



UNIVERSIDAD DE VALLADOLID

**ESCUELA DE INGENIERÍA INFORMÁTICA DE
SEGOVIA**

**Grado en Ingeniería Informática
de Servicios y Aplicaciones**

FACTOR

Plataforma web para la publicación y validación investigativa y de interés general

Alumno: Mario Olmos Manso

Tutor: Fernando Díaz Gómez

Agradecimientos

Quiero expresar mi más sincero agradecimiento a cuantas personas me han ayudado, de una u otra forma, a la realización de este trabajo.

De forma especial a mi tutor D.Fernando Díaz Gómez por su dedicación, accesibilidad y cercanía. Y, por supuesto, a mi familia y amigos, por su compañía, palabras de ánimo y su ayuda incondicional a lo largo de todo el proyecto.

Resumen

La finalidad principal de este proyecto es la de proporcionar una plataforma web en la que poder publicar, buscar y votar diferentes artículos divulgativos. Uno de los fundamentos que persigue es la creación de un ecosistema que sea mantenido por la propia comunidad de la plataforma, priorizando como pilares la calidad y la veracidad de la información. Todo ello con un toque un poco más informal, pensado para su uso más como tipo red social que como un repositorio de artículos científicos.

Palabras clave: Aplicación web, artículos, autores, veracidad

Abstract

The main objective of this project is to provide a web platform where users can publish, search for, and vote on various informative articles. One of its core principles is to foster an ecosystem maintained by the community itself, placing quality and accuracy of information as top priorities. The approach is somewhat informal, leaning more toward a social network style than a repository of scientific articles.

Keywords: web application, articles, authors, veracity

Contenido

Índice de figuras.....	8
Índice de tablas	10
Capítulo 1: Introducción.....	12
1.1 Organización del documento.....	12
1.2 Introducción	13
1.3 Motivación	13
1.4 Objetivos.....	14
1.5 Alcance	17
1.6 Entorno y tecnologías utilizadas	20
1.6.1 <i>Backend</i>	20
1.6.2 <i>Frontend</i>	22
1.6.3 Herramientas adicionales.....	23
Capítulo 2: Estado del arte.....	26
2.1 Solución propuesta	26
2.2 Entorno de la aplicación	26
2.2.1 <i>Google Scholar</i>	27
2.2.2 <i>Reddit</i>	27
2.2.3 Comparativa	28
Capítulo 3: Planificación y presupuestos	29
3.1 Metodología del trabajo.....	29
3.2 Planificación Temporal y Seguimiento del Proyecto	30
3.2.1 Planificación Temporal Inicial	30
3.2.2 Seguimiento del Proyecto.....	31
3.3 Presupuesto	34
3.3.1 Método de <i>Albretch</i>	35
3.3.2 Estimación por COCOMO	41
3.3.3 Comparativa de modelos de estimación.....	45
Capítulo 4: Análisis del sistema	46
4.1 Requisitos de Usuario	46
4.2 Casos de Uso	47
4.2.1 Listado de Casos de Uso	47
4.2.2 Diagrama de Casos de Uso	51
4.2.3 Especificación de Casos de Uso	51
4.3 Requisitos funcionales	62
4.3.1 Requisitos de información	65

4.4 Requisitos no funcionales	67
Capítulo 5: Diseño.....	69
5.1 Arquitectura lógica	69
5.2 Arquitectura física	70
5.3 Modelo de datos	72
5.3.1 Modelo lógico.....	72
5.3.2 Diccionario de datos	73
5.3.3 Consideraciones adicionales	76
5.4 Categorización de artículos según su veracidad	77
5.4.1 Parámetros de veracidad.....	77
5.4.2 Sistema de categorización de artículos y de aumento/disminución de la reputación	83
5.4.3 Sistema de <i>feed</i>	87
5.5 Diagramas de Secuencia.....	88
5.5.1 Diagrama de secuencia iniciar sesión.	88
5.5.2 Diagrama de secuencia publicar artículo	89
5.6 Diseño de interfaces de usuario	90
Capítulo 6: Implementación y pruebas	100
6.1 Cuestiones de implementación	100
6.1.1 <i>Backend</i>	100
6.1.2 <i>Frontend</i>	105
6.2 Pruebas	109
6.2.1 Pruebas de caja blanca	109
6.2.2 Pruebas de caja negra.....	110
Capítulo 7: Conclusiones y propuestas de mejora	117
7.1 Conclusiones	117
7.2 Propuestas de mejora	117
Webgrafía	119
Anexo 1 Manual de Instalación	121
Anexo 2 Manual de usuario	122

