

## Trabajo Fin de Grado

Desarrollo de una aplicación para la gestión de contenido en el Gabinete de Imagen y Comunicación de la Universidad de Zaragoza.

Development of an application for content management in the Image and Communication Office at the University of Zaragoza.

Autor

Inés López Mareca

Director

Francisco de Borja Pérez Oñate

Ponente

José Javier Merseguer Hernáiz



## Resumen

Este proyecto se enfoca en el desarrollo completo de una aplicación web para el Gabinete de Imagen y Comunicación de la Universidad de Zaragoza.

Los usuarios de esta aplicación podrán gestionar y difundir notas de prensa a través de ella. Además, la aplicación incluye una API REST que permite exportar las notas de prensa a la página corporativa de la Universidad de Zaragoza ([www.unizar.es](http://www.unizar.es)).

El planteamiento de esta aplicación se ha realizado utilizando UML y se han seguido los criterios establecidos por el software hábil.

Al seleccionar las tecnologías, se ha tenido en cuenta el uso de software libre.



## Índice

1.	Introducción.....	1
1.1.	Contexto y motivación.....	1
1.2.	Objetivos del proyecto.....	1
1.3.	Metodología empleada.....	1
1.4.	El manifiesto para el desarrollo de software ágil.....	2
1.5.	Cronograma .....	5
2.	Estado del arte.....	6
2.1.	Herramientas de gestión de contenidos existentes.....	6
2.2.	Aplicación anterior de gestión de contenido.....	6
2.3.	Tecnologías relevantes para el proyecto.....	7
3.	Requisitos.....	9
3.1.	Definiciones, acrónimos y abreviaturas.....	9
3.2.	Requisitos funcionales.....	9
3.3.	Requisitos no funcionales.....	12
4.	Análisis.....	14
4.1.	Identificación de los casos de uso.....	14
4.2.	Modelado de datos.....	25
4.3.	Análisis dinámico.....	30
5.	Diseño.....	33
5.1.	Diseño de la arquitectura.....	33
5.2.	Estructura de la aplicación.....	33
5.3.	Diseño de la interfaz de usuario.....	34
6.	Implementación.....	36
6.1.	Definiciones, acrónimos y abreviaturas.....	36
6.2.	Proceso de desarrollo y herramientas empleadas.....	36
6.3.	API Rest y migración de datos de la antigua aplicación.....	37
7.	Conclusiones y trabajos futuros.....	38
7.1.	Conclusiones.....	38
7.2.	Trabajos futuros.....	38
8.	Bibliografía.....	40
	Anexo A. Análisis de las clases que forman la aplicación.....	41
	Anexo B. Diagramas de secuencia relevantes en la aplicación.....	46
	Anexo C. Tecnologías empleadas en la aplicación.....	48
	Anexo D. Configuración de la API REST de la aplicación Notas de Prensa y Drupal..	52
	Anexo E. Capturas de la aplicación y cumplimiento de los requisitos funcionales.....	56



## 1. Introducción.

### 1.1. Contexto y motivación.

El Gabinete de Imagen y Comunicación de la Universidad de Zaragoza (GIC-UZ) es responsable de gestionar la comunicación de las actividades y eventos de la Universidad de Zaragoza. Este trabajo se lleva a cabo mediante la creación y distribución de notas de prensa. Hasta la fecha, el GIC-UZ ha utilizado una aplicación interna para esta tarea. La aplicación presenta limitaciones y fallos que dificultan la labor de los usuarios y afectan a la calidad y efectividad de la comunicación.

Por tanto, surge la necesidad de desarrollar una nueva aplicación que cubra las necesidades de los usuarios del GIC-UZ. La motivación de este Trabajo de Fin de Grado es crear una aplicación que permita la gestión de notas de prensa de forma sencilla. Mejorando así la eficacia en los procesos de comunicación desarrollados por el GIC-UZ. La nueva aplicación supondrá una mejora para el trabajo del GIC-UZ, ya que podrá contar con una herramienta adaptada a sus necesidades y desarrollada con tecnologías que se integran con el resto de las aplicaciones corporativas.

### 1.2. Objetivos del proyecto.

- Desarrollar una nueva aplicación que permita la gestión y distribución de notas de prensa de forma sencilla, eficiente y segura para el GIC-UZ.
- Utilizar tecnologías en el estado del arte para el desarrollo de la aplicación, que garanticen su mantenimiento a largo plazo.
- Diseñar una arquitectura, que permita la integración de nuevas funcionalidades en el futuro, en función de las necesidades del GIC-UZ.
- Definir e implementar una API REST [1] que permita la integración de la nueva aplicación con la página corporativa de la Universidad. En consecuencia, las notas de prensa elaboradas en la aplicación podrán ser publicadas automáticamente en la página.
- Realizar pruebas exhaustivas de la aplicación para garantizar su correcto funcionamiento.
- Migrar los datos de la aplicación anterior a la nueva, garantizando la integridad y seguridad de estos.

La integración con la aplicación iUNIZAR no formará parte del alcance inmediato del proyecto, aunque se considerará como una posible futura mejora de la aplicación.

### 1.3. Metodología empleada.

En el desarrollo de la aplicación, se utilizó una combinación del ciclo de vida con prototipado [2] y la metodología Scrum [3]. Esta elección se basó en diversas consideraciones que tenían en cuenta una gestión eficiente del proyecto y una satisfacción óptima de las necesidades del cliente, tal y como se describe a continuación.

El ciclo de vida con prototipado fue seleccionado debido a su capacidad para adaptarse a las particularidades de un proyecto en constante evolución, como es el caso del desarrollo del software que nos ocupa. Este método se caracteriza por la repetición de fases a lo largo del tiempo, con el fin de añadir nuevas funcionalidades y mejoras en cada una de ellas. Cada iteración implica el desarrollo de prototipos que permiten evaluar la nueva funcionalidad del producto.

Esta metodología coincide con uno de los principios del desarrollo ágil, el de satisfacción del cliente mediante la entrega temprana y continua de software con valor, ya que cada iteración genera resultados, los cuales permiten la participación del cliente a lo largo del desarrollo del proyecto.

Por otro lado, se eligió Scrum como metodología ágil para la gestión del proyecto. Scrum tiene un enfoque iterativo y adaptable, lo que la convierte en una elección adecuada para proyectos en los que las necesidades del cliente pueden cambiar a lo largo del tiempo. Como es común en esta metodología, se estableció un marco de trabajo que permitió una comunicación constante con el cliente para recibir así retroalimentación de forma regular. Las reuniones mensuales con el cliente desempeñaron un papel crucial en este proceso, ya que se utilizaron para validar los requisitos y establecer nuevos.

La combinación del ciclo de vida con prototipado y la metodología Scrum permitió una planificación y ejecución eficiente del proyecto. El uso de prototipos en cada iteración facilitó la identificación temprana de posibles mejoras y ajustes, mientras que la metodología Scrum permitió una gestión efectiva de las tareas y la adaptación a los cambios en los requisitos del cliente.

### CICLO DE VIDA PROTOTIPADO

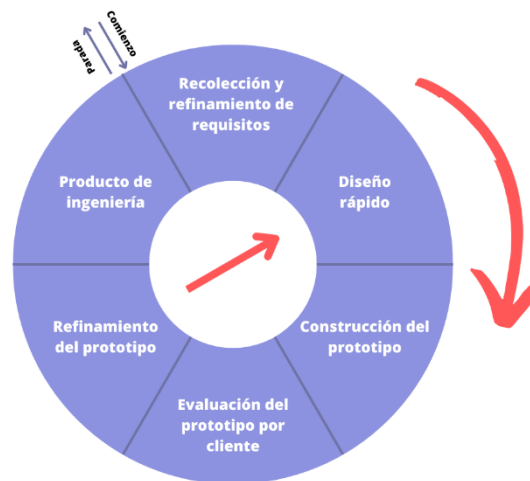


Figura 1. Esquema funcionamiento Ciclo de vida con prototipado.

#### 1.4. El manifiesto para el desarrollo de software ágil.

La importancia de desarrollar software usando metodologías ágiles puede ser evaluada desde múltiples puntos de vista. A continuación, se realiza una valoración personal de cómo cada uno de los 12 principios del Manifiesto Ágil [4] se han materializado en el proyecto a través de la Metodología Scrum.

- *“Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.”*

La elección del ciclo de vida con prototipado y la metodología Scrum ha permitido entregar regularmente prototipos y funcionalidades valiosas. Las reuniones periódicas con el cliente han asegurado la participación continua y la retroalimentación del cliente en cada etapa del desarrollo, lo que ha garantizado la satisfacción del cliente al recibir software con valor de manera temprana y constante.

Para lograr esto, se planificaron entregas regulares del producto final, asegurando así la corrección de requisitos y la participación del cliente en la evolución del proyecto. Además, se programaron varias reuniones tanto con el GIC-UZ como con el Área de Servicios de la Red de la Universidad de Zaragoza (ASR-UZ), con el propósito de abordar posibles problemas y soluciones, nuevas funcionalidades y próximos pasos.

- *“Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.”*

El ciclo de vida con prototipado y Scrum se adapta a los cambios en los requisitos del cliente. Las iteraciones y prototipos permiten evaluar los requisitos a medida que avanza el proyecto. Las reuniones periódicas con el cliente han facilitado la comunicación para proporcionar soluciones a los posibles problemas planteados.

- *“Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.”*

Es importante el constante feedback del cliente ante los avances de la aplicación. De esta manera, hemos conseguido minimizar pérdidas de tiempo en las rectificaciones o cambios en el producto final.

Se establecieron reuniones mensuales con el cliente. Estas reuniones se llevaron a cabo con el propósito de presentar las nuevas funcionalidades incorporadas al producto y permitir al cliente interactuar con ellas. Durante estas sesiones, se proporcionó una demostración de las características implementadas, brindando al cliente la oportunidad de evaluar su funcionalidad y realizar comentarios en directo.

- *“Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.”*

La metodología Scrum promueve la colaboración continua entre el equipo de desarrollo y los responsables de negocio, que en este caso son los periodistas del GIC-UZ. Las reuniones periódicas con el cliente han permitido una interacción frecuente entre ambas partes. Esta colaboración constante ha garantizado una comprensión clara de los requisitos y una alineación entre las necesidades del cliente y el desarrollo del proyecto.

El ASR-UZ se ha reunido mensualmente durante el transcurso del desarrollo de la aplicación, garantizando que no surgían adversidades a la hora de trabajar en los entornos informáticos de la Universidad de Zaragoza (UZ).

- *“Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.”*

El proyecto, a pesar de haber sido desarrollado en su integridad por la alumna, ha sido guiado por sus tutores. El recibir ayuda externa en el caso de dificultades ha ayudado a que no haya bajones en la motivación y por tanto en la productividad. También se me ha dado libertad a la hora de elegir muchas de las tecnologías utilizadas confiando en mi criterio de elección.

- *“El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.”*

Las revisiones del prototipo siempre se han desarrollado en persona o por videollamada. De esta forma, es mucho más sencillo entender los puntos de vista tanto del cliente como del equipo informático.

- *“El software funcionando es la medida principal de progreso.”*

La elección del ciclo de vida con prototipado y la metodología Scrum se centra en la entrega continua de software funcional. Al entregar prototipos y funcionalidades en cada iteración, se ha podido medir el progreso del proyecto en base a los resultados tangibles y funcionales obtenidos. Esto ha proporcionado una visibilidad clara del avance del proyecto y ha permitido realizar ajustes y mejoras según sea necesario.

- *“Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.”*

El ciclo de vida con prototipado y la metodología Scrum promueven el desarrollo sostenible al enfocarse en entregas regulares y mantenimiento de un ritmo constante. Al dividir el proyecto en iteraciones y establecer objetivos alcanzables en cada una, se ha evitado la sobrecarga de trabajo y el agotamiento de la alumna. Además, la retroalimentación constante y las mejoras continuas garantizan que el proyecto pueda mantener un ritmo sostenible a largo plazo.

- *“La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.”*

La elección del ciclo de vida con prototipado y la metodología Scrum fomenta la excelencia técnica y el buen diseño. Mediante la entrega frecuente de prototipos y funcionalidades, se ha tenido la oportunidad de evaluar y mejorar continuamente el diseño y la calidad del software. La retroalimentación constante del cliente y la colaboración activa con el ASR-UZ ha permitido realizar ajustes y mejoras en el diseño técnico, lo que ha contribuido a la Agilidad y a la entrega de un producto de alta calidad.

- *“La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.”*

El ciclo de vida con prototipado y la metodología Scrum promueven la simplicidad al enfocarse en la entrega de funcionalidades esenciales y evitar el desarrollo de características innecesarias. Al establecer prioridades claras y centrarse en los requisitos

fundamentales, se ha maximizado la eficiencia y minimizado el trabajo no productivo. Esto ha permitido un enfoque ágil y una entrega más rápida de valor al cliente.

- *“Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.”*

La elección del ciclo de vida con prototipado y la metodología Scrum fomentó la auto-organización de la alumna. Al permitir que ésta tome decisiones y se organice de forma autónoma, se fomenta la creatividad. Esto ha permitido que surjan las mejores arquitecturas, requisitos y diseños. La participación activa del cliente en las reuniones periódicas ha ayudado a refinar y mejorar aún más estos aspectos.

- *“A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.”*

La metodología Scrum incluye retrospectivas regulares, en las que el equipo reflexiona sobre su desempeño y busca formas de mejorar. Estas reflexiones permiten que la alumna identifique aspectos en los que mejorar y aplique cambios para aumentar su efectividad. En este proyecto, se han llevado a cabo reuniones periódicas con el cliente y pruebas exhaustivas, lo que ha brindado oportunidades para la reflexión y la mejora continua. Estas acciones han permitido a la alumna ajustar su comportamiento y lograr una mayor eficiencia en el desarrollo del software.

### 1.5. Cronograma

En la Figura 2 se puede observar un cronograma de Gantt que planifica cada etapa del proyecto.

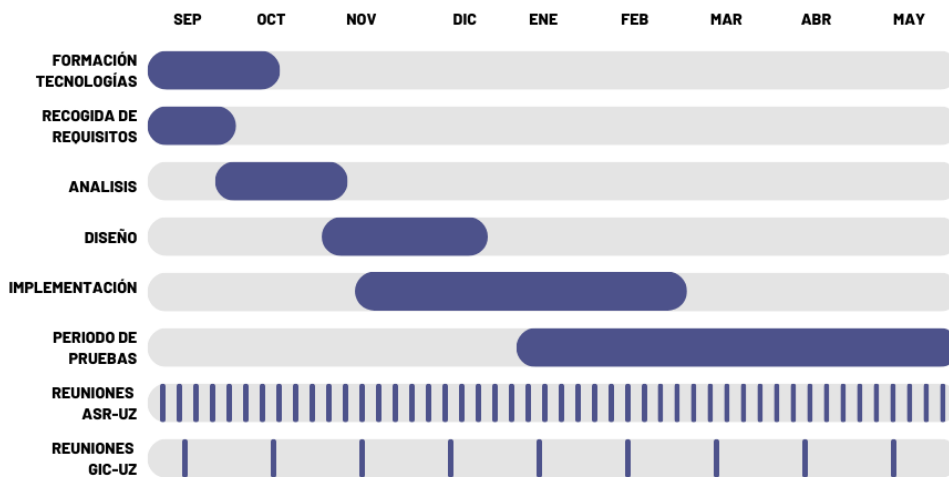


Figura 2. Diagrama Gantt – App “Notas de prensa”.

## 2. Estado del arte.

### 2.1. Herramientas de gestión de contenidos existentes.

Para la gestión y distribución de notas de prensa, varias herramientas populares de gestión de contenido combinadas con funcionalidades de envío de correo electrónico ofrecen soluciones efectivas. Entre ellas se encuentran MailChimp [5], Constant Contact [6], HubSpot [7] y SendinBlue [8].

Estas herramientas proporcionan la capacidad de crear y enviar correos electrónicos personalizados a una lista de destinatarios. Permiten segmentar audiencias por medios de listas y programar envíos automáticos. Además, ofrecen plantillas y diseños predefinidos para facilitar la creación de contenido visualmente atractivo.

Sin embargo, la opción de desarrollar una aplicación nueva exclusiva para la UZ ofrece beneficios adicionales. Al tener una aplicación personalizada, se puede adaptar y agregar funcionalidades específicas que el cliente requiera para una gestión eficiente de las notas de prensa.

Al desarrollar una aplicación interna, se tiene un mayor control sobre la implementación de características adicionales y la integración con los procesos existentes de la UZ. Al reducir la dependencia de tecnologías independientes, la aplicación será más compatible con las aplicaciones ya utilizadas en la institución.

Además, una aplicación exclusiva para la UZ permite mayor privacidad y seguridad de los datos. Al mantenerlos dentro de la infraestructura de la UZ, se minimizan los riesgos asociados con el uso de herramientas de terceros.

### 2.2. Aplicación anterior de gestión de contenido.

En el ámbito universitario, actualmente se utiliza una aplicación desactualizada para la gestión de notas de prensa, la cual presenta diversas limitaciones y deficiencias. Esta aplicación ha mostrado errores frecuentes. Por ejemplo, al cargar archivos, lo que dificultaba la correcta gestión de contenidos. Además, se han reportado problemas relacionados con pantallas en blanco, lo que afecta la experiencia del usuario y su capacidad para acceder y utilizar la aplicación de manera efectiva.

Una de las principales limitaciones de esta aplicación desactualizada radica en la imposibilidad de añadir nuevas funcionalidades. Esto significa que la UZ se encuentra limitada en términos de mejoras y adaptaciones que pueda requerir en el futuro. No poder implementar nuevas funcionalidades puede afectar negativamente la eficiencia y productividad del equipo de comunicación de la UZ, así como su capacidad para mantenerse actualizado con las últimas tendencias y necesidades del entorno de notas de prensa.

