNIVERSITY

Conócenos Futuros estudiantes Estudiantes ▾ PDI ▾ PTGAS ▾ Empresas Alumni
 SEDE ELECTRÓNICA SECRETARÍA VIRTUAL UBUVIRTUAL BIBLIOTECA 🔎
 English Mapas +34 947 258700 info@ubu.es

ESTUDIOS ▾ ADMIÓN Y MATRÍCULA ▾ INVESTIGACIÓN ▾ INTERNACIONAL ▾ LA UNIVERSIDAD ▾

Inicio > Te interesa > Ref.: 2009/00006/003/117...

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

Se convocan 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

La coordinadora del proyecto RUN-EU 2.0, cofinanciado por el programa ERASMUS+ de la Unión Europea según acuerdo de subvención Ref. 101124674, convoca ayudas para financiar el desarrollo de proyectos denominados SAP (Short Advanced Programmes/Programas Cortos Avanzados) comprometidos en RUN-EU. Se trata de una convocatoria abierta sujeta a disponibilidad de fondos.

DENTRO DE PLAZO

INFORMACIÓN Y CONTACTO

Servicio de Gestión de la Investigación. Vic. de Investigación, Transferencia e Innovación
Edificio de Administración y Servicios. C/ Don Juan de

Figura E.13: Convocatoria en la web de una universidad determinada

Apéndice F

Anexo de sostenibilización curricular

F.1. Introducción

Los Objetivos de desarrollo sostenible 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 cuales 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 numero 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 laborables 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 esté 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 surgen 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