Índice de figuras

Ilustración 1 - Diagrama Árbol de características	19
Ilustración 2 - Logo node.js.....	20
Ilustración 3 - Logo express.js	21
Ilustración 4 - Logo MongoDB.....	21
Ilustración 5 - Logo Angular	22
Ilustración 6 - Logo Postman	23
Ilustración 7 - Logo git.....	23
Ilustración 8 - Logo github.....	24
Ilustración 9 - Logo Studio3T	24
Ilustración 10 - Logo Google Scholar	27
Ilustración 11 - Logo Reddit	27
Ilustración 12 - Modelo de desarrollo por prototipos	29
Ilustración 13 -Diagrama de Casos de Uso	51
Ilustración 14 - Diagrama de la Arquitectura Lógica	69
Ilustración 15 - Diagrama de la Arquitectura Física.....	71
Ilustración 16 - Diagrama Entidad-Relación.....	72
Ilustración 17 -Diagrama de Secuencia Iniciar Sesión.....	89
Ilustración 18 - Diagrama de Secuencia Publicar Artículo	90
Ilustración 19 - Interfaz Login	91
Ilustración 20 - Interfaz Registro	91
Ilustración 21 - Interfaz Feed	92
Ilustración 22 - Interfaz Explorador	93
Ilustración 23 - Interfaz Subir Artículo	94
Ilustración 24 - Interfaz Detalle Artículo	95
Ilustración 25 - Interfaz Perfil Ajeno - Información.....	96
Ilustración 26 - Interfaz Perfil Ajeno - Artículos.....	97
Ilustración 27 - Interfaz Perfil Personal - Información	98
Ilustración 28 - Interfaz Perfil Personal -Artículos	99
Ilustración 29 - Estructura Backend	100
Ilustración 30 - Controladores.....	101
Ilustración 31 - Middlewares.....	102
Ilustración 32 – Modelos Backend.....	102
Ilustración 33 - Estructura Frontend.....	105
Ilustración 34 - Componentes	107
Ilustración 35 - Servicios.....	107
Ilustración 36 - Modelos Frontend	108
Ilustración 37 -Modelo UserProfile y UserArticle	108
Ilustración 38 - Función Original (Lineal + Raíz Cuadrada)	79
Ilustración 39 - Función Suavizada Dividiendo por 100.....	80
Ilustración 40 - Función con Transición Suave (50 -70)	81
Ilustración 41 - Comparativa de Funciones.....	82
Ilustración 42 - Sistema de categorización de artículos y de aumento/disminución de la reputación I.....	84
Ilustración 43 - Sistema de categorización de artículos y de aumento/disminución de la reputación II	85

Ilustración 44 - Sistema de categorización de artículos y de aumento/disminución de la reputación III.....	85
Ilustración 45 - Sistema de categorización de artículos y de aumento/disminución de la reputación IV.....	87
Ilustración 46 - MU - Página de Registro	122
Ilustración 47 - MU - Página de Inicio de Sesión	123
Ilustración 48 - Página Home - Temáticas	124
Ilustración 49 - Página Home - Artículos.....	124
Ilustración 50 - Página Explorador - Buscar Temática desde el Home	125
Ilustración 51 - Página Explorador - Temáticas y Parámetros de Ordenación	125
Ilustración 52 - Vista Artículo	126
Ilustración 53 - Página Detalle Artículo.....	127
Ilustración 54 - Página Perfil no Personal - Información.....	128
Ilustración 55 - Página Perfil no Personal - Artículos	129
Ilustración 56 - Navbar.....	129
Ilustración 57 - Página Subir Artículo.....	130
Ilustración 58 - Página Perfil Personal - Información.....	130
Ilustración 59 - Página Perfil Personal - Menú Edición.....	131
Ilustración 60 - Página Perfil Personal - Artículos	132
Ilustración 61- Perfil Personal - Acciones	132
Ilustración 62 - Perfil Personal - Consultar Privilegios.....	133
Ilustración 63 - Perfil Personal - Eliminar Cuenta	133