Otra preocupación a tener en cuenta es el lenguaje de programación utilizado en la aplicación, PHP [9], cuya versión está desactualizada. El uso de una versión obsoleta de PHP puede conllevar riesgos de seguridad y limitar la compatibilidad con tecnologías modernas y frameworks. El uso de nuevas tecnologías mejoraría significativamente la funcionalidad y el rendimiento de la aplicación.

Además, el diseño de la interfaz de la aplicación se encuentra desactualizado, lo que resulta en una experiencia poco atractiva para los usuarios. Esto podría afectar negativamente a la buena gestión de notas de prensa, ya que una interfaz desactualizada puede dificultar el uso de las funcionalidades disponibles.

Considerando este escenario, la opción de desarrollar una nueva aplicación exclusiva para la UZ se presenta como una solución más viable. Al contar con una nueva aplicación, se pueden superar los problemas actuales y ofrecer una plataforma más moderna, eficiente y segura para la gestión de notas de prensa. Esta nueva aplicación permitiría abordar las limitaciones actuales, mientras brinda la flexibilidad de añadir nuevas funcionalidades.

### 2.3. Tecnologías relevantes para el proyecto.

En el desarrollo de la aplicación de gestión de notas de prensa para el GIC-UZ, se han utilizado múltiples tecnologías que han sido fundamentales para su funcionamiento. Es importante destacar que se ha procurado utilizar tecnologías de software libre debido a sus ventajas, como el acceso al código fuente y la independencia tecnológica. A continuación, se describen las tecnologías más relevantes utilizadas:

Para la experiencia de usuario, se han empleado los lenguajes de programación HTML [10], CSS [11] y JavaScript [12]. Estos lenguajes permiten diseñar una interfaz intuitiva y atractiva para los usuarios, brindando una experiencia agradable durante el uso de la aplicación.

En cuanto al desarrollo del backend, se ha utilizado el framework Django [13], que se basa en el lenguaje de programación Python. Django ofrece un conjunto de herramientas y características que agilizan el desarrollo y proporcionan una estructura sólida para la aplicación. Django permite el manejo de tareas como la autenticación de usuarios, gestión de base de datos, uso de formularios, etc.

La base de datos utilizada en el proyecto ha sido MariaDB [14], un sistema de gestión de bases de datos relacional de código abierto. MariaDB garantiza eficiencia y seguridad en el almacenamiento de la información de las notas de prensa y otros datos.

La autenticación de usuarios se realiza mediante el Central Authentication Service (CAS) de la UZ, lo que permite que los usuarios utilicen sus credenciales institucionales para acceder a la aplicación. Esto asegura la confidencialidad y la integridad de los datos en el entorno productivo de la UZ.

La aplicación ofrece una API (Application Program Interface) que permite importar los contenidos de la anterior aplicación utilizada, así como exportarlos a la página corporativa de la UZ, que ha sido desarrollada en este proyecto con Drupal [15]. Estas operaciones se realizan mediante el intercambio de datos en formatos JSON [16] (JavaScript Object Notation) y CSV (Comma-Separated Values), lo que facilita la transferencia de información entre las diferentes plataformas.

En cuanto al despliegue, la aplicación se encuentra alojada en un entorno Docker [17] en un servidor Linux de la UZ. Docker proporciona una forma eficiente y segura de empaquetar y desplegar la aplicación, garantizando su escalabilidad.

En términos de seguridad, la aplicación se ha asegurado utilizando una conexión HTTPS [18], lo que garantiza la confidencialidad y la integridad de los datos transmitidos entre los usuarios y el servidor. Esto brinda confianza a los usuarios al utilizar la aplicación.

Además, el uso de un framework ayuda a mitigar posibles agujeros de seguridad. El hecho de ser un software libre hace que exista una comunidad activa de desarrolladores que trabajan constantemente en mejorar la seguridad del software, mediante la identificación y corrección de vulnerabilidades. Las actualizaciones regulares aseguran que se incorporen las últimas medidas y parches de seguridad necesarios.

### 3. Requisitos.

#### 3.1. Definiciones, acrónimos y abreviaturas.

**Nota de prensa.** Documento redactado por el GIC-UZ en el que se aborda un tema de actualidad realizado por la UZ. Este se envía a diversos medios de prensa.

**Fecha de publicación.** Fecha establecida por el periodista del GIC-UZ para el envío de la nota de prensa.

**Autor.** Se refiere al departamento de redacción de la UZ encargado de la elaboración de esa nota de prensa.

**Título.** Frase concisa que resume el contenido de la nota y atrae la atención del lector.

**Subtítulo.** Líneas que complementan el título y proporcionan más información sobre el contenido de la nota.

**Cuerpo.** Contenido de la nota de prensa.

**Contacto.** Datos de contacto a los que el lector puede recurrir para obtener más información sobre el contenido abordado en la nota.

**Multimedia.** Contenido visual que complementa la nota de prensa.

**Firma.** Información del autor de la nota, que incluye los datos de contacto y el logotipo del departamento en la UZ.

**Usuario.** Persona afiliada a la UZ que ha sido autenticada exitosamente en la aplicación de notas de prensa.

**Lector.** Cualquier persona que lea la nota de prensa una vez publicada.

**Web corporativa.** Se trata de la página web <https://www.unizar.es/>, donde se puede encontrar toda la información relacionada con la UZ y donde se publican todas las notas de prensa para que los lectores puedan acceder a ellas.

**Medios de prensa.** Son los destinatarios de la base de datos de la aplicación a quienes se envían las notas de prensa.

**iUnizar.** Se trata del boletín informativo diario por correo electrónico, donde se notifican todas las novedades de la UZ a los miembros de la institución.

#### 3.2. Requisitos funcionales.

Creación y edición de Notas de prensa.

##### **RF-1 Crear Nota de Prensa.**

**DESCRIPCIÓN** La aplicación debe permitir que el usuario cree una Nota de prensa.

El usuario debe incluir obligatoriamente los campos: **Fecha de publicación, Autor, Título y Cuerpo**. Además, puede incluir opcionalmente los siguientes elementos: **Subtítulo, Contacto y Multimedia**.

**RF-2                    Añadir autores a la Nota de prensa y firmarla.**

**DESCRIPCIÓN** La aplicación debe permitir que una Nota de Prensa pueda tener más de un Autor. Además, una vez que la nota ha sido creada, debe ser posible agregar una **firma** para cada uno de los Autores involucrados.

**RF-3                    Editor de texto para campos determinados de la Nota de Prensa.**

**DESCRIPCIÓN** Los campos de Título, Subtítulo, Cuerpo y Contacto de las Notas de Prensa deben contar con un editor de texto que permita al usuario darles el formato deseado. Este editor de texto debe incluir las siguientes opciones de edición: negritas, cursivas, subrayados, cambios de tamaño y de tipo de letra y alineación de texto.

El objetivo de este requisito es que el **usuario** pueda personalizar y mejorar la presentación de la información que se incluye en cada campo de la Nota de Prensa, para que esta sea más atractiva y legible para los **lectores**.

**RF-4                    Gestión de archivos multimedia en la Nota de Prensa.**

**DESCRIPCIÓN** La sección “Apartado multimedia” en la aplicación deberá permitir subir múltiples archivos, los cuales podrán ser renombrados y reordenados según la preferencia del usuario.

Una vez subidos los archivos, se mostrarán una serie de enlaces con el nombre que el usuario haya decidido, y al hacer clic en ellos, se redireccionará al archivo multimedia correspondiente.

## Gestión de Notas de prensa.

<b>RF-5</b>	<b>Editar Nota de Prensa.</b>
<b>DESCRIPCIÓN</b>	El usuario tendrá la opción de editar cualquier Nota de Prensa creada tantas veces como desee.
<b>RF-6</b>	<b>Duplicar Nota de Prensa.</b>
<b>DESCRIPCIÓN</b>	<p>La aplicación debe permitir al usuario crear una copia idéntica de una Nota de Prensa existente incluyendo todos los elementos y campos de la copia original. La copia creada, será totalmente independiente de la Nota de Prensa original.</p> <p>De esta manera el usuario podrá crear nuevas Notas de Prensa a partir de las ya existentes sin tener que volver a crearlas de cero facilitándole el trabajo al usuario.</p>
<b>RF-7</b>	<b>Eliminar Nota de Prensa.</b>
<b>DESCRIPCIÓN</b>	La aplicación debe permitir al usuario eliminar una Nota de Prensa existente.
<b>RF-8</b>	<b>Destacar Nota de Prensa.</b>
<b>DESCRIPCIÓN</b>	La aplicación debe permitir al usuario destacar o no destacar una Nota de Prensa. Si una Nota de Prensa está destacada, se mostrará en la portada de la <b>web corporativa</b> de la UZ.
<b>RF-9</b>	<b>Enviar Nota de Prensa.</b>
<b>DESCRIPCIÓN</b>	La aplicación debe permitir al usuario enviar una Nota de Prensa por correo electrónico a una dirección específica o a una lista de destinatarios según las categorías de <b>Medios de prensa</b> que el usuario desee.
<b>RF-10</b>	<b>Insertar Nota de Prensa en IUNIZAR.</b>
<b>DESCRIPCIÓN</b>	La aplicación debe permitir al usuario insertar la Nota de Prensa en la base de datos del iUnizar.
<b>RF-11</b>	<b>Publicar Nota de Prensa en <a href="http://www.unizar.es">www.unizar.es</a></b>
<b>DESCRIPCIÓN</b>	La aplicación debe permitir al usuario publicar la Nota de Prensa en la web corporativa de la UZ.
<b>RF-12</b>	<b>Generar PDF de la Nota de Prensa.</b>
<b>DESCRIPCIÓN</b>	La aplicación debe permitir al usuario generar un archivo PDF a partir de la Nota de Prensa creada.
<b>RF-13</b>	<b>Listar de Notas de Prensa.</b>
<b>DESCRIPCIÓN</b>	El usuario tendrá acceso al listado de todas las Notas de Prensa donde podrá ver la fecha de publicación y su título. Debajo de cada título habrá una serie de botones que permitirán realizar las

siguientes acciones: editar, duplicar, eliminar, destacar/no destacar, enviar mail, insertar en iUnizar, publicar y generar PDF.

**RF-14 Exportar de Notas de Prensa.**

**DESCRIPCIÓN** La aplicación tendrá la opción de exportar a la web corporativa las últimas Notas de prensa por Autor. El usuario podrá elegir entre dos opciones:

- Especificar el número de Notas de Prensa a exportar por orden de publicación inverso.
- Exportar las notas que aún no hayan sido publicadas en la web corporativa de la UZ.

Creación y gestión de Medios de prensa.

**RF-15 Crear Medios de Prensa.**

**DESCRIPCIÓN** La aplicación permitirá al usuario añadir Medios de Prensa a la base de datos, con los siguientes elementos obligatorios: Nombre del medio, Correo electrónico de Medio y Categorías asociadas al Medio. Además, el usuario podrá incluir opcionalmente los siguientes elementos: Nombre de contacto dentro del Medio y Teléfono de contacto del Medio. Si el usuario no selecciona ninguna categoría asociada, se asignará por defecto la categoría “General” al Medio de Prensa creado.

**RF-16 Listar Medios de Prensa.**

**DESCRIPCIÓN** El usuario tendrá acceso al listado de Medios de Prensa creados donde se mostrarán todos los datos de cada uno a simple vista. Debajo de cada Medio de Prensa se encontrarán los botones “Editar” y “Eliminar”, que permitirán al usuario modificar o borrar el Medio de Prensa correspondiente.

3.3. Requisitos no funcionales.

**RNF-1 Capacidad de envío diario de al menos 5000 correos electrónicos**

**DESCRIPCIÓN** La aplicación debe ser capaz de enviar de forma confiable y eficiente un mínimo de 5000 correos electrónicos diarios. Esta capacidad asegura que la aplicación pueda gestionar una carga de trabajo significativa relacionada con el envío de correos, sin comprometer el rendimiento y garantizando una entrega exitosa de los mensajes.

Este requisito tiene como objetivo satisfacer las necesidades de comunicación del Gabinete de Periodistas de la UZ, brindando una capacidad adecuada para la distribución eficaz de notas de prensa.

**RNF-2****Comunicación segura a través de HTTPS**

La aplicación debe estar configurada para funcionar exclusivamente en un entorno web seguro utilizando el protocolo HTTPS. Esto implica que toda la comunicación entre los usuarios y el servidor estará encriptada, garantizando la confidencialidad, integridad y autenticidad de los datos transmitidos.

La implementación de HTTPS asegura que la aplicación cumpla con los estándares de seguridad y protección de la información, brindando a los usuarios una experiencia confiable y segura al interactuar con la aplicación.

Durante el desarrollo y la configuración, se utilizarán certificados SSL/TLS para habilitar la conexión segura, lo que proporcionará una capa adicional de seguridad al prevenir posibles ataques.

**RNF-3****Compatibilidad con la página corporativa en Drupal para exportar notas de prensa**

La aplicación debe ser compatible con la página corporativa en Drupal de la UZ, permitiendo la exportación de las notas de prensa generadas desde la aplicación hacia dicha plataforma. Esta compatibilidad asegura la integración efectiva entre la aplicación y la página corporativa, facilitando la publicación y difusión de las notas de prensa de manera transparente.

Al cumplir con este requisito, se logra una gestión eficiente de los contenidos y una mayor coherencia en la presencia online de la UZ, asegurando una experiencia de usuario consistente en ambos sistemas.

La exportación de las notas de prensa a la página corporativa en Drupal se realizará a través de un mecanismo basado en formatos de datos estándar, como JSON o CSV, garantizando la interoperabilidad y facilitando la actualización y difusión de la información de manera ágil.

**RNF-4****Capacidad de uso simultáneo por múltiples usuarios**

La aplicación debe tener la capacidad de ser utilizada por varios usuarios de manera simultánea, permitiendo que múltiples usuarios accedan y trabajen con la aplicación al mismo tiempo sin afectar el rendimiento ni la funcionalidad.

Este requisito garantiza que la aplicación pueda manejar la concurrencia de manera efectiva, proporcionando una experiencia de usuario fluida y sin interrupciones, incluso cuando hay un alto número de usuarios interactuando con la aplicación de forma simultánea.

## 4. Análisis.

### 4.1. Identificación de los casos de uso.

Los casos de uso [19] son una técnica clásica y muy efectiva para capturar y detallar, desde el punto de vista del cliente, los diferentes comportamientos y funcionalidades que ofrece una aplicación. La Figura 2, muestra el diagrama de casos de uso de nuestra aplicación, en el que se pueden ver tanto los actores como los casos de uso y sus relaciones. A continuación, se presenta la descripción de los casos de uso identificados.

#### ACCEDER A LA APP

<b>Servicio esperado</b>	Un usuario de la UZ quiere entrar a la aplicación y hacer uso de los servicios de los que esta dispone.
<b>Precondiciones</b>	El usuario no ha iniciado sesión todavía. El usuario debe tener conexión a internet. El usuario debe pertenecer a la UZ.
<b>Frecuencia de uso</b>	Este caso de uso se da cada vez que un usuario registrado quiere acceder al sistema.
<b>Actores</b>	Usuario
<b>Flujo de eventos</b>	<ol style="list-style-type: none"><li>1. Un usuario de la UZ abre la aplicación.</li><li>2. El sistema solicita los datos de inicio de sesión.</li><li>3. El usuario introduce datos de sesión.</li><li>4. &lt;&lt;include&gt;&gt; El sistema valida correctamente al usuario.</li><li>5. El usuario tiene acceso al sistema.</li></ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"><li>1. El usuario no introduce de manera correcta los datos de inicio de sesión.</li><li>2. El sistema no deja acceder al usuario.</li></ol>

#### RECIBIR ROL

<b>Servicio esperado</b>	Un usuario de la UZ quiere recibir un rol dentro de la aplicación.
<b>Precondiciones</b>	El usuario debe de haber iniciado sesión correctamente en la aplicación. El usuario debe pertenecer a la UZ.
<b>Frecuencia de uso</b>	Este caso de uso se da cada vez que un nuevo usuario quiera hacer uso de la aplicación o cuando se deba reasignar un rol a un usuario existente.
<b>Actores</b>	Usuario y Administrador
<b>Flujo de eventos</b>	<ol style="list-style-type: none"><li>1. Un usuario contacta con un administrador de la aplicación para que este le asigne un rol.</li><li>2. El administrador comprueba los usuarios registrados en la aplicación y le asigna el rol que le corresponde.</li><li>3. El usuario tiene acceso a todos los servicios que ofrece su rol.</li></ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"><li>1. Un usuario contacta con un administrador de la aplicación para que este le asigne un rol.</li><li>2. El administrador comprueba los usuarios registrados en la aplicación y no le asigna ningún rol.</li></ol>

3. El usuario no tiene acceso a ningún servicio dentro de la aplicación.

### **VALIDAR CREDENCIALES**

<b>Servicio esperado</b>	El sistema identifica al usuario y comprueba que sea un usuario de la UZ.
<b>Precondiciones</b>	El usuario debe haber intentado iniciar sesión. El usuario debe tener conexión a internet.
<b>Frecuencia de uso</b>	Se realiza cada vez que se da un inicio de sesión.
<b>Actores</b>	Usuario y BBDD usuarios.
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El sistema que las credenciales introducidas por el usuario coincidan con las de la BBDD usuarios.</li> <li>2. El sistema concede el acceso a la aplicación al usuario si las credenciales introducidas coinciden con las de la BBDD usuarios.</li> </ol>

### **LISTAR NOTAS DE PRENSA / CONVOCATORIAS PARA LOS MEDIOS**

<b>Servicio esperado</b>	Un usuario desea visualizar los títulos de todas las notas de prensa de la aplicación.
<b>Precondiciones</b>	El usuario debe tener conexión a internet. El usuario debe de haber iniciado sesión correctamente. El usuario debe tener asignado un rol dentro de la aplicación.
<b>Frecuencia de uso</b>	Este caso de uso se da cada vez que el usuario accede a la pestaña de “Listar notas”
<b>Actores</b>	Usuario.
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. Un usuario hace click sobre el botón “Listar notas”.</li> <li>2. El sistema le devuelve el listado de notas de prensa.</li> </ol>

### **FILTRAR NOTAS DE PRENSA DESTACADAS**

<b>Servicio esperado</b>	Un usuario desea visualizar los títulos de todas las notas de prensa destacadas de la aplicación.
<b>Precondiciones</b>	El usuario debe tener conexión a internet. El usuario debe de haber iniciado sesión correctamente. El usuario debe tener asignado un rol dentro de la aplicación.
<b>Frecuencia de uso</b>	Este caso de uso se da cada vez que el usuario pulsa el botón “Filtrar destacadas” dentro de la pestaña “Listar notas”.
<b>Actores</b>	Usuario.
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. Un usuario pulsa sobre el botón “Filtrar destacadas”.</li> <li>2. El sistema le devuelve el listado de notas de prensa destacadas.</li> </ol>

### **CREAR NOTA DE PRENSA / CONVOCATORIA PARA LOS MEDIOS**

<b>Servicio esperado</b>	El usuario podrá crear un nuevo registro de nota de prensa o convocatoria para los medios.
<b>Precondiciones</b>	El usuario debe tener conexión a internet. El usuario debe de haber iniciado sesión correctamente.

	El usuario debe tener asignado un rol dentro de la aplicación.
<b>Frecuencia de uso</b>	Este caso de uso se da cada vez que el usuario le da al botón de “Nueva nota” o “Nueva convocatoria”
<b>Actores</b>	Usuario
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. Un usuario pulsa el botón “Nueva nota” o “Nueva convocatoria”.</li> <li>2. El sistema le devuelve al usuario un formulario de creación de nota.</li> <li>3. El usuario rellena los campos del formulario.</li> <li>4. El usuario pulsa el botón “Guardar”.</li> <li>5. El sistema comprueba que los datos del formulario sean correctos.</li> <li>6. Una nueva nota de prensa se ha creado.</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. El sistema detecta que el formulario no se ha rellenado correctamente.</li> <li>2. El sistema notifica al usuario los campos del formulario que han sido rellenado erróneamente para que este los corrija.</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón “Cancelar”.</li> <li>2. El sistema lleva al usuario a la pestaña “Listar notas” de nuevo.</li> <li>3. Todo lo que el usuario hubiera escrito en el formulario se pierde.</li> </ol>

## EDITAR NOTA DE PRENSA / CONVOCATORIA PARA LOS MEDIOS

<b>Servicio esperado</b>	El usuario podrá modificar un registro existente de una nota de prensa o convocatoria para los medios.
<b>Precondiciones</b>	<p>El usuario debe tener conexión a internet.</p> <p>El usuario debe de haber iniciado sesión correctamente.</p> <p>El usuario debe tener asignado un rol dentro de la aplicación.</p> <p>La nota/convocatoria a modificar debe existir en la aplicación.</p>
<b>Frecuencia de uso</b>	Se da este caso de uso cada vez que un usuario pulsa el botón “Editar” correspondiente a un registro.
<b>Actores</b>	Usuario
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. Un usuario pulsa el botón “Editar” que se encuentra debajo del título de la nota en la pestaña “Listar notas”.</li> <li>2. El sistema devuelve al usuario un formulario de nota relleno con los datos guardados en la BBDD.</li> <li>3. El usuario edita los campos que considere del formulario.</li> <li>4. El usuario pulsa el botón “Guardar nota”.</li> <li>5. El sistema comprueba que los datos del formulario sean correctos.</li> <li>6. La nueva versión de la nota de guarda en la BBDD.</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. El sistema detecta que el formulario no se ha rellenado correctamente.</li> <li>2. El sistema notifica al usuario los campos del formulario que han sido rellenado erróneamente para que este los corrija.</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón “Cancelar”.</li> </ol>

2. El sistema lleva al usuario a la pestaña “Listar notas” de nuevo.
3. Todo lo que el usuario hubiera escrito en el formulario se pierde.

### **ELIMINAR NOTA DE PRENSA / CONVOCATORIA PARA LOS MEDIOS**

**Servicio esperado** El usuario podrá eliminar un registro existente de una nota de prensa o convocatoria para los medios.

**Precondiciones** El usuario debe tener conexión a internet.  
El usuario debe de haber iniciado sesión correctamente.  
El usuario debe tener asignado un rol dentro de la aplicación.  
La nota/convocatoria a eliminar debe existir en la aplicación.

**Frecuencia de uso** Se da este caso de uso cada vez que un usuario pulsa el botón “Eliminar” correspondiente a un registro.

**Actores** Usuario.

**Flujo de eventos**

1. Un usuario pulsa el botón “Eliminar” que se encuentra debajo del título de la nota en la pestaña “Listar notas”.
2. El sistema devuelve al usuario la previsualización de la nota.
3. El usuario pulsa el botón “Eliminar”.
4. La nota es borrada de la BBDD.

**Flujo alternativo**

1. El usuario pulsa el botón “Cancelar”.
2. El sistema lleva al usuario a la pestaña “Listar notas” de nuevo.
3. La nota permanece en la BBDD.

### **DESTACAR NOTA DE PRENSA / CONVOCATORIA PARA LOS MEDIOS**

**Servicio esperado** El usuario podrá destacar un registro existente de una nota de prensa o convocatoria para los medios.

**Precondiciones** El usuario debe tener conexión a internet.  
El usuario debe de haber iniciado sesión correctamente.  
El usuario debe tener asignado un rol dentro de la aplicación.  
La nota/convocatoria a destacar debe existir en la aplicación.

**Frecuencia de uso** Se da este caso de uso cada vez que un usuario pulsa el botón “Destacar” correspondiente a un registro.

**Actores** Usuario

**Flujo de eventos**

1. Un usuario pulsa el botón “Destacar” que se encuentra debajo del título de la nota en la pestaña “Listar notas”.
2. El sistema marca esa nota como destacada en la BBDD.

### **DUPLICAR NOTA DE PRENSA / CONVOCATORIA PARA LOS MEDIOS**

**Servicio esperado** El usuario podrá duplicar un registro existente de una nota de prensa o convocatoria para los medios.

**Precondiciones** El usuario debe tener conexión a internet.  
El usuario debe de haber iniciado sesión correctamente.  
El usuario debe tener asignado un rol dentro de la aplicación.  
La nota/convocatoria a duplicar debe existir en la aplicación.

**Frecuencia de uso** Se da este caso de uso cada vez que un usuario pulsa el botón “Duplicar” correspondiente a un registro.

<b>Actores</b>	Usuario.
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. Un usuario pulsa el botón “Duplicar” que se encuentra debajo del título de la nota en la pestaña “Listar notas”.</li> <li>2. El sistema devuelve al usuario la previsualización de la nota.</li> <li>3. El usuario pulsa el botón “Duplicar”.</li> <li>4. La nota es duplicada en la BBDD.</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón “Cancelar”.</li> <li>2. El sistema lleva al usuario a la pestaña “Listar notas” de nuevo.</li> <li>3. La nota no se duplica en la BBDD.</li> </ol>

### **GENERAR PDF DE LA NOTA DE PRENSA / CONVOCATORIA PARA LOS MEDIOS**

<b>Servicio esperado</b>	El usuario podrá descargar en formato pdf una nota de prensa o convocatoria para los medios existente.
<b>Precondiciones</b>	<p>El usuario debe tener conexión a internet.</p> <p>El usuario debe de haber iniciado sesión correctamente.</p> <p>El usuario debe tener asignado un rol dentro de la aplicación.</p> <p>La nota/convocatoria de la que se genera el PDF debe existir en la aplicación.</p>
<b>Frecuencia de uso</b>	Se da este caso de uso cada vez que un usuario pulsa el botón “Generar PDF” correspondiente a un registro.

<b>Actores</b>	Usuario.
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. Un usuario pulsa el botón “Generar PDF” que se encuentra debajo del título de la nota de prensa en la pestaña “Listar notas”.</li> <li>2. Se inicia una descarga automática del PDF de la nota.</li> </ol>

### **INSERTAR NOTA DE PRENSA / CONVOCATORIA PARA LOS MEDIOS EN EL IUNIZAR**

<b>Servicio esperado</b>	El usuario podrá insertar una nota de prensa o convocatoria para los medios en la BBDD iUnizar.
<b>Precondiciones</b>	<p>El usuario debe tener conexión a internet.</p> <p>El usuario debe de haber iniciado sesión correctamente.</p> <p>El usuario debe tener asignado un rol dentro de la aplicación.</p> <p>La nota/convocatoria a insertar debe existir en la aplicación.</p>
<b>Frecuencia de uso</b>	Se da este caso de uso cada vez que un usuario pulsa el botón “Insertar iUnizar” correspondiente a un registro.

<b>Actores</b>	Usuario.
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. Un usuario pulsa el botón “Insertar iUnizar” que se encuentra debajo del título de la nota de prensa en la pestaña “Listar notas”.</li> <li>2. La nota se crea en la BBDD iUnizar.</li> </ol>

## **EXPORTAR NOTAS DE PRENSA / CONVOCATORIAS PARA LOS MEDIOS**

<b>Servicio esperado</b>	El usuario podrá exportar un listado de notas de prensa o convocatorias.
<b>Precondiciones</b>	El usuario debe tener conexión a internet. El usuario debe de haber iniciado sesión correctamente. El usuario debe tener asignado un rol dentro de la aplicación. Las notas/convocatorias a exportar deben existir en la aplicación.
<b>Frecuencia de uso</b>	Se da este caso de uso cada vez que un usuario accede a la pestaña “Exportar notas” e inicia el procedimiento de exportación.
<b>Actores</b>	Usuario.
<b>Flujo de eventos</b>	<ol style="list-style-type: none"><li>1. Un usuario accede a la pestaña “Exportar notas”</li><li>2. El usuario selecciona los autores de los cuales quiere exportar las notas.</li><li>3. El usuario selecciona si quiere exportar no publicadas o los últimos registros creados. En el caso de seleccionar la última opción, debe especificar el número de notas que quiere exportar.</li><li>4. El usuario presiona el botón “Continuar”.</li><li>5. El sistema comprueba que el usuario hecho la solicitud de exportación correctamente.</li><li>6. El sistema le devuelve al usuario un listado con los títulos de las notas que se van a exportar con su correspondiente ID.</li><li>7. El usuario puede desmarcar las notas que desee para que no se incluyan en la exportación.</li><li>8. El usuario pulsa el botón “Exportar”.</li><li>9. El sistema comprueba que hay al menos una nota marcada para la exportación.</li><li>10. El sistema genera un JSON y un CSV con los datos de las notas.</li></ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"><li>1. El usuario pulsa el botón “Cancelar” en lugar de “Exportar”.</li><li>2. El sistema lleva al usuario a la pestaña “Listar notas” de nuevo.</li></ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"><li>1. El sistema detecta que el usuario no ha realizado la solicitud de exportación correctamente.</li><li>2. El sistema notifica al usuario la razón por la cual no ha realizado la solicitud de exportación correctamente.</li></ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"><li>1. El sistema detecta que no hay ninguna nota marcada para la exportación.</li><li>2. El sistema notifica al usuario que para la exportación al menos debe de haber una nota marcada.</li></ol>

## **ENVIAR NOTA DE PRENSA / CONVOCATORIA PARA LOS MEDIOS**

<b>Servicio esperado</b>	El usuario podrá enviar una nota de prensa o convocatoria para los medios a un listado de medios de prensa.
<b>Precondiciones</b>	El usuario debe tener conexión a internet. El usuario debe de haber iniciado sesión correctamente.

El usuario debe tener asignado un rol dentro de la aplicación.  
La nota/convocatoria a enviar deben existir en la aplicación.

**Frecuencia de uso** Se da este caso de uso cada vez que un usuario pulsa el botón “Enviar mail” correspondiente a un registro.

**Actores** Usuario

**Flujo de eventos**

1. Un usuario pulsa el botón “Enviar mail” que se encuentra debajo del título de la nota de prensa en la pestaña “Listar notas”.
2. El sistema le devuelve al usuario la vista previa de la nota que quiere enviar.
3. El usuario selecciona la categoría de los medios a los que quiere enviar la nota.
4. El usuario introduce manualmente los correos independientes a los que quiera mandar la nota.
5. El usuario pulsa el botón “Enviar”.
6. El sistema comprueba que el usuario haya seleccionado destinatarios.
7. El sistema procede con el envío de la nota de prensa.

**Flujo alternativo**

1. El sistema detecta que no hay ningún destinatario seleccionado.
2. El sistema notifica al usuario del error detectado.

## LISTAR MEDIOS DE PRENSA

**Servicio esperado** Un usuario desea visualizar todos los medios de prensa almacenados en la BBDD de la aplicación.

**Precondiciones** El usuario debe tener conexión a internet.  
El usuario debe de haber iniciado sesión correctamente.  
El usuario debe tener asignado un rol dentro de la aplicación.

**Frecuencia de uso** Este caso de uso se da cada vez que el usuario accede a la pestaña de “Listar medios”

**Actores** Usuario.

**Flujo de eventos**

1. Un usuario hace click sobre el botón “Listar notas”.
2. El sistema le devuelve el listado de notas de prensa.

## CREAR MEDIO DE PRENSA

**Servicio esperado** Un usuario podrá crear un nuevo medio de prensa en la BBDD.

**Precondiciones** El usuario debe tener conexión a internet.  
El usuario debe de haber iniciado sesión correctamente.  
El usuario debe tener asignado un rol dentro de la aplicación

**Frecuencia de uso** Este caso de uso se dará cada vez que el usuario pulse sobre el botón “Nuevo medio”.

**Actores** Usuario.

**Flujo de eventos**

1. Un usuario pulsa el botón “Nuevo medio”.
2. El sistema le devuelve al usuario un formulario de creación de medio.
3. El usuario rellena los campos del formulario.
4. El usuario pulsa el botón “Guardar”.

	<ol style="list-style-type: none"> <li>5. El sistema comprueba que los datos del formulario sean correctos.</li> <li>6. Un nuevo medio de prensa se ha creado.</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. El sistema detecta que el formulario no se ha rellenado correctamente.</li> <li>2. El sistema notifica al usuario los campos del formulario que han sido rellenado erróneamente para que este los corrija.</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón “Cancelar”.</li> <li>2. El sistema lleva al usuario a la pestaña “Listar medios” de nuevo.</li> <li>3. Todo lo que el usuario hubiera escrito en el formulario se pierde.</li> </ol>

### **EDITAR MEDIO DE PRENSA**

<b>Servicio esperado</b>	El usuario podrá modificar un registro existente de un medio de prensa.
<b>Precondiciones</b>	<p>El usuario debe tener conexión a internet.</p> <p>El usuario debe de haber iniciado sesión correctamente.</p> <p>El usuario debe tener asignado un rol dentro de la aplicación.</p> <p>El medio de prensa a modificar debe existir en la aplicación.</p>
<b>Frecuencia de uso</b>	Se da este caso de uso cada vez que un usuario pulsa el botón “Editar” correspondiente a un registro.
<b>Actores</b>	Usuario
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. Un usuario pulsa el botón “Editar” que se encuentra debajo del título del medio en la pestaña “Listar medios”.</li> <li>2. El sistema devuelve al usuario un formulario de medio relleno con los datos guardados en la BBDD.</li> <li>3. El usuario edita los campos que considere del formulario.</li> <li>4. El usuario pulsa el botón “Guardar medio”.</li> <li>5. El sistema comprueba que los datos del formulario sean correctos.</li> <li>6. La nueva versión del medio se guarda en la BBDD.</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. El sistema detecta que el formulario no se ha rellenado correctamente.</li> <li>2. El sistema notifica al usuario los campos del formulario que han sido rellenado erróneamente para que este los corrija.</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón “Cancelar”.</li> <li>2. El sistema lleva al usuario a la pestaña “Listar medios” de nuevo.</li> <li>3. Todo lo que el usuario hubiera escrito en el formulario se pierde.</li> </ol>

### **ELIMINAR MEDIO DE PRENSA**

<b>Servicio esperado</b>	El usuario podrá eliminar un registro existente de un medio de prensa.
<b>Precondiciones</b>	<p>El usuario debe tener conexión a internet.</p> <p>El usuario debe de haber iniciado sesión correctamente.</p> <p>El usuario debe tener asignado un rol dentro de la aplicación.</p>

	El medio de prensa a eliminar debe existir en la aplicación.
<b>Frecuencia de uso</b>	Se da este caso de uso cada vez que un usuario pulsa el botón “Eliminar” correspondiente a un registro.
<b>Actores</b>	Usuario.
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. Un usuario pulsa el botón “Eliminar” que se encuentra debajo del título del medio en la pestaña “Listar medios”.</li> <li>2. El sistema devuelve al usuario la previsualización del medio.</li> <li>3. El usuario pulsa el botón “Eliminar”.</li> <li>4. El medio es borrado de la BBDD.</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón “Cancelar”.</li> <li>2. El sistema lleva al usuario a la pestaña “Listar medios” de nuevo.</li> <li>3. El medio permanece en la BBDD.</li> </ol>

### ACCEDER A LOS LOGS DE LA APLICACIÓN

<b>Servicio esperado</b>	El usuario podrá acceder a los logs de la aplicación.
<b>Precondiciones</b>	<p>El usuario debe tener conexión a internet.</p> <p>El usuario debe de haber iniciado sesión correctamente.</p> <p>El usuario debe tener asignado un rol de superusuario o administrador.</p>
<b>Frecuencia de uso</b>	Se da este caso de uso cada vez que un usuario accede al panel de administración.
<b>Actores</b>	Usuario
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. Un usuario pulsa sobre el botón “Panel de Administración”</li> <li>2. El sistema le devuelve al usuario un listado con los logs de la aplicación en orden cronológico.</li> </ol>

### REALIZAR BUSQUEDA EN LA BASE DE DATOS

<b>Servicio esperado</b>	El usuario podrá realizar búsqueda entre las notas de prensa y convocatorias para los medios almacenadas en la BBDD.
<b>Precondiciones</b>	<p>El usuario debe tener conexión a internet.</p> <p>El usuario debe de haber iniciado sesión correctamente.</p> <p>El usuario debe tener asignado un rol.</p>
<b>Frecuencia de uso</b>	Se da este caso de uso cada vez que el usuario acceda a la pestaña “Búsqueda”.
<b>Actores</b>	Usuario.
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. Un usuario pulsa sobre “Búsqueda”.</li> <li>2. El sistema le devuelve un formulario al usuario donde puede introducir parámetros de búsqueda.</li> <li>3. El usuario rellena este formulario.</li> <li>4. El usuario pulsa sobre el botón “Buscar”.</li> <li>5. El sistema comprueba que el formulario este relleno correctamente.</li> <li>6. El sistema le devuelve al usuario los resultados de búsqueda.</li> </ol>

**Flujo alternativo**

1. El sistema detecta que el formulario ha sido relleno de forma errónea.
2. El sistema informa al usuario de los fallos detectados.