Índice de tablas

Tabla 1 - Objetivos del Proyecto	14
Tabla 2 - Criterios de Aceptación para OBJ-01	15
Tabla 3 - Criterios de Aceptación para OBJ-02	15
Tabla 4 - Criterios de Aceptación para OBJ-03	16
Tabla 5 - Criterios de Aceptación para OBJ-04	16
Tabla 6 - Criterios de Aceptación para OBJ-05	17
Tabla 7 - Comparativa entre distintas aplicaciones	28
Tabla 8 - Seguimiento Diseño inicial y especificación de requisitos	31
Tabla 9 – Seguimiento Desarrollo del prototipo inicial y validación de funcionalidades esenciales	32
Tabla 10 - Seguimiento Iteraciones y mejoras progresivas	33
Tabla 11 - Seguimiento Pruebas finales y documentación	33
Tabla 12 - Seguimiento Resumen global y conclusiones	34
Tabla 13 - Tabla grado complejidad para Entradas Externas	36
Tabla 14 - Entradas Externas	36
Tabla 15 - Tabla grado de complejidad para Salidas Externas	36
Tabla 16 - Salidas Externas	36
Tabla 17 – Tabla Grado de complejidad para Consultas Externas	37
Tabla 18 - Consultas Externas	37
Tabla 19 - Tabla de grado de complejidad para ALI (Archivos internos) y AIE (Archivos externos)	37
Tabla 20 – ALI (Archivos internos) y AIE (Archivos externos)	37
Tabla 21 - Puntos de Función sin Ajustar	38
Tabla 22 - Ponderación Factor de Ajuste	38
Tabla 23 - Características para el cálculo del Factor de Ajuste	39
Tabla 24 – Costes de Componentes Hardware	41
Tabla 25 - Estimación Económica Final por el Método de Albretch	41
Tabla 26 - Equivalencia de Líneas de Código por Puntos de Función según la tecnología utilizada	42
Tabla 27 - Valor de los Factores	43
Tabla 28 - Cálculo del Factor de Ajuste COCOMO	44
Tabla 29 - Constantes Modelo Orgánico	44
Tabla 30 - Comparativa Modelos de Estimación	45
Tabla 31 - Requisitos de Usuario	47
Tabla 32 - Casos de Uso	49
Tabla 33 - Relación entre Casos de Uso y Requisitos de Usuario	50
Tabla 34 - Especificación CU-01	52
Tabla 35 - Especificación CU-02	53
Tabla 36 - Especificación CU-03	53
Tabla 37 - Especificación CU-04	54
Tabla 38 - Especificación CU-05	55
Tabla 39 - Especificación CU-06	55
Tabla 40 - Especificación CU-07	56
Tabla 41 - Especificación CU-07	57
Tabla 42 - Especificación CU-09	58
Tabla 43 - Especificación CU-10	58
Tabla 44 - Especificación CU-11	59
Tabla 45 - Especificación CU-12	60