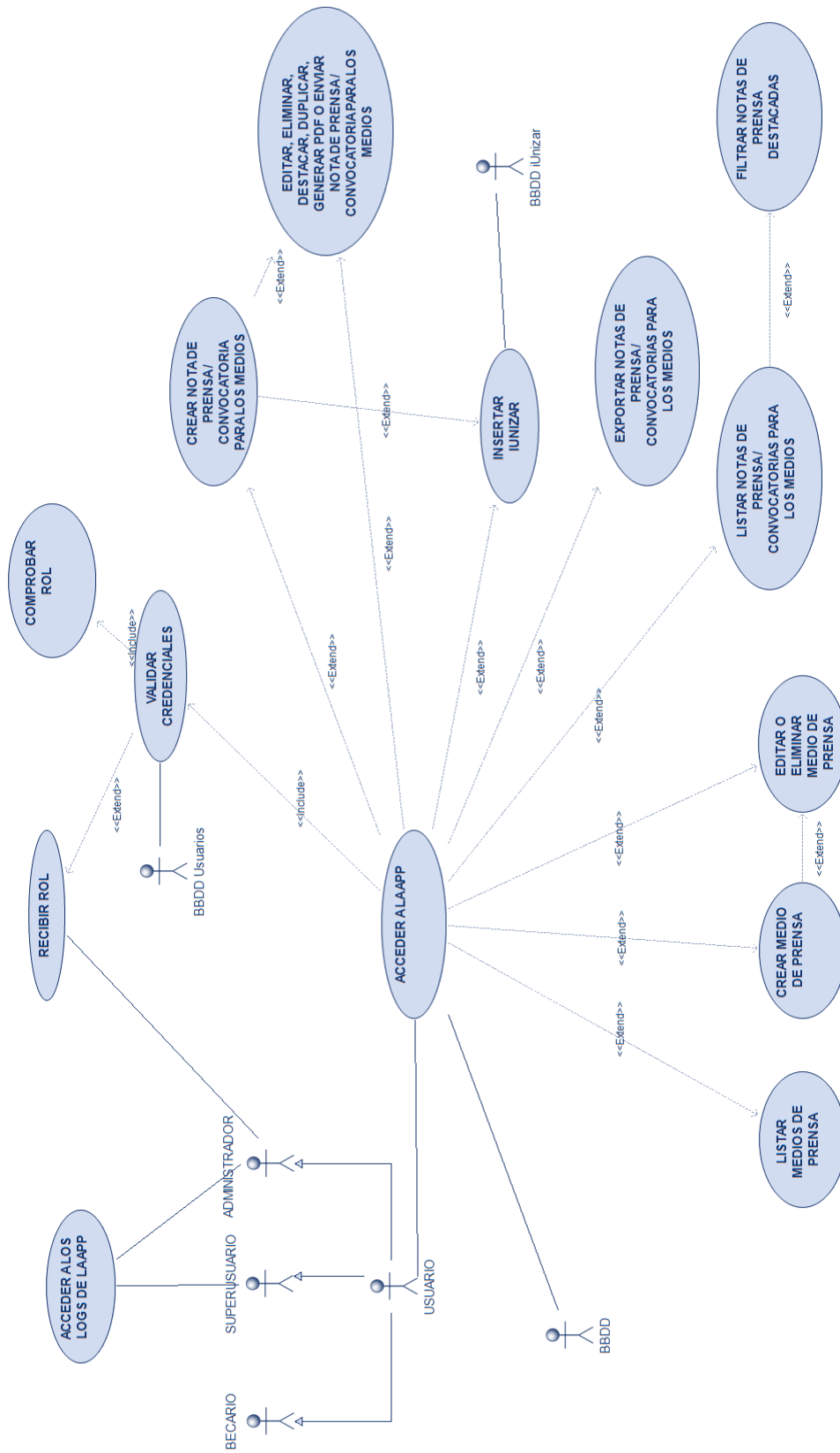


Figura 2. Diagrama de casos de uso.

## 4.2. Modelado de datos.

El modelado de datos desempeña un papel crucial en el proceso de diseño de software, ya que permite representar de manera estructurada y visual las entidades y las relaciones que conforman el sistema. En este apartado, se presenta el diagrama de clases que conforma el análisis de nuestra aplicación. Para realizar este diagrama hemos utilizado el Lenguaje de Modelado Unificado (UML) [20]. El mismo se presenta en la Figura 3.

En el diseño orientado a objetos, una clase se define como una abstracción de un objeto del mundo real que o bien pertenece al sistema o interactúa con el mismo. Cada clase encapsula tanto los datos como los comportamientos relacionados, lo que favorece la legibilidad del código. Al definir una clase, se identifican y describen los atributos que almacenan información relevante y los métodos que definen las operaciones realizadas por la clase. Las clases tienen conocimiento entre sí a través de sus relaciones, que pueden ser de asociación o herencia. Las asociaciones definen relaciones estructurales entre las clases y permiten el acceso a las funcionalidades ofrecidas. Por su parte, la herencia es un mecanismo de reutilización de código.

En el Anexo A, se examinarán las clases pertinentes a la aplicación, proporcionando una descripción detallada de cada una de ellas. Se detallarán los atributos específicos de cada clase, resaltando la información que se almacena en ellos y su relevancia dentro del sistema. Asimismo, se describirán los métodos asociados a cada clase, indicando las operaciones que pueden realizar y su propósito en el contexto de la aplicación.

Seguidamente, se procede a hacer una breve descripción de cada una de las clases identificadas para el desarrollo de la aplicación:

- Usuario: Individuo perteneciente a la UZ que se ha autenticado correctamente en la aplicación.
- Autor: Hace referencia al departamento responsable de la elaboración de la nota.
- Nota: Documento utilizado para difundir información relacionada con la UZ.
- Imagen: Elemento visual que complementa la nota de prensa.
- Medio: Destinatario almacenado en la base de datos de la aplicación al cual se le puede enviar la nota de prensa.
- Categoría: Grupos a los que los medios de prensa pueden pertenecer dentro de la aplicación, para agilizar el proceso de envío.
- Envío: Almacena información sobre el envío de una nota para su posterior revisión.
- Exportación: Almacena información sobre la exportación de un conjunto de notas para su posterior revisión.
- Log: Almacena información sobre cada acción realizada en la aplicación para su posterior revisión.

En la Figura 3, podemos observar un diagrama de clases. Este diagrama permite identificar las clases del sistema, así como las propiedades y métodos asociados a cada una. Además, muestra las relaciones entre las clases, como asociaciones, composiciones, herencias y dependencias.

Una de las ventajas del diagrama de clases es que nos proporciona información sobre la multiplicidad entre las instancias de las clases. Esto significa que podemos determinar cuántas instancias de una clase pueden estar relacionadas con otras, ya sea en uno a uno, uno a muchos o muchos a muchos.

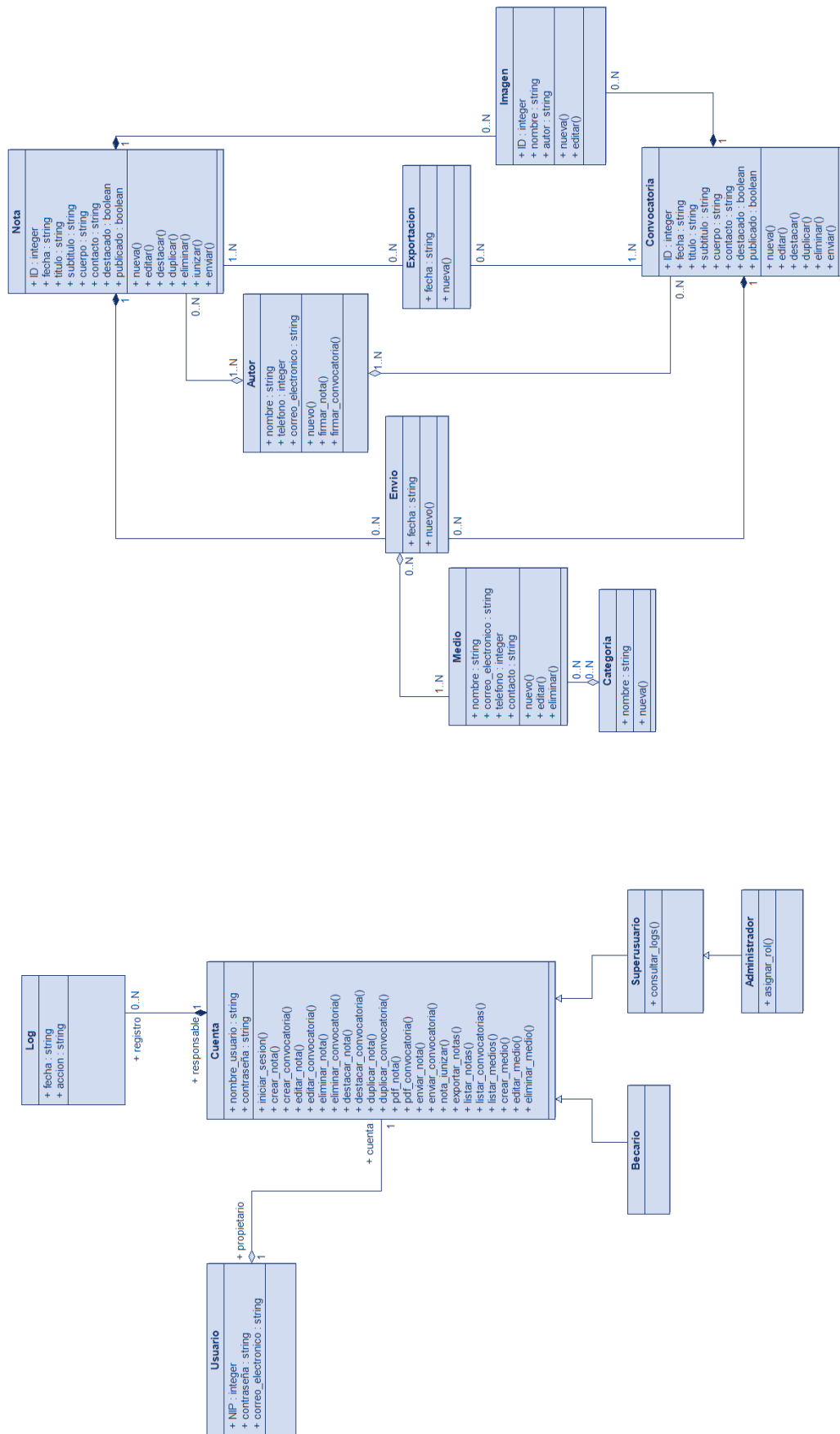


Figura 3. Diagrama de clases.

Diagrama de objetos.

Los diagramas de objetos son útiles para comprender la interacción de los objetos durante la ejecución de un sistema. En la Figura 4, se ilustra el funcionamiento de los logs dentro de la aplicación y cómo cada cuenta puede tener varios logs asociados. Además, se muestra cómo los roles son utilizados por los usuarios.

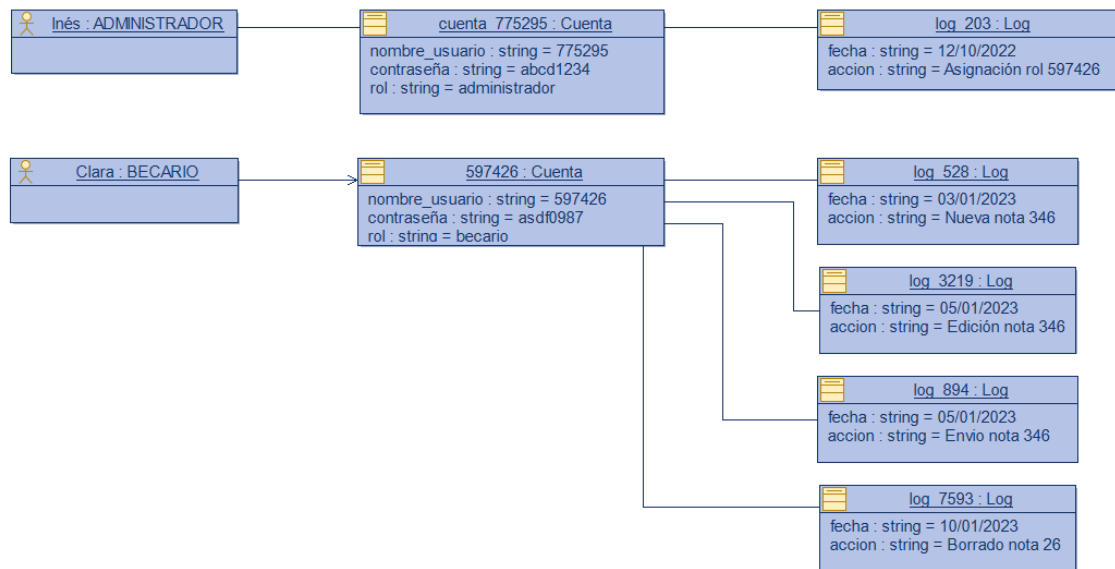


Figura 4. Diagrama de objetos.

En la Figura 5, se representa otro diagrama de objetos que muestra cómo una nota puede tener varias instancias de la clase imagen, las cuales no son obligatorias. Asimismo, se observa que las exportaciones pueden incluir más de una nota, mientras que los envíos están relacionados siempre con una única nota, pero pueden tener uno o más medios como destinatarios, los cuales pueden agruparse en categorías. Además, se aprecia que una nota debe tener al menos un autor, pero existe la posibilidad de tener múltiples autores.

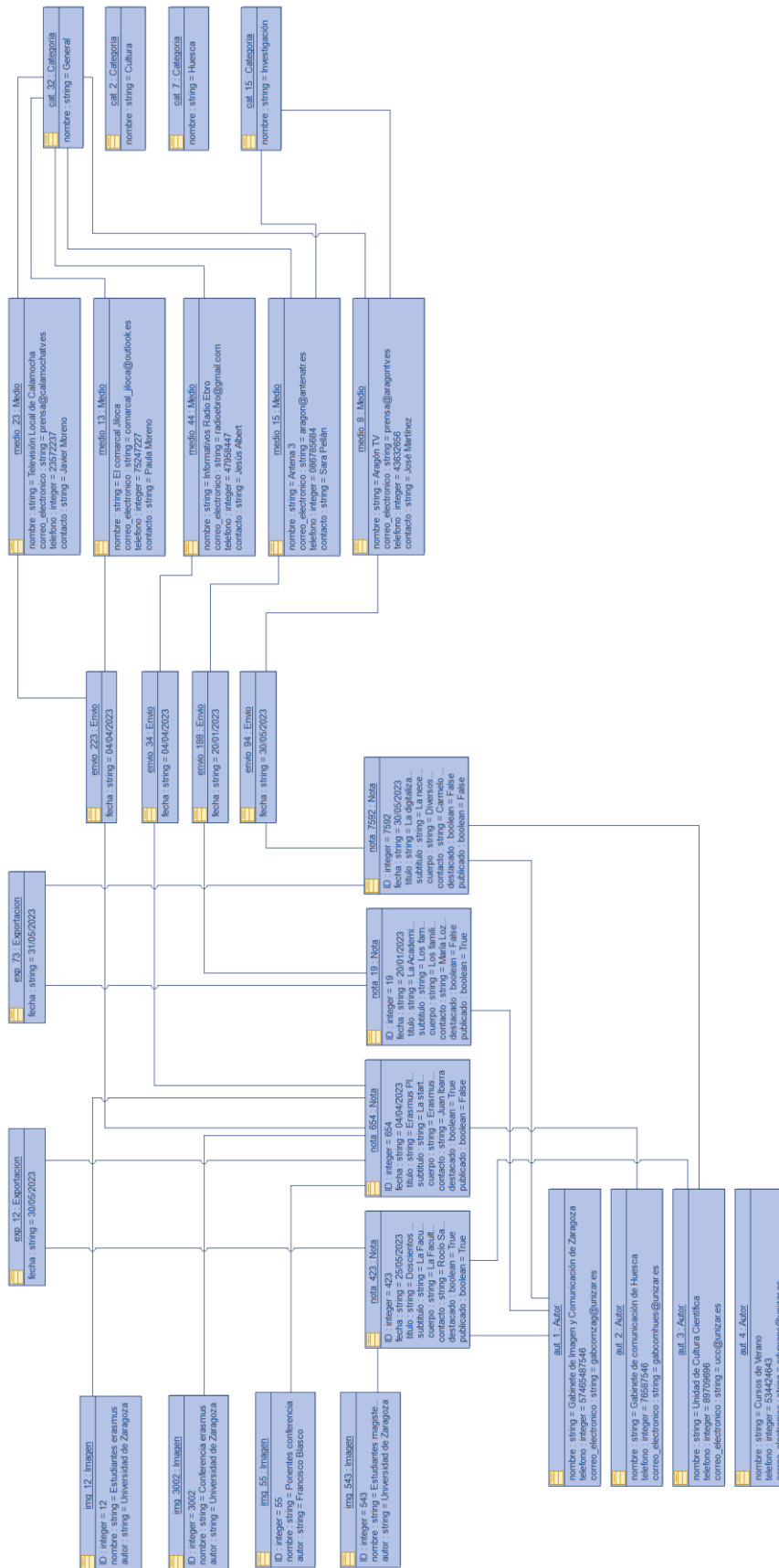


Figura 5. Diagrama de objetos.

### 4.3. Análisis dinámico.

Este apartado se centra en el análisis dinámico de UML, este se enfoca en comprender el flujo de información, los eventos y las transiciones que ocurren durante la ejecución del sistema.

Dentro del análisis dinámico, los diagramas de secuencia desempeñan un papel crucial. Estos diagramas representan la secuencia de interacciones entre los objetos a lo largo del tiempo. Muestran cómo los objetos se comunican entre sí, qué mensajes se intercambian y en qué orden.

En este apartado, se explicará en detalle el diagrama de secuencia más complejo. El resto de los diagramas de secuencia serán abordados en el Anexo B.

En la Figura 6, se presenta un diagrama de secuencia que ilustra el proceso de exportación de notas. En este diagrama, se observa cómo un usuario interactúa con la aplicación a través de formularios, permitiendo a la aplicación solicitar los datos necesarios a la base de datos y determinar qué notas deben ser exportadas. Una vez que el usuario confirma la lista de notas a exportar, la aplicación crea una nueva instancia de la clase exportación, incluyendo toda la información relevante.

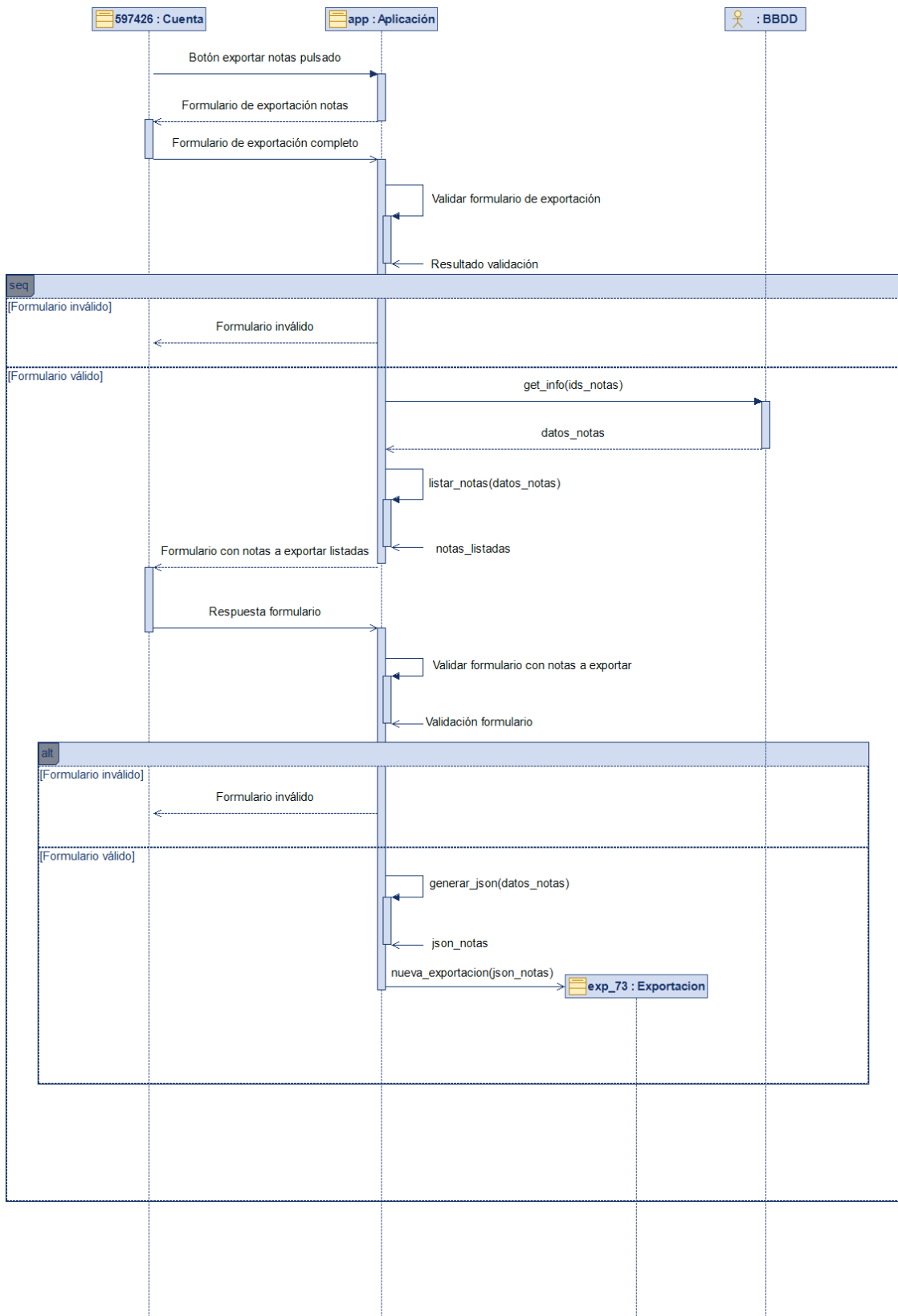


Figura 6. Diagrama de secuencia – Exportación notas.

Para concluir el análisis dinámico, se procederá con la presentación de un diagrama de actividades. Este tipo de diagrama en UML representa el flujo de actividades y decisiones en un proceso o procedimiento. El diagrama de actividades es útil para visualizar la lógica de los flujos de trabajo y las rutinas operativas.

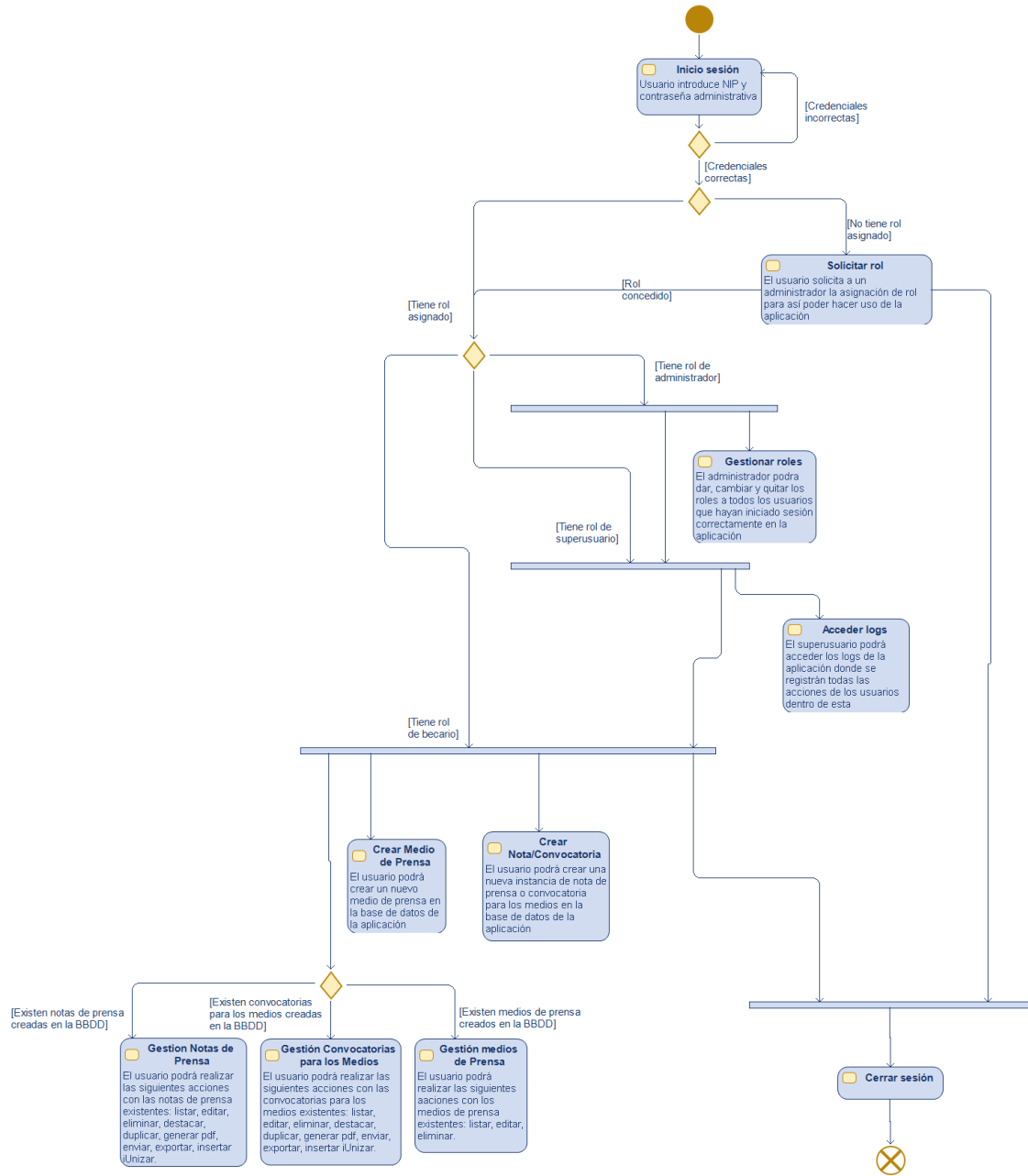


Figura 6. Diagrama de actividades.

## 5. Diseño.

### 5.1. Diseño de la arquitectura.

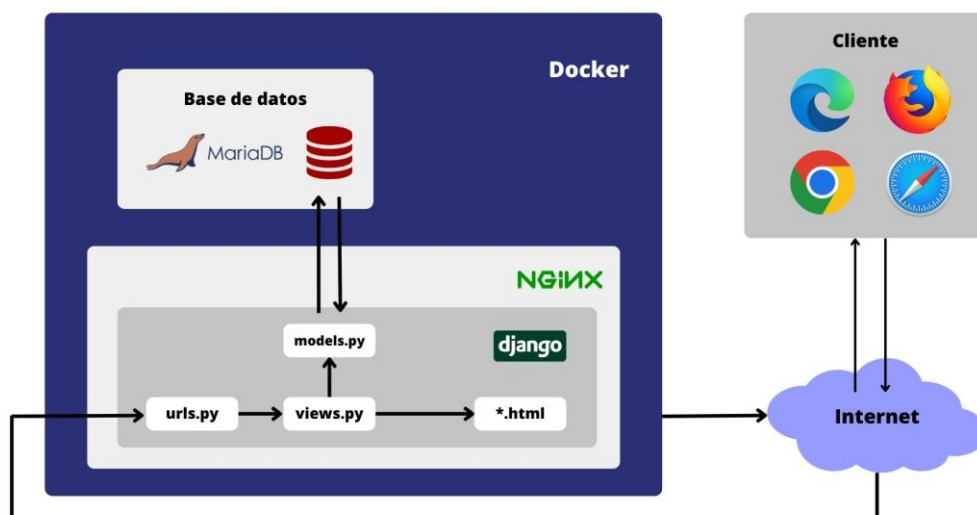


Figura 7. Esquema arquitectura aplicación.

La arquitectura de la aplicación, tal y como se muestra en la Figura 7, está formada por los siguientes componentes:

- Cliente: Es una aplicación que se carga en el navegador web y realiza las peticiones al servidor web.
- Aplicación de notas de prensa: Es una aplicación desarrollada en Django encargada de gestionar y responder las peticiones del cliente.
- Gestor de base de datos: Administra la información utilizada tanto por el cliente como por el servidor. En este caso, se utiliza MariaDB.
- Nginx [21]: Es un servidor web donde se aloja la aplicación de notas de prensa.
- HTTPS: Es un protocolo seguro de comunicación entre el cliente y el servidor.

### 5.2. Estructura de la aplicación.

Para lograr la creación de la estructura de la aplicación, se ha organizado cada una de las funciones solicitadas por el cliente en microservicios individuales. Cada microservicio tiene un propósito específico y opera con sus propios conjuntos de datos, a los cuales otras aplicaciones pueden acceder para llevar a cabo sus tareas.

Al dividir las funciones en microservicios, se establece una clara separación de responsabilidades y se permite un desarrollo y mantenimiento más eficiente. Cada microservicio puede centrarse en su área específica de trabajo y utilizar los datos necesarios sin depender de otros microservicios. Esto mejora la modularidad y facilita la

escalabilidad del sistema, ya que se pueden agregar nuevas funcionalidades sin afectar a las existentes.

Además, al permitir el acceso controlado a los datos compartidos, se evita la duplicación innecesaria y la inconsistencia en la gestión de estos.



*Figura 8. Microservicios aplicación.*

A continuación, se presenta una breve descripción de cada una de las aplicaciones:

- API: Esta aplicación se utiliza cuando un usuario accede a la pestaña de Exportar notas y completa los formularios correspondientes. Una vez que la aplicación recopila la información de las notas a exportar, genera un archivo JSON que la página institucional de la UZ puede utilizar como fuente de datos.
- Autenticación: Esta aplicación solicita al usuario las credenciales de acceso y verifica su validez consultando la base de datos de usuarios.
- Búsqueda: Esta aplicación proporciona al usuario un formulario con diversos filtros de búsqueda. Al recibir el formulario completo, realiza una consulta en la base de datos y muestra los resultados de búsqueda al usuario.
- Envío de correos electrónicos: Esta aplicación solicita al usuario los destinatarios del correo electrónico que acompaña a las notas de prensa o convocatorias dirigidas a los medios. Se encarga de dar formato al correo y enviarlo a todos los destinatarios especificados.
- Gestión de convocatorias: Esta aplicación se encarga de la creación y gestión de convocatorias dentro del sistema.
- Gestión de notas: Esta aplicación se encarga de la creación y gestión de notas dentro del sistema.
- Gestión de medios: Esta aplicación se encarga de la creación y gestión de los medios de comunicación dentro del sistema.

### 5.3. Diseño de la interfaz de usuario.

A continuación, se muestra en la Figura 9 una captura de pantalla de la aplicación en la pestaña donde se cumplen la mayoría de los requisitos funcionales. En el Anexo E se pueden encontrar el resto de las capturas de la aplicación donde se cumplen los demás requisitos. En la captura, se observa cómo se cumplen los siguientes requisitos: RF1 (Crear Nota de Prensa), RF5 (Editar Nota de Prensa), RF6 (Duplicar Nota de Prensa), RF7 (Eliminar Nota de Prensa), RF8 (Destacar Nota de Prensa), RF9 (Enviar Nota de Prensa), RF10 (Insertar Nota de Prensa en IUNIZAR), RF11 (Publicar Nota de Prensa en [www.unizar.es](http://www.unizar.es)), RF12 (Generar PDF de la Nota de Prensa) y RF13 (Listar Notas de Prensa). Estos requisitos fueron explicados previamente en el apartado 3.2.

Universidad Zaragoza

Notas Convocatorias Medios Búsqueda

Nueva nota Filtrar destacadas

08-06-2023 ☆ fvfv PUBLICADO  
Editar | Duplicar | Eliminar | Insertar iUnizar | Destacar | Enviar mail | Generar PDF | Auditoría envíos |

18-05-2023 ☆ **Presentado el nuevo Reglamento de Enseñanzas Propias de la Universidad de Zaragoza** PUBLICADO  
Editar | Duplicar | Eliminar | Insertar iUnizar | Destacar | Enviar mail | Generar PDF | Auditoría envíos |

04-05-2023 ☆ **Huesca celebra este miércoles el Día del Juego en la Calle** PUBLICADO  
Editar | Duplicar | Eliminar | Insertar iUnizar | Destacar | Enviar mail | Generar PDF | Auditoría envíos |

☆ **El físico zaragozano Héctor Gómez Maluenda, formado en la Universidad de Zaragoza, clave en el descubrimiento de un pasadizo secreto en la pirámide de Kéops** PUBLICADO  
Editar | Duplicar | Eliminar | Insertar iUnizar | No destacar | Enviar mail | Generar PDF | Auditoría envíos |

03-05-2023 ☆ **La Universidad de Zaragoza ha celebrado el Acto Solemne de Investidura de Nuevos Doctores** PUBLICADO  
Editar | Duplicar | Eliminar | Insertar iUnizar | Destacar | Enviar mail | Generar PDF | Auditoría envíos |

☆ **El físico zaragozano Héctor Gómez Maluenda, formado en la Universidad de Zaragoza, clave en el descubrimiento de un pasadizo secreto en la pirámide de Kéops**  
Editar | Duplicar | Eliminar | Insertar iUnizar | Destacar | Enviar mail | Generar PDF | Auditoría envíos |

Figura 9. Captura aplicación – Listar notas.