Tabla 46 - Especificación CU-13.....	60
Tabla 47 - Especificación CU-14.....	61
Tabla 48 - Especificación CU-15.....	61
Tabla 49 - Especificación CU-16.....	62
Tabla 50 - Requisitos de Usuario.....	65
Tabla 51 - Requisitos de Información.....	66
Tabla 52 - Requisitos no Funcionales.....	68
Tabla 53 - Diccionario de Datos - Usuario	74
Tabla 54 - Diccionario de Datos - Artículo.....	75
Tabla 55 - Diccionario de Datos - Temática	75
Tabla 56 - Bloque de datos authorInfo	76
Tabla 57 - Reglas de publicación.....	77
Tabla 58 - Reglas de voto	78
Tabla 59 - Comparativa entre funciones de peso de voto.....	83
Tabla 60 - Categorías de un artículo	86
Tabla 61 - Repercusiones en la reputación	86
Tabla 62 - PCN-01 Crear cuenta de usuario	110
Tabla 63- PCN-02 - Inicio de Sesión.....	110
Tabla 64 - PCN-03 - Verificar sesión.....	111
Tabla 65 - PCN-04 - Listar artículos del feed.....	111
Tabla 66 - PCN-05 - Listar temáticas del feed	111
Tabla 67 - PCN-06 - Ver temática.....	112
Tabla 68 - PCN-07 - Visualizar detalle artículo.....	112
Tabla 69 - PCN-08 - Visualizar perfil.....	112
Tabla 70 - PCN-09 - Listar actividad perfil.....	112
Tabla 71 - PCN-10 - Listar artículos por autor.....	113
Tabla 72 - PCN-11 - Subir un artículo	113
Tabla 73 - PCN-12 - Eliminar un artículo	113
Tabla 74 - PCN-013 - Eliminar un artículo	114
Tabla 75 - PCN-14 - Votar un artículo.....	114
Tabla 76 - PCN-15 - Explorador artículos.....	114
Tabla 77 - PCN-16 - Editar perfil	114
Tabla 78 - PCN-01 - Añadir acreditación	115
Tabla 79 - PCN-18 - Eliminar acreditación	115
Tabla 80 - PCN-19 -Consultar privilegios.....	115
Tabla 81 - PCN-20 - Cerrar sesión	116
Tabla 82 - PCN-21 - Eliminar cuenta con artículos.....	116
Tabla 83 - PCN-22 – Eliminar cuenta sin artículos	116

Capítulo 1: Introducción

1.1 Organización del documento

Este primer apartado explica la manera en la que está organizada la documentación del proyecto, el cual consta de ocho capítulos y el anexo de referencias. A continuación, se realiza una breve explicación del contenido de cada capítulo:

- **Capítulo 1-Descripción del proyecto:** Presenta los aspectos genéricos del proyecto como son la introducción, motivación, objetivos planteados, el alcance previsto y las tecnologías utilizadas.
- **Capítulo 2-Estado del arte:** Análisis de otras aplicaciones o proyectos afines y su comparativa. De este modo se justifica la aportación del proyecto.
- **Capítulo 3-Planificación y presupuestos:** Metodología de trabajo aplicada, estimación de los costes del proyecto y secuencia temporal de las tareas.
- **Capítulo 4-Análisis del sistema:** Detalla los requisitos funcionales y no funcionales que debe cumplir la aplicación, además de describir los actores implicados y sus interacciones.
- **Capítulo 5-Diseño:** Muestra la arquitectura del sistema tanto a nivel físico como lógico, el sistema de categorización de artículos y la veracidad, diagramas de secuencia y actividad, modelo de datos y el diseño de las interfaces de usuario.
- **Capítulo 6-Implementación y pruebas:** Explica la implementación del *backend* y del *frontend*, junto con el resto de las funcionalidades y algoritmos desarrollados para la aplicación. Reúne todas las verificaciones y test realizados para garantizar el correcto funcionamiento de la plataforma, incluyendo los resultados obtenidos.
- **Capítulo 7-Conclusiones y mejoras:** Ofrece una evaluación del proyecto ya finalizado, reflexiona sobre su éxito en relación con los objetivos y sugiere propuestas de mejora a futuro.
- **Bibliografía:** Recopila las fuentes consultadas que han servido de apoyo para desarrollar el trabajo.
- **Anexo 1-Manual de instalación:** Detalla los pasos necesarios para instalar y desplegar la aplicación.
- **Anexo 2-Manual de usuario:** Proporciona instrucciones de uso de la aplicación web, abarcando sus funcionalidades y recomendaciones.

1.2 Introducción

Hoy en día, vivimos inmersos en un entorno digital donde cualquier persona con conexión a Internet puede acceder a una cantidad de información que, hasta hace nada, habría sido inimaginable. El auge de las redes sociales, los dispositivos móviles y las plataformas de contenido han propiciado que la búsqueda de noticias y datos se realice de forma inmediata y desde cualquier lugar. Sin embargo, este escenario brinda un arma de doble filo: si bien hace posible la difusión rápida de conocimientos y la democratización del acceso a la información, también puede ser el caldo de cultivo perfecto para propagar datos inexactos o medias verdades.

Las capacidades de muchas plataformas ya superan en la mayoría de los casos en alcance y velocidad a los medios tradicionales. Plataformas como X (anteriormente Twitter), Reddit o Instagram han tomado el relevo de periódicos y noticieros en el día a día de millones de usuarios, convirtiéndose en la principal ventana al mundo para buena parte de la población. Estas capacidades podrían suponer un beneficio para la sociedad siempre y cuando se usen de la manera correcta, es por ello por lo que hay que buscar una serie de herramientas para fomentar su buen uso y evitar de esta manera la propagación de informaciones erróneas, lo cual enturbia el debate y complica mucho la búsqueda de fuentes de información fiables.

Conscientes de este problema, el proyecto nace como una respuesta que busca proponer una plataforma web especializada en la publicación de artículos divulgativos, donde la calidad y la veracidad se sitúen en primer plano. En el proyecto, se contemplan mecanismos de autorregulación basados en la propia comunidad, de modo que sean los usuarios los encargados de publicar, valorar y mejorar la información compartida.

1.3 Motivación

La idea de desarrollar una plataforma web para la publicación y validación de artículos divulgativos surge de la preocupación creciente ante la desinformación, la cual se ha asentado gracias a la inmediatez y el alcance viral que caracterizan a las redes sociales. El objetivo principal es dar a la sociedad una herramienta transparente y colaborativa que fomente la verificación de datos de forma colectiva, de modo que los propios usuarios asuman un rol activo en la detección y corrección de informaciones inexactas.

Asimismo, el desarrollo de la plataforma responde a la necesidad de contar con un espacio de consulta en el que la calidad, la veracidad y la accesibilidad sean prioridades. Por un lado, no es requisito ser experto en la materia para publicar información; por otro, se incentiva la responsabilidad individual a la hora de publicar, sabiendo que toda afirmación

podría ser sometida al escrutinio de la comunidad y puede tener repercusiones negativas para el usuario con el fin de velar por el buen funcionamiento de la plataforma. En definitiva, ofrecer una plataforma centrada en la fiabilidad y calidad de los datos y que sea un punto de encuentro para usuarios que deseen informarse y compartir información de forma segura.

1.4 Objetivos

El objetivo principal de FACTOR es proporcionar una plataforma intuitiva y sencilla para buscar y publicar información de calidad. Este objetivo principal es dividido en los siguientes subobjetivos:

ID Objetivo	Nombre Objetivo
OBJ-01	Permitir a los usuarios la publicación y visualización de artículos de forma sencilla e intuitiva.
OBJ-02	Diseñar un sistema de votaciones y verificación colectiva que permita la categorización y calidad de la información.
OBJ-03	Implementar un sistema de reputación que restrinja acciones en función de la credibilidad del usuario, premiando las contribuciones positivas y penalizando las negativas.
OBJ-04	Desarrollar un sistema de <i>feed</i> que combine las tendencias de la comunidad y la calidad de los contenidos.
OBJ-05	Desarrollar perfiles donde se refleje la trayectoria, aportaciones y confiabilidad de cada usuario.

Tabla 1 - Objetivos del Proyecto

A continuación, se enumeran los criterios de aceptación asociados a cada objetivo para su cumplimiento.

ID Criterio Aceptación	ID Objetivo asociado	OBJ-01
	Descripción Criterio de Aceptación	
CA-01	Los usuarios deben poder subir un artículo (PDF y metadatos) en 3 pasos o menos.	
CA-02	Cada artículo publicado se muestra en el <i>feed</i> o en el buscador con título, descripción, jerarquía de temáticas, botones de interacciones, información del autor e información de la calidad.	
CA-03	El detalle de un artículo publicado se muestra con título, descripción, jerarquía de temáticas, visor de pdf, botones de interacciones, fuente, información del autor e información de la calidad.	
CA-04	El perfil de un autor mostrará sus artículos publicados y sus interacciones con título, descripción, jerarquía de temáticas, botones de interacciones e información de la calidad.	
CA-05	El explorador de artículos deberá permitir filtrar por cualquier nivel en la jerarquía de temáticas, por veracidad y por fecha de publicación.	