## 6. Implementación.

### 6.1. Definiciones, acrónimos y abreviaturas.

**Backend:** Se refiere a la gestión de datos y la comunicación entre componentes de una aplicación.

**Framework:** Conjunto de herramientas que simplifican el desarrollo de aplicaciones.

**Contenedorización:** Tecnología que encapsula una aplicación para proporcionar un entorno consistente y aislado.

**Base de datos:** Sistema para almacenar datos estructurados de forma organizada.

**Frontend:** Parte visible de una aplicación con la que los usuarios interactúan.

**Proxy:** Actúa como intermediario entre cliente y servidor para enrutar solicitudes.

**API REST:** Permite la comunicación y el intercambio de datos eficiente entre sistemas.

**URL:** Dirección para acceder a recursos en línea, como páginas web.

**JSON:** Formato de texto para representar datos estructurados de manera legible para humanos y máquinas.

**Módulo de Drupal:** Extensión que agrega funcionalidades a un sitio web construido con Drupal.

**CSV:** Formato de archivo para representar datos en texto plano.

**Sentencia INSERT:** Instrucción en bases de datos para insertar registros en una tabla.

### 6.2. Proceso de desarrollo y herramientas empleadas.

En este apartado se abordarán las herramientas utilizadas para la implementación de la aplicación. Se hace referencia al Anexo F, donde se proporciona una definición detallada de todas las tecnologías empleadas y su aplicación específica en el desarrollo de esta aplicación.

Para el **backend**, se utilizó Django, un **framework** de desarrollo web basado en Python. El uso de Django permitió crear una aplicación robusta y escalable. Además, la base de datos empleada fue MariaDB, un sistema de gestión de bases de datos relacional, compatible con Django y conocido por su rendimiento y escalabilidad.

La aplicación de Django se implementó en un entorno de **contenedorización** utilizando Docker. Se creó un contenedor de Linux específico para la UZ, donde se instaló y configuró la aplicación de Django junto con la **base de datos** MariaDB. Esta elección permitió una fácil gestión y despliegue de la aplicación en el entorno de la UZ, garantizando la portabilidad y la independencia del sistema operativo.

En cuanto al **frontend**, se emplearon tecnologías web estándar como HTML, CSS y JavaScript. Estas tecnologías son ampliamente utilizadas en el desarrollo de interfaces de usuario y permiten crear páginas web interactivas y atractivas. Se utilizaron HTML para la estructura del contenido, CSS para el diseño visual y JavaScript para agregar interactividad y mejorar la experiencia de usuario. Se aseguró que el diseño y la funcionalidad cumplieran con los requisitos establecidos por el cliente, buscando siempre maximizar la experiencia de usuario.

Además, para mejorar el rendimiento y la escalabilidad de la aplicación de Django, se utilizó Nginx como servidor web y **proxy** inverso. Nginx se configuró para manejar las solicitudes entrantes y distribuir el tráfico de manera eficiente, mejorando así la capacidad de respuesta y la estabilidad de la aplicación.

### 6.3. API Rest y migración de datos de la antigua aplicación.

Para la integración de la aplicación Django con la **API REST**, se ha implementado un flujo de exportación e importación de notas de prensa entre ambas aplicaciones.

En la aplicación Django, se ha desarrollado un módulo dedicado a la exportación de notas de prensa. Los usuarios pueden seleccionar las notas que desean exportar, y la aplicación genera un archivo que contiene todos los datos relevantes de estas notas. Este archivo se muestra en una **URL** específica, donde se encuentran los datos en formato **JSON**.

Por otro lado, en la aplicación Drupal, que aloja la página de la UZ, se ha instalado el **módulo** feed\_ex. A través de la configuración de este módulo, se ha indicado la URL donde se encuentran las notas a importar desde la aplicación Django. Además, se ha establecido un mecanismo de revisión automática cada hora, para que el módulo feed\_ex verifique si existen nuevas notas en el enlace proporcionado. Si se encuentran notas nuevas, el módulo las importa al sistema.

Para evitar la importación duplicada de notas, se ha utilizado un identificador único para cada nota. Esto garantiza que solo las notas que no hayan sido importadas previamente sean incluidas en el sistema.

Es importante destacar que, anteriormente, la importación de notas se realizaba a través de archivos **CSV**. Sin embargo, este enfoque limitaba la capacidad de adjuntar más de una imagen a cada nota importada. Con la implementación actual, utilizando la API REST, se ha superado esta limitación y se permite una mayor flexibilidad en la importación de noticias.

Una vez importadas las notas, estas aparecen en la página web de la UZ, permitiendo a los usuarios acceder a la información actualizada de las notas de prensa.

En el proceso de migración de los datos de la antigua aplicación, se ha llevado a cabo una serie de pasos para exportar y luego importar todas las notas de prensa desde la base de datos anterior a la nueva base de datos.

Para realizar esta migración, se ha accedido a la base de datos de la aplicación antigua y se han exportado todos los datos relevantes de las notas de prensa. Esta exportación ha generado un archivo que contiene la información necesaria para realizar la transferencia.

Posteriormente, se ha desarrollado una función simple en Python para leer el archivo exportado, donde se encuentran los datos de la antigua base de datos. Mediante el proceso de parsing, se han extraído los diferentes campos y se han procesado para que sean compatibles con el formato de la nueva base de datos.

Utilizando una **sentencia INSERT**, los datos parseados de las notas de prensa se han cargado en la nueva base de datos, que también es una base de datos MariaDB, asegurando así una transición fluida de los datos.

Este proceso de migración ha permitido transferir correctamente las notas de prensa desde la antigua base de datos a la nueva base de datos, asegurando la continuidad de la información y su disponibilidad en la aplicación actualizada.

## 7. Conclusiones y trabajos futuros.

### 7.1. Conclusiones.

Elegir una metodología y un ciclo de vida para el desarrollo de la aplicación, que se ajusten a las necesidades del cliente, antes de comenzar el proyecto. Esto facilita la planificación de las fechas de entrega.

En cuanto al análisis y diseño de la aplicación, UML es un modelo muy útil ya que permite tener una perspectiva completa del proyecto, asegurando la coherencia y la integridad del sistema.

El uso de software libre resulta muy beneficioso en el desarrollo de un proyecto para una universidad debido a: características como el acceso al código fuente, que permite realizar modificaciones que se adapten a las necesidades específicas de la institución; y a la colaboración activa de la comunidad de desarrolladores, que constantemente buscan soluciones, parches de seguridad y mejoras.

La utilización de un framework resulta interesante, ya que facilita la construcción rápida y eficiente de la aplicación, con actualizaciones que implican una mejora continua, sin perder la posibilidad de crear módulos adicionales que se ajusten a las necesidades del cliente.

Realizar un trabajo de fin de grado en colaboración con una institución brinda la oportunidad de aplicar los conocimientos adquiridos durante los estudios a situaciones reales. Además, amplía significativamente la experiencia del estudiante.

### 7.2. Trabajos futuros.

Esta aplicación ha sido concebida como el inicio de un extenso proyecto en el ámbito universitario, donde todas las aplicaciones utilizadas en la institución siguen una misma métrica. El hecho de que todas las aplicaciones estén construidas de manera consistente garantiza su compatibilidad mutua y facilita el mantenimiento, ya que el enfoque utilizado es uniforme en todas ellas. El siguiente paso en este proyecto es la implementación de la aplicación iUnizar, que ya tiene conexiones con la aplicación desarrollada en este TFG. A través de la gestión de notas de prensa en esta aplicación, se alimentará la base de datos de iUnizar. En esta nueva aplicación se generarán boletines informativos diarios con una

presentación agradable y personalizada para los usuarios, seguido de su distribución a todos los miembros de la universidad.

## 8. Bibliografía.

- [1] IBM. APIs de REST. <https://www.ibm.com/es-es/topics/rest-apis>
- [2] Budde, Reinhard, et al. "What is prototyping?." *Information Technology & People* 6.2/3 (1990): 89-95
- [3] Metodología Scrum. <https://www.scrum.org>
- [4] Manifiesto Ágil para el desarrollo de Software <https://agilemanifesto.org>
- [5] MailChimp. <https://mailchimp.com/>
- [6] Constant Contact. <https://www.constantcontact.com/>
- [7] HubSpot. <https://www.hubspot.es/>
- [8] SendinBlue. <https://academy.sendinblue.com/>
- [9] PHP. <https://www.php.net/manual/es/intro-whatism.php>
- [10] HTML. <https://developer.mozilla.org/es/docs/Web/HTML>
- [11] CSS. <https://developer.mozilla.org/es/docs/Web/CSS>
- [12] Javascript. <https://developer.mozilla.org/es/docs/Web/JavaScript>
- [13] Django. <https://www.djangoproject.com/>
- [14] MariaDB. <https://mariadb.org/>
- [15] Drupal. <https://www.drupal.org/>
- [16] JSON. <https://www.ibm.com/docs/es/baw/20.x?topic=formats-javascript-object-notation-json-format>
- [17] Docker. <https://www.docker.com/>
- [18] HTTPS. <https://www.siteground.es/blog/que-es-https-y-para-que-sirve-guia-completa/>
- [19] Casos de uso. Jacobson, I., Spence, I., Bittner, K. (2011). USE-CASE 2.0 The Guide to Succeeding with Use Cases. Ivar Jacobson International SA.
- [20] UML. OMG (2011). OMG Unified Modeling Language (OMG UML), Superstructure, Version 2.4.1 Object Management Group (Technical report, Object Management Group)
- [21] Nginx. <https://www.nginx.com>

## Anexo A. Análisis de las clases que forman la aplicación.

A continuación, se va a realizar un análisis a profundidad de cada una de las clases de la aplicación.

### **Clase: Usuario**

Atributos:

- **NIP:** Es un número entero de 6 dígitos que actúa como identificador único del usuario dentro de la UZ. Se utiliza para iniciar sesión en todos los entornos relacionados con la UZ.
- **Contraseña:** Es una cadena que representa la contraseña administrativa del usuario en la UZ. Esta contraseña se utiliza para acceder a los servicios y recursos de la UZ.
- **Correo electrónico:** Es una cadena que representa la dirección de correo electrónico del usuario en la UZ. Se utiliza como medio de contacto para comunicarse con el usuario cuando sea necesario.

Además de estos atributos, la clase "Usuario" también tiene un rol dentro de la aplicación. Aunque todos los miembros de la UZ pueden autenticarse en la aplicación, los permisos y acciones que pueden realizar están determinados por el rol asignado.

Inicialmente, cuando un usuario se autentica por primera vez, no tiene ningún rol asignado y no puede realizar ninguna acción. Para obtener un rol específico, el usuario debe ponerse en contacto con un administrador y solicitarlo.

Existen tres roles posibles que un administrador puede asignar al usuario:

- **Becario:** Un usuario con el rol de becario puede realizar todas las acciones relacionadas con la gestión de notas de prensa, convocatorias para los medios y medios de prensa.
- **Superusuario:** Además de las acciones relacionadas con la gestión de contenidos mencionados anteriormente, un superusuario tiene la capacidad de visualizar los registros de actividad (logs) de todos los usuarios de la aplicación.
- **Administrador:** Un administrador tiene todos los permisos y acciones de los roles anteriores. Además, puede gestionar los roles de los demás usuarios, lo que implica asignar, modificar o revocar roles en la aplicación.

## Clase: Autor

Atributos:

- Nombre: Es una cadena que representa el nombre del departamento responsable de la redacción de la nota de prensa o convocatoria. Proporciona información sobre la entidad o área específica que emite el comunicado.
- Teléfono: Es un número entero que almacena el valor del teléfono de contacto del departamento. Este atributo permite proporcionar un medio de comunicación adicional para aquellos que deseen ponerse en contacto directamente con el departamento responsable.
- Correo electrónico: Es una cadena que almacena la dirección de correo electrónico de contacto del departamento. Proporciona un canal de comunicación importante para aquellos que necesiten enviar consultas, comentarios o solicitar información adicional.

Además de estos atributos, la clase "Autor" cuenta con un método privado llamado "firmar". Este método genera una firma a partir de sus atributos privados que se inserta al final de cada nota de prensa o convocatoria creada desde ese departamento.

## Clase: Nota

Atributos:

- ID: Es un número entero que identifica de manera única cada instancia de la clase "Nota". Cada nota creada tendrá un ID diferente que la distingue de las demás.
- Fecha: Es una cadena que indica la fecha para la cual está programada la nota de prensa, es decir, la fecha en la que se espera que sea enviada y publicada en la página web de la UZ.
- Título: Es una cadena que almacena el título de la nota de prensa. Proporciona un resumen conciso y llamativo del contenido de la nota.
- Subtítulo: Es una cadena que contiene el subtítulo de la nota de prensa. Puede ser utilizado para agregar información adicional o destacar un aspecto específico de la nota.
- Cuerpo: Es una cadena que almacena el contenido completo de la nota de prensa. Aquí se incluye toda la información relevante y detallada sobre el tema tratado.
- Contacto: Es una cadena que proporciona información de contacto para aquellos lectores que deseen obtener más información sobre el tema de la nota de prensa. Puede incluir un nombre, número de teléfono o dirección de correo electrónico.
- Destacado: Es un valor booleano que determina si la nota de prensa debe aparecer entre las noticias destacadas en la página web de la UZ. Si es "true", la nota se mostrará como destacada; si es "false", no se destacará.
- Publicado: Es un valor booleano que indica si la nota de prensa ha sido publicada en la página web de la UZ. Si es "true", la nota se ha publicado; si es "false", aún no se ha publicado.

La clase "Nota" también cuenta con los siguientes métodos:

- **editar\_nota:** Este método permite a un usuario modificar los valores de la nota de prensa. Al invocar este método, se pueden actualizar el título, subtítulo, cuerpo, contacto y otros atributos de la nota.
- **duplicar\_nota:** Cuando este método es invocado por un usuario, se crea una copia exacta de la nota con un ID diferente. Esto permite generar duplicados de una nota existente sin tener que volver a ingresar todos los detalles manualmente.
- **eliminar\_nota:** Un usuario puede invocar este método para eliminar la nota de prensa. Al llamar a este método, la instancia de la clase "Nota" correspondiente se elimina del sistema.
- **destacar\_nota:** Este método permite a un usuario marcar el atributo "destacado" de una instancia de la clase "Nota" como "true". Esto asegura que la nota se muestre entre las noticias destacadas en la página web de la UZ.
- **no\_destacar\_nota:** Este método es el contrario del anterior. Permite a un usuario cambiar el atributo "destacado" de una instancia de la clase "Nota" a "false", eliminando la nota de las noticias destacadas.
- **enviar\_nota:** Al invocar este método, un usuario puede enviar la nota de prensa a los medios de comunicación que desee. Este método facilita el proceso de distribución de la nota a través de los canales de comunicación apropiados.
- **insertar\_iunizar:** Cuando un usuario invoca este método, la nota de prensa se inserta en la base de datos de la aplicación del boletín informativo de la UZ. Esto implica almacenar la información de la nota en la base de datos para su posterior uso y referencia.
- **generar\_pdf:** Cuando un usuario invoca este método, se devuelve un archivo PDF que representa la instancia de la clase "Nota" a la cual se le ha invocado el método. Este archivo PDF contiene la información completa de la nota, incluyendo el título, subtítulo, cuerpo, contacto, y los autores de esta.

### **Clase: Imagen**

Atributos:

- **ID:** Es un número entero que actúa como identificador único de la imagen. Cada instancia de la clase "Imagen" tiene un ID diferente, lo que permite distinguir una imagen de otra.
- **Nombre:** Es una cadena que representa el título o descripción de la imagen. Este atributo proporciona contexto o información adicional sobre el contenido visual que se está mostrando.
- **Autor:** Es una cadena que indica el autor de la imagen. Proporciona información sobre la persona o entidad responsable de la creación de la imagen.

Cuando un usuario sube una imagen como parte de una nota o convocatoria, se crea una instancia de la clase "Imagen" con un ID único y se asignan los valores correspondientes a los atributos "Nombre" y "Autor".

## **Clase: Medio**

Atributos:

- **Nombre:** Es una cadena que almacena el nombre del medio de prensa. Proporciona una identificación única para cada medio, como el nombre de un periódico, una estación de televisión o una revista.
- **Correo:** Es una cadena que almacena la dirección de correo electrónico a la cual se debe enviar la nota de prensa para ese medio en particular. Este atributo especifica el destinatario principal dentro del medio de prensa.
- **Contacto:** Es una cadena que contiene el nombre de la persona asociada a ese medio. Proporciona el nombre del contacto o responsable de recibir y gestionar la información relacionada con las notas de prensa.
- **Teléfono:** Es un número entero que almacena el número de teléfono del medio de prensa. Proporciona un medio adicional de contacto para comunicarse con el medio en caso de ser necesario.

La clase "Medio" también cuenta con los siguientes métodos:

- **eliminar\_medio:** Este método permite eliminar un medio de prensa. Al invocar este método, se elimina la instancia de la clase "Medio" correspondiente, eliminando así la información asociada a ese medio de prensa en particular.
- **editar\_medio:** Al invocar este método, se pueden actualizar el nombre, correo electrónico, contacto y número de teléfono asociados a ese medio.

## **Clase: Categoría**

Atributos:

- **Nombre:** Es una cadena que almacena el nombre de la categoría. Este atributo proporciona una identificación descriptiva de la temática o área a la que pertenecen los medios agrupados en esa categoría.

La clase "Categoría" se utiliza para organizar y clasificar los medios de prensa en función de su contenido o enfoque. Por ejemplo, se pueden crear categorías como "General", "Investigación", "Cultura", "Huesca", etc. Cada categoría contendría una lista de medios relacionados con esa temática específica.

La clase "Categoría" se utiliza para facilitar la organización y gestión de los medios de prensa en la aplicación. Permite enviar notas de prensa a medios aislados o a categorías completas, lo que simplifica el proceso de distribución de información a través de diferentes temáticas.