Tabla 2 - Criterios de Aceptación para OBJ-01

ID Criterio Aceptación	ID Objetivo asociado	OBJ-02
	Descripción Criterio de Aceptación	
CA-01	Los artículos dispondrán de botones para su votación tanto positiva como negativa.	
CA-02	El artículo se encasilla después de un número determinado de votos en una categoría según su calidad.	
CA-03	El <i>feed</i> priorizará artículos categorizados como “verificados”.	
CA-04	Los votos de los diferentes usuarios pesarán de manera diferente en base a su reputación.	
CA-05	Los usuarios reciben retroalimentación visual inmediata al votar, mostrando la puntuación, la categoría (en caso de que cambie con ese voto) y el recuento actualizado de votos.	
CA-06	El artículo será susceptible de volver a ser categorizado si consigue un número determinado de votos.	

Tabla 3 - Criterios de Aceptación para OBJ-02

ID Criterio Aceptación	ID Objetivo asociado	OBJ-03
	Descripción Criterio de Aceptación	
CA-01	Los votos realizados por los usuarios repercutirán positiva o negativamente en su reputación.	
CA-02	En función de su reputación, se establecerán las veces que un usuario puede publicar un artículo al mes y las veces que puede votar otros artículos a la semana.	
CA-03	En base a la reputación que tenga un usuario a la hora de publicar un artículo, dicho artículo tendrá o no un impulso inicial en términos de veracidad.	
CA-04	En base a la reputación que tenga un usuario a la hora de publicar un artículo, dicho artículo tendrá o no un impulso inicial en términos de veracidad.	
CA-05	La reputación de un usuario se verá afectada positiva o negativamente en función de la categorización de sus artículos publicados.	
CA-06	La reputación de cada usuario será visualizada mediante un sistema de etiquetas para indicar su confiabilidad.	

Tabla 4 - Criterios de Aceptación para OBJ-03

ID Criterio Aceptación	ID Objetivo asociado	OBJ-04
	Descripción Criterio de Aceptación	
CA-01	El <i>feed</i> mostrará los artículos destacados en base a un índice creado a partir de su fecha de publicación, su veracidad y su categoría.	
CA-02	El <i>feed</i> mostrará las temáticas donde se han publicado artículos recientemente.	
CA-03	El <i>feed</i> permitirá filtrar por temáticas de cualquier nivel, por veracidad y por fecha de publicación.	

Tabla 5 - Criterios de Aceptación para OBJ-04

ID Criterio Aceptación	ID Objetivo asociado	OBJ-05
	Descripción Criterio de Aceptación	
CA-01	Permitir al usuario editar la información de su propio perfil, modificando su nombre, apellidos, fecha de nacimiento e imagen de perfil y eliminando o añadiendo sus acreditaciones.	
CA-02	Permitir al usuario en su propio perfil visualizar y eliminar sus artículos publicados, y en un perfil ajeno visualizar los artículos publicados por el usuario visitado.	
CA-03	Visualizar en los perfiles propios y ajenos información del usuario propietario del perfil, así como un registro de sus artículos votados.	
CA-04	Dotar al usuario en su propio perfil de una barra de acciones para cerrar sesión, consultar sus privilegios de votos y publicaciones y eliminar su cuenta.	
CA-05	Permitir al usuario al querer eliminar su cuenta mantener sus artículos en la aplicación.	

Tabla 6 - Criterios de Aceptación para OBJ-05

1.5 Alcance

FACTOR está destinado a cualquier tipo de persona que quiera buscar o compartir información, no solo a personas altamente cualificadas, sino también a personas con menos cualificación que ya sea por talento o esfuerzo aportan contenido de calidad para la sociedad.

La plataforma es intuitiva y directa, permitiendo contribuir y explorar de manera sencilla y rápida, además de ser completamente transparente en todos los indicadores de calidad del contenido.

Como consideración para tener en cuenta, la primera versión de esta aplicación web estará destinada a la rama del conocimiento de la Informática, siendo dividida en las diferentes ramas y subramas de esta disciplina. Con esta medida podemos asegurar el buen funcionamiento de la aplicación, en un entorno acotado y que, entre otras cosas, permite una búsqueda muy precisa de entre todas las subdisciplinas que comprende la Informática.

En una segunda etapa, estas características pueden ampliarse tanto horizontal, como verticalmente. Verticalmente ya que el sistema permitirá granularizar las ramas y subramas existentes dependiendo de las necesidades de la comunidad, o de avances en investigación, y horizontalmente, ya que se podrá abrir en cualquier momento a otras

ramas o subramas diferentes de conocimiento. Estas dos variables aportan mucha flexibilidad a la aplicación y brindarán al usuario información de cualquier materia que esté buscando de forma precisa y sencilla.

Con el fin de mostrar de manera más visual las características de la aplicación, se ha elaborado el siguiente árbol de características:

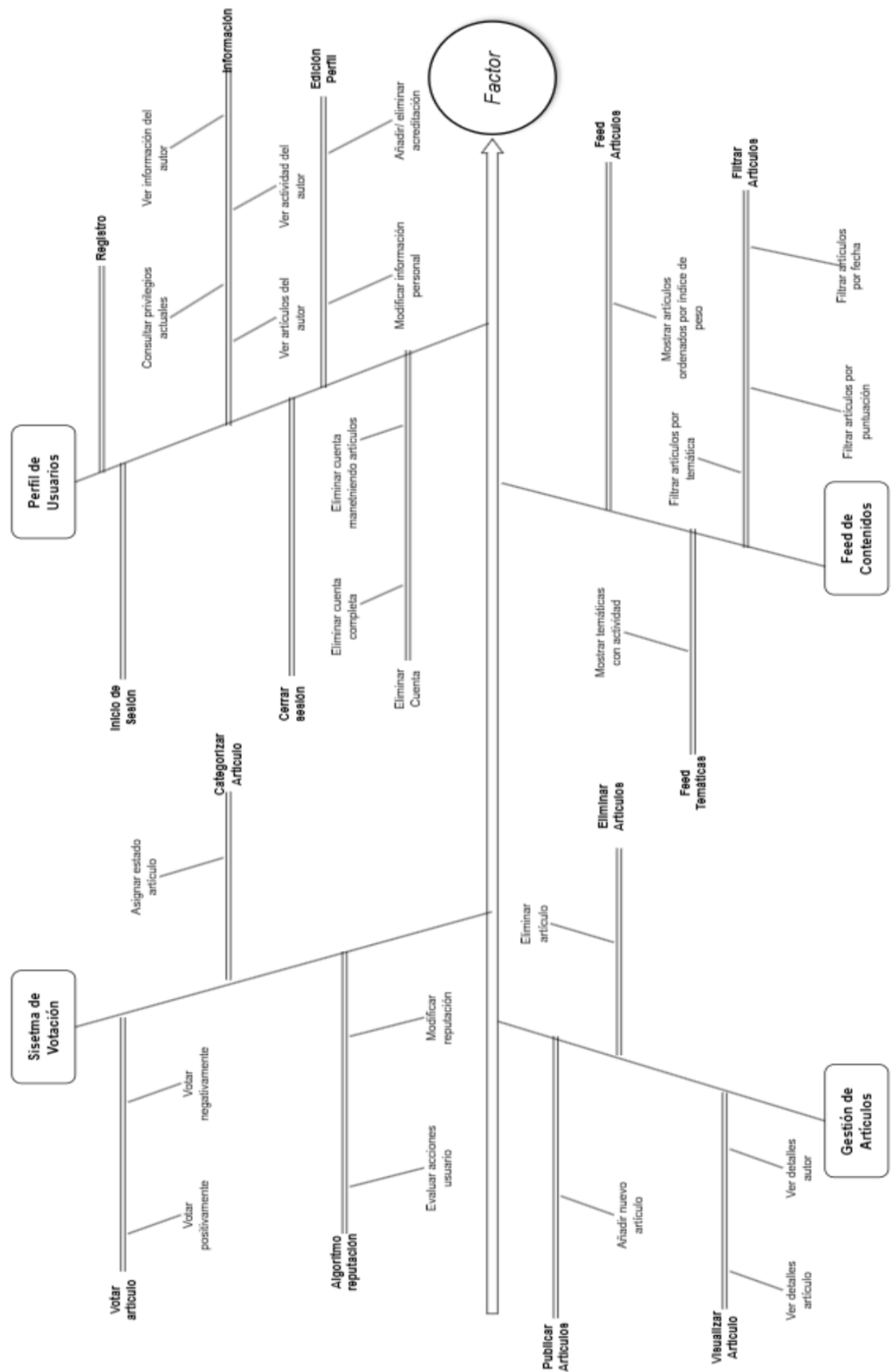


Ilustración 1 - Diagrama Árbol de características

1.6 Entorno y tecnologías utilizadas

FACTOR ha sido desarrollada utilizando una pila MEAN (*MongoDB, Express, Angular y Node.js*), junto con una serie de herramientas de apoyo para facilitar el desarrollo, pruebas y gestión del código.

A continuación, se enumerarán las principales características de las tecnologías utilizadas, divididas en *backend* y *frontend*, junto con las herramientas de apoyo.

1.6.1 Backend

1.6.1.1 Node.js



Ilustración 2 - Logo node.js

Es un entorno de ejecución para JavaScript en el lado del servidor. Permite ejecutar código JavaScript fuera del navegador, lo que lo hace ideal para construir servidores y aplicaciones escalables.

Características principales:

- Arquitectura basada en eventos y no bloqueante (modelo asíncrono).
- Alto rendimiento que permite ejecutar código de manera rápida y sencilla.
- Compatible con paquetes y módulos NPM.
- Permite construir APIs REST escalables.
- Perfecta para aplicaciones en tiempo real debido a su compatibilidad con *WebSockets*.

1.6.1.2 *Express.js*

The logo for Express.js, featuring the word "express" in a lowercase, sans-serif font.

Ilustración 3 - Logo express.js

Express es un *framework* para *Node.js* que hace más sencilla la creación de servidores web y API REST. Proporciona una estructura que permite la gestión eficiente de rutas y *middleware*.

Características principales:

- Facilita la creación de *API RESTful* con rutas definidas de manera clara.
- Compatible con *middleware* que permite, entre otras, agregar funcionalidades como la autenticación, registro de logs y manejo de errores.
- Ligero y modular.
- Se integra fácilmente con bases de datos como *MongoDB* mediante MONGOOSE.

1.6.1.3 *MongoDB*



Ilustración 4 - Logo MongoDB

MongoDB es una base de datos no relacional orientado a documentos, lo que quiere decir que, en lugar de almacenar datos en filas y tablas, los almacena en documentos JSON (BSON internamente). Es comúnmente utilizado por su flexibilidad y escalabilidad.

Características principales:

- Modelo flexible sin esquema fijo, adaptable a cambios en los datos.
- Escalabilidad horizontal, permite distribuir datos en múltiples servidores con el fin de mejorar el rendimiento.
- Integración con *Node.js* a través de MONGOOSE.

- Soporte para búsquedas avanzadas, agregaciones y búsqueda por índices.

1.6.2 *Frontend*

1.6.2.1 *Angular*



Ilustración 5 - Logo Angular

Angular es un *framework* de desarrollo *frontend* basado en *TypeScript*. Permite construir aplicaciones web de una sola página (SPA, *Single Page Application*) de manera modular y escalable.

Características principales:

- Arquitectura basada en componentes, lo que mejora la reutilización de código.
- Vinculación bidireccional de datos, lo que proporciona la sincronización entre el modelo de datos y la interfaz de usuario.
- Inyección de dependencias, lo que permite el desacoplamiento de componentes.
- Soporte nativo para enrutamiento mediante *Angular Router*.
- Integración con la API RxJS (*Reactive Extensions for JavaScript*), permitiendo programación reactiva y eventos asíncronos.

1.6.3 Herramientas adicionales

1.6.3.1 *Postman*



Ilustración 6 - Logo Postman

Postman es una herramienta que permite probar, documentar y automatizar APIs REST de manera gráfica y sencilla.

Características principales:

- Envío de peticiones HTTP a *endpoints* para probar su funcionalidad.
- Almacenamiento de pruebas a través de las colecciones.
- Manejo de variables y parametrización de peticiones.
- Generación automática de documentación API.

1.6.3.2 *Git*



Ilustración 7 - Logo git

Sistema de control de versiones que permite registrar los cambios en el código y colaborar con otros desarrolladores de manera eficiente.

Características principales:

- Control de versiones para evitar pérdidas de código y gestionar modificaciones.

- Permite trabajar en diferentes ramas, con el fin de desarrollar nuevas funcionalidades de forma segura sin afectar a la versión principal del código.
- Soporte para fusionar ramas (*merge*) y resolver conflictos.
- Compatibilidad