## **Clase: Envio**

Atributos:

- Fecha: Es un atributo que almacena la fecha y hora en la que se realizó el envío de la nota de prensa. Este atributo registra el momento preciso en el que se ejecutó el envío.

El atributo "Fecha" de la clase "Envio" captura el instante exacto en el que se realizó el envío de la nota. Esto puede ser útil para llevar un registro de los envíos realizados y rastrear la cronología de las acciones realizadas.

Cada instancia de la clase "Envio" se crea individualmente para cada envío de nota de prensa. Esto permite tener un objeto específico que representa ese envío particular, con su propia fecha asociada.

## **Clase: Exportacion**

Atributos:

- Fecha: Es un atributo que almacena la fecha y hora en la que se realizó la exportación de las notas a la web de la UZ. Este atributo registra el momento preciso en el que se ejecutó la exportación.

El atributo "Fecha" de la clase "Exportacion" captura el instante exacto en el que se llevó a cabo la exportación de las notas. Esto permite tener un registro de las exportaciones realizadas.

Cada instancia de la clase "Exportacion" se crea individualmente para cada exportación de notas a la web de la UZ.

## **Clase: Log**

Atributos:

- Fecha: Es una cadena que almacena el momento exacto en el que se realizó la acción. Este atributo registra la fecha y hora en la que ocurrió la acción.
- Acción: Es una cadena descriptiva que indica la acción realizada por el usuario. Proporciona información sobre la acción específica que se llevó a cabo.

La clase "Log" se utiliza para mantener un registro de las acciones realizadas por los usuarios con el fin de tener un historial de actividad y facilitar la supervisión, el seguimiento y la resolución de problemas.

## Anexo B. Diagramas de secuencia relevantes en la aplicación.

### Diagrama de creación de cuenta y asignación de roles.

En la Figura 10, se observa que el proceso comienza con una autenticación exitosa en la aplicación. Una vez que un usuario de la UZ proporciona sus credenciales en la pantalla de inicio de sesión, la aplicación verifica la corrección de estos datos en la base de datos de usuarios.

Si las credenciales son correctas, se crea una instancia de la clase Cuenta utilizando el nombre de usuario y la contraseña administrativa de la UZ. Con el fin de restringir el acceso a la aplicación, la cuenta se crea inicialmente sin asignar un rol por defecto.

Para solicitar la asignación de un rol, el usuario debe hacerlo a través de un administrador. El administrador evaluará si debe asignarle un rol y, de ser así, determinará cuál será el rol asignado.

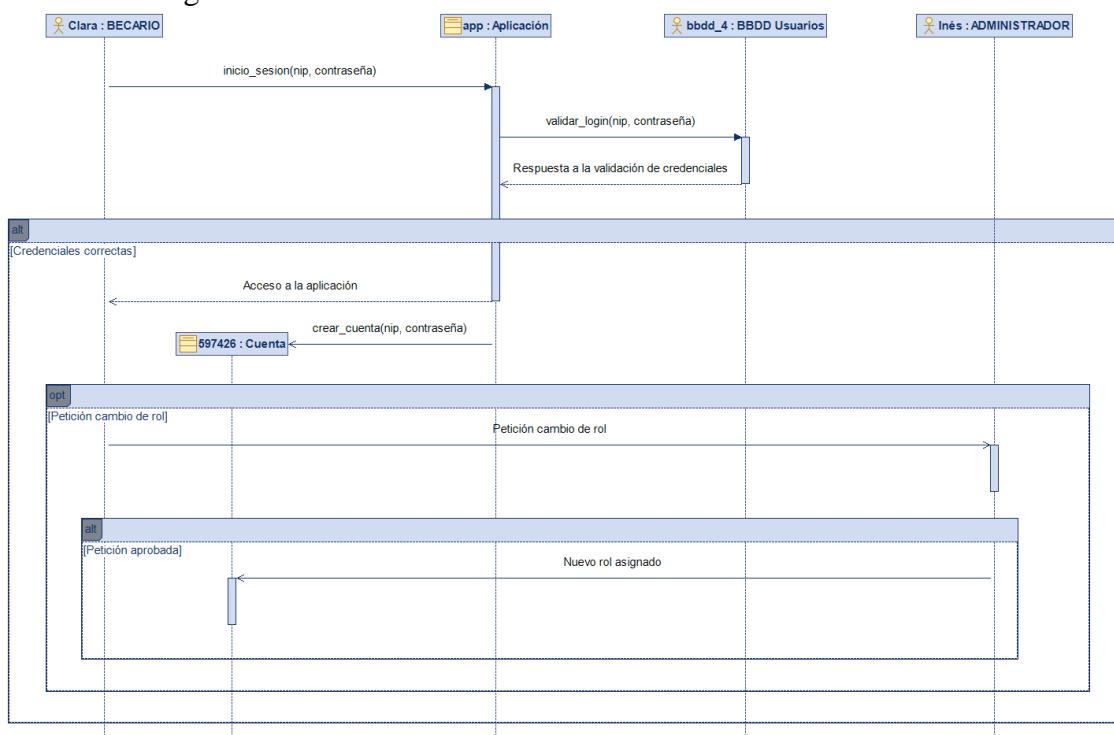


Figura 10. Diagrama de secuencia – Creación de cuenta y asignación de roles.

### Diagrama de creación de nota

En la Figura 11 se muestra el proceso de creación de una Nota de prensa. Cuando una cuenta con un rol asignado hace clic en el botón "Nueva nota", se le presentará un formulario de creación de nota. Una vez que dicho formulario haya sido completado, la aplicación guardará la nota en la base de datos y creará una nueva instancia de la clase Nota.

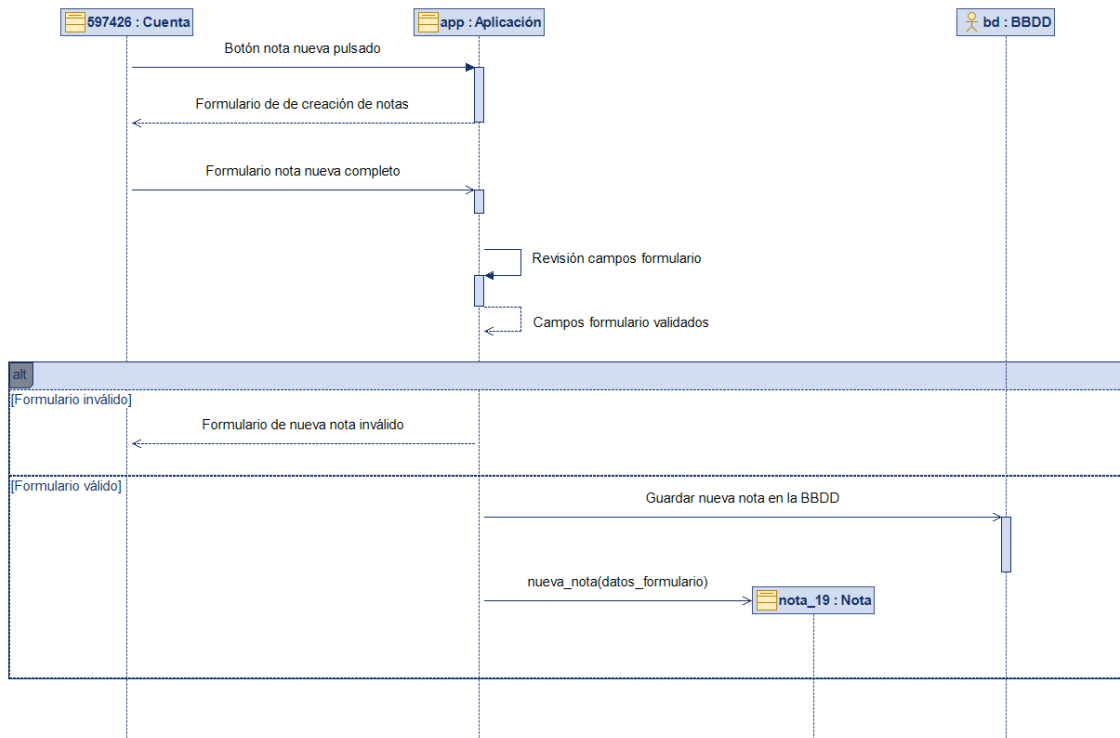


Figura 11. Diagrama de secuencia – Creación de una nota.

### Diagrama de envío de notas

En la Figura 12, se muestra el proceso de envío de una nota. Cuando una cuenta con cualquier rol asignado selecciona el botón "Enviar mail" asociado a una nota, la aplicación le proporciona un formulario de envío. En dicho formulario, el usuario puede visualizar una previsualización de la nota que se enviará.

En este formulario, se deben especificar los destinatarios del envío. Si el usuario completa correctamente el formulario, se llevará a cabo el envío y se creará una nueva instancia de la clase Envío que almacenará la información relevante a dicho envío.

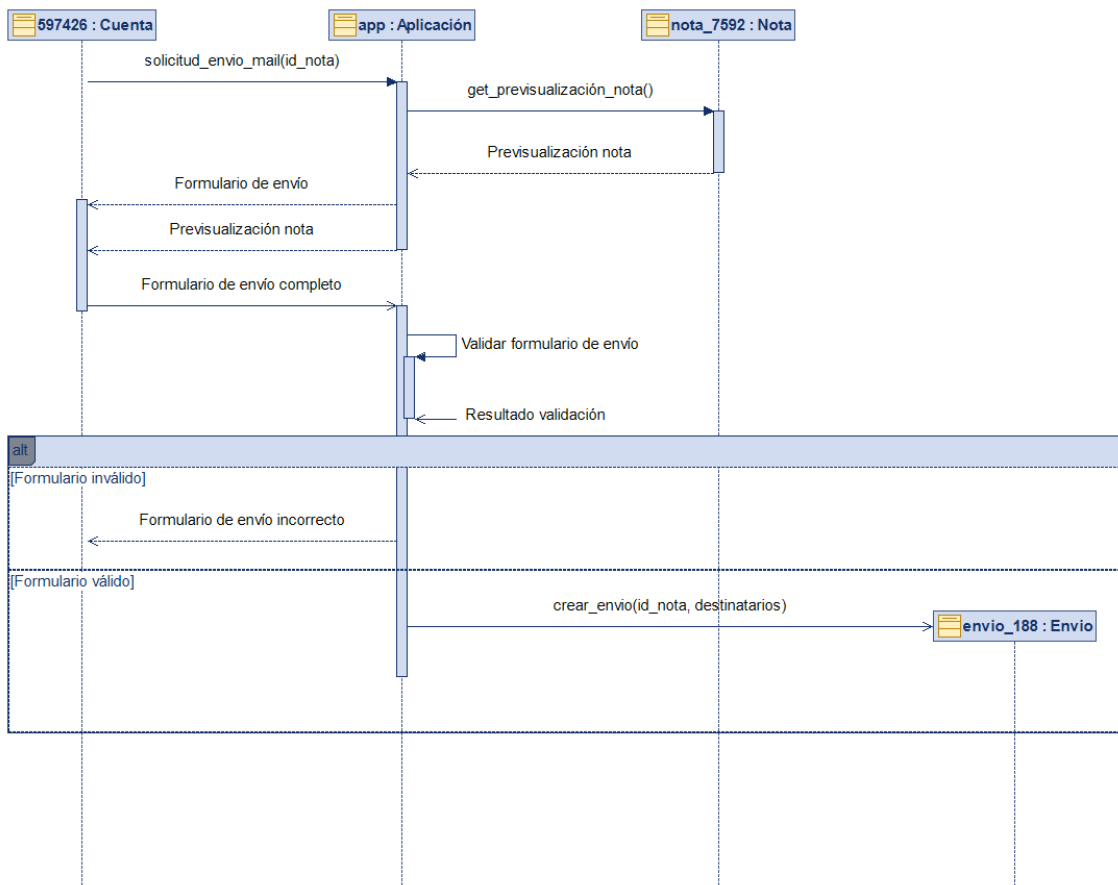


Figura 12. Diagrama de secuencia – Envío de una nota.

## Anexo C. Tecnologías empleadas en la aplicación.

### Docker

Es una plataforma de contenedores que permite empaquetar y desplegar aplicaciones de forma independiente del sistema operativo subyacente. La aplicación de Django se ha desplegado en Docker, lo que proporciona un entorno de ejecución consistente y facilita la portabilidad.



Figura 13. Logo Docker.

### Nginx

Es un servidor web de código abierto que actúa como intermediario entre el contenedor de Docker y la aplicación Django. Nginx se utiliza para gestionar las solicitudes HTTP.



*Figura 14. Logo Nginx.*

### **Django**

Es un framework de desarrollo web de código abierto, escrito en Python. Django proporciona una estructura sólida y eficiente para desarrollar aplicaciones web, facilitando tareas como la gestión de bases de datos y la autenticación de usuarios. La aplicación se ha programado utilizando el framework Django.



*Figura 15. Logo Django.*

### **Python**

Es un lenguaje de programación de código abierto. Django se basa en Python y se utiliza como el lenguaje principal de programación en la aplicación. Python proporciona una sintaxis sencilla que facilita el desarrollo de aplicaciones web.



*Figura 16. Logo Python.*

### **MariaDB**

Es un sistema de gestión de bases de datos relacional. Ha sido elegido como la base de datos para la aplicación, brindando una estructura y almacenamiento eficiente para los datos relacionados con la aplicación.



*Figura 17. Logo MariaDB.*

### **JSON**

Es un formato de intercambio de datos. Se ha seleccionado como el formato para la comunicación entre la API y la página institucional de la UZ, permitiendo una transferencia de datos eficiente y comprensible.



*Figura 18. Logo JSON.*

### **Drupal**

Es un framework de gestión de contenidos de código abierto utilizado en las páginas web de la UZ. En estas páginas, se han implementado módulos que se alimentan de las notas exportadas por la aplicación a través de la API, permitiendo la actualización de contenido de manera automatizada.



*Figura 19. Logo Drupal.*

### **HTML**

Es el lenguaje estándar utilizado para la presentación de páginas web. Cada página de la aplicación cuenta con una plantilla HTML personalizada para facilitar la navegación del usuario.



*Figura 20. Logo HTML.*

## CSS

Es un lenguaje de hojas de estilo utilizado para definir la apariencia de las páginas web. En la aplicación, se ha utilizado CSS para mejorar la experiencia del usuario y lograr un diseño moderno y sofisticado.



*Figura 21. Logo CSS.*

## JavaScript

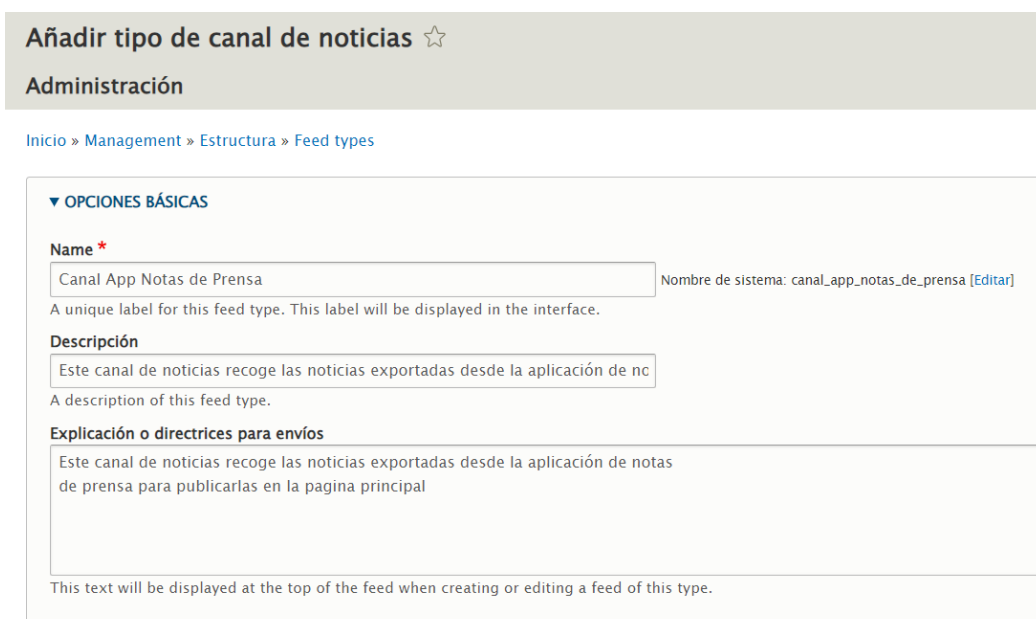
Es un lenguaje de programación utilizado en el desarrollo web para agregar interactividad y funcionalidades dinámicas a las páginas. En la aplicación, JavaScript se ha empleado para implementar funcionalidades adicionales que no están contempladas en Django, como la edición de atributos de imágenes dentro de un formulario de creación de notas.



*Figura 22. Logo JavaScript.*

## Anexo D. Configuración de la API REST de la aplicación Notas de Prensa y Drupal.

1. Para implementar la alimentación automatizada de contenido con Drupal, se deben seguir los siguientes pasos:
2. Instalar los módulos necesarios: Es necesario instalar los siguientes módulos en el sitio de Drupal:
  - Feeds: <https://www.drupal.org/project/feeds>
  - Feeds Extensible Parsers: [https://www.drupal.org/project/feeds\\_ex](https://www.drupal.org/project/feeds_ex)
  - Feeds Tamper: [https://www.drupal.org/project/feeds\\_tamper](https://www.drupal.org/project/feeds_tamper)
3. Acceder a la configuración del sitio: Una vez instalados los módulos, dirigirse al menú de configuración del sitio.
4. Crear un nuevo tipo de feed: En la sección "Feed Types", agregar un nuevo tipo de feed con un nombre y descripción que faciliten su identificación.



**Añadir tipo de canal de noticias** ☆

Administración

[Inicio](#) » [Management](#) » [Estructura](#) » [Feed types](#)

▼ OPCIONES BÁSICAS

**Name \***

Nombre de sistema: canal\_app\_notas\_de\_prensa [Editar]

A unique label for this feed type. This label will be displayed in the interface.

**Descripción**

A description of this feed type.

**Explicación o directrices para envíos**

This text will be displayed at the top of the feed when creating or editing a feed of this type.

*Figura 23. Creación Feed Type – Opciones básicas.*

5. Configurar el tipo de feed: Especificar que la obtención de información se realizará a través de una URL y que el formato utilizado es JsonPath. Indicar que la información importada se usará para crear nodos de tipo "Noticia". Establecer la frecuencia de importación, por ejemplo, cada hora.

Fetcher: 
 Parser: 
 Processor:

Tipo de contenido \*:

**Configuración**

Import period: 
  
Choose how often a feed should be imported. [Requiere que se configure cron.](#)

Figura 24. Creación Feed Type – Configuración general.

6. Configurar el procesador: En la configuración del procesador, se debe indicar que se importe el contenido nuevo y evitar la duplicación de contenido repetido. Esto se logra utilizando identificadores (IDs) para cada nota. Seleccionar la opción de conservar los elementos importados previamente para que no se eliminen al importar nuevo contenido.

**Configuración**

Idioma \*:

**Insert new elementos de contenido**
  
 Insert new elementos de contenido
   
 Do not insert new elementos de contenido
   
New elementos de contenido will be determined using mappings that are a "unique target".

**Actualizar elementos de contenido existentes**
  
 No actualizar las elementos de contenido existentes
   
 Replace existing elementos de contenido
   
 Actualizar elementos de contenido existentes
   
Existing elementos de contenido will be determined using mappings that are unique.

**Previously imported items**
  

  
Select what to do with items that were previously imported, but are now no longer in the feed.

**Expire elementos de contenido**
  

  
Select after how much time elementos de contenido should be deleted.
  
 Owner: Feed author
   
Use the feed author as the owner of the entities to be created.

**Propietario**
  

  
Select the owner of the entities to be created. Leave blank for *Anónimo*.

Figura 25. Creación Feed Type – Configuración del Procesador.

7. Guardar y asignar correspondencias: Una vez realizada la configuración, guardar los cambios y asignar las correspondencias de los campos. En el apartado "Origen", especificar la ruta que apunta a la información necesaria para cada campo. En el apartado "Destino", indicar el campo de creación de una nueva noticia en Drupal. Mapear todos los campos requeridos y guardar los cambios.

[Inicio](#) » [Administración](#) » [Estructura](#) » [Feed types](#) » [Editar JSON](#)

Define which elements of a single item of a feed (= *Fuentes*) map to which content pieces in Drupal (= *Targets*). To avoid importing duplicate content, the URL `http://example.com/content/1` can exist.

On *Read only* targets a value can only be set the first time. Blank sources can be used for tampering: see the [documentation](#).

See the [Mapping documentation](#) for more information.

Contexto \*

\$.\*

The base query to run. See the [Context query documentation](#) for more information.

ORIGEN	DESTINO
tit_notas	Título (title)
- Select a source -	Texto de la noticia (body): Resumen
cuer_notas	Texto de la noticia (body): Texto
subt_notas	Subtítulo (field_subtitulo_noticia)
fecha_prog	Fecha de la noticia (field_fecha)
nodo_imagenes[*].url_img (url_img)	Galería de imágenes (field_foto_galeria): ID Archivo
nodo_imagenes[*].aut_img (aut_img)	Galería de imágenes (field_foto_galeria): Texto alternativo
nodo_imagenes[*].nom_img (nom_img)	Galería de imágenes (field_foto_galeria): Título

Figura 26. Creación Feed Type – Correspondencias.

```
{
  "id_notas": 136,
  "fecha_prog": "2023-06-07",
  "img_port": {
    "url_img": "https://ip-sitio-web/AplicacionNP/0_01_cascaras_pachykrok - copia - copia - copia_ge2dS2b.jpg",
    "nom_img": "0_01_cascaras_pachykrok - copia - copia - copia.jpg",
    "aut_img": "Universidad de Zaragoza."
  },
  "nodo_imagenes": [
    {
      "url_img": "https://ip-sitio-web/AplicacionNP/0_01_cascaras_pachykrok - copia - copia - copia_ge2dS2b.jpg",
      "nom_img": "0_01_cascaras_pachykrok - copia - copia - copia.jpg",
      "aut_img": "Universidad de Zaragoza."
    },
    {
      "url_img": "https://ip-sitio-web/AplicacionNP/0_encuentro_inma_2 - copia - copia_k2XF1qP.jpg",
      "nom_img": "0_encuentro_inma_2 - copia - copia.jpg",
      "aut_img": "Universidad de Zaragoza."
    },
    {
      "url_img": "https://ip-sitio-web/AplicacionNP/0_foto_1_ponentes_en_el_pasillo - copia_Ayk2Evs.jpg",
      "nom_img": "0_foto_1_ponentes_en_el_pasillo - copia.jpg",
      "aut_img": "Universidad de Zaragoza."
    }
  ],
  "dest": false,
  "tit_notas": "Título 1",
  "subt_notas": "Subtítulo 1",
  "cuer_notas": "<p>Cuerpo 1</p>"
}
```

Figura 27. Estructura JSON generado por la app “Notas de prensa”.

8. Crear un canal de noticias: En el menú de configuración, acceder a la sección de contenido y seleccionar "Agregar canal de noticias" en la pestaña correspondiente. Seleccionar el feed creado anteriormente y proporcionar la URL desde la cual se alimentará el canal.

[Inicio](#) » [Añadir canal de noticias](#)

**Título \***

The title of this feed, always treated as non-markup plain text.

**URL del canal de noticias \***

<b>Import options</b>	<input checked="" type="checkbox"/> Activo Uncheck the above checkbox to disable periodic import for this feed.
<a href="#">Información de autoría</a>	

**Save and import** ▾

Figura 28. Creación canal de noticias.

9. Una vez completados estos pasos, el módulo de Drupal estará configurado y funcionando correctamente en Drupal.

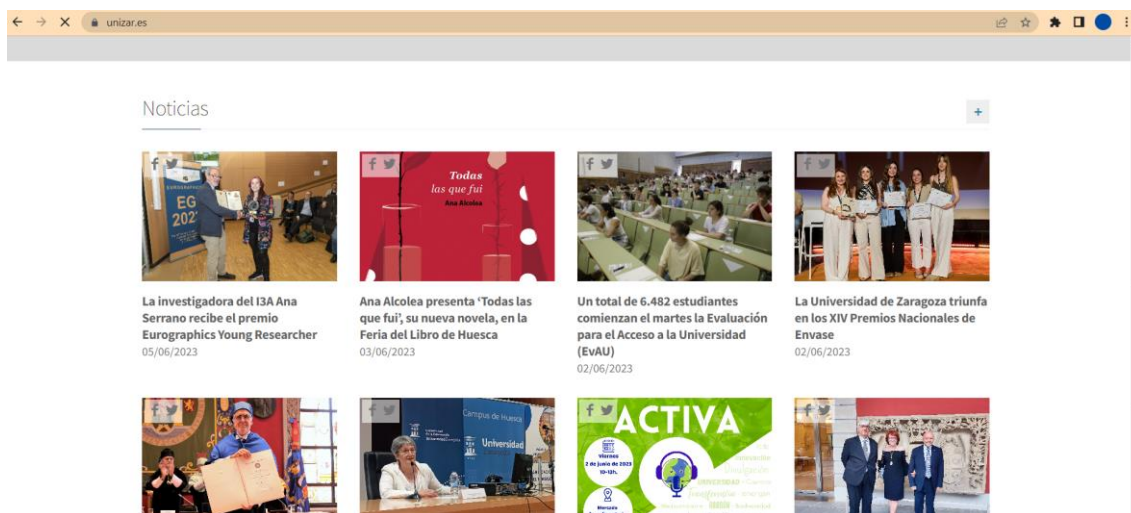


Figura 29. Vista de las noticias importadas en la web de [www.unizar.es](http://www.unizar.es)

## Anexo E. Capturas de la aplicación y cumplimiento de los requisitos funcionales.

En la Figura 30, se presenta la pantalla de inicio de sesión, que es la primera vista que un usuario encuentra al acceder a la aplicación. Si el usuario no tiene un rol asignado, no podrá acceder a ninguna otra pantalla.

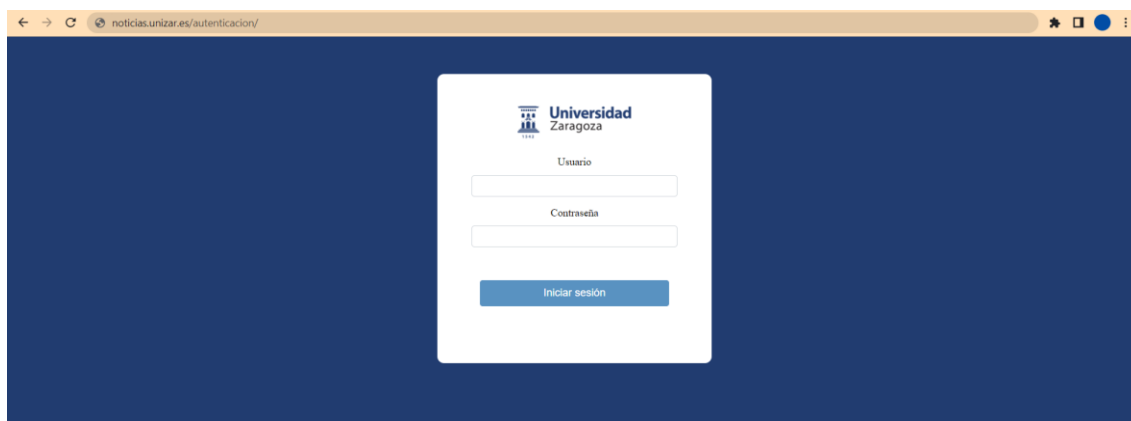


Figura 30. Captura app “Notas de prensa” – Pantalla inicio sesión.

En la Figura 31, se muestra la pantalla de creación de una nueva nota de prensa en la que se cumplen los requisitos RF1 (Crear Nota de Prensa), RF2 (Añadir autores a la Nota de Prensa y firmarla) y RF3 (Editor de texto para campos específicos de la Nota de Prensa).

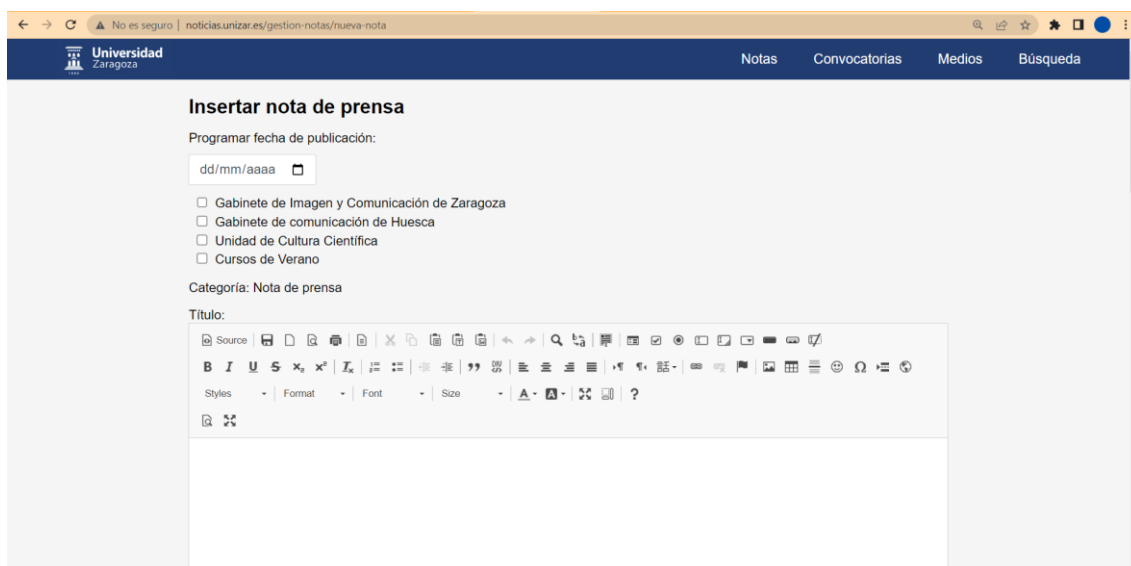
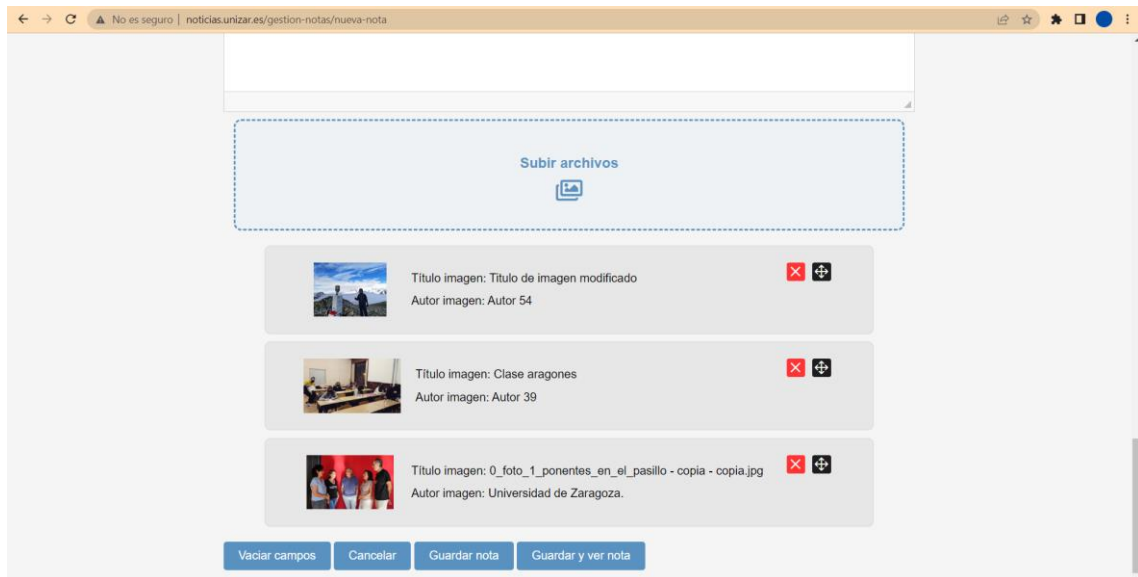


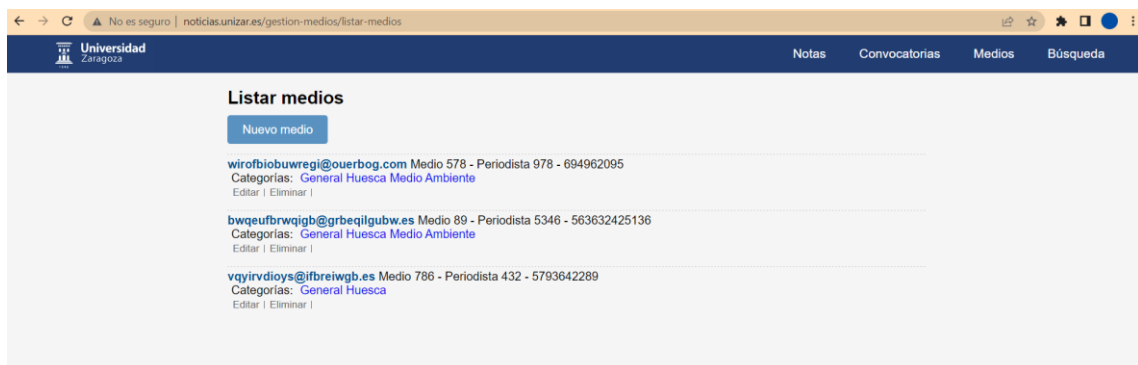
Figura 31. Captura app “Notas de prensa”- Pantalla creación nueva nota.

En la Figura 32, se presenta el formulario de creación de una nota de prensa, donde se puede gestionar el contenido multimedia, cumpliendo así con el requisito RF4 (Gestión de archivos multimedia en la Nota de Prensa).



*Figura 32. Captura app “Notas de prensa” – Pantalla creación nueva nota (Gestión multimedia).*

En la Figura 33, se muestra la lista de medios de prensa en pantalla, cumpliendo los requisitos RF15 (Crear Medios de Prensa) y RF16 (Listar Medios de Prensa).



*Figura 33. Captura app “Notas de prensa”- Pantalla listado medios de prensa.*

En la Figura 34, se puede observar la pantalla de búsqueda, donde se ofrece la posibilidad de buscar por fecha, categoría, autor y palabras clave.

The image shows a web browser window with the URL `noticias.unizar.es/busqueda`. The page header includes the University of Zaragoza logo and navigation links for 'Notas', 'Convocatorias', 'Medios', and 'Búsqueda'. The main content area is titled 'Buscar en la base de datos:' and contains the following search filters:

- Fecha inicial:** 30/05/2023
- Fecha final:** 22/06/2023
- Noticia de prensa:**  Nota de prensa,  Convocatoria para los medios
- Autor:**  Gabinete de Imagen y Comunicación de Zaragoza,  Gabinete de comunicación de Huesca,  Unidad de Cultura Científica,  Cursos de Verano
- Palabras contenidas en la entrada:** A text input field with an 'Añadir' button.
- Tags:** Three tags are displayed: 'caer x', 'congreso x', and 'cultura x'.
- Enviar:** A blue button to submit the search.

*Figura 34. Captura app “Notas de prensa” – Pantalla búsqueda.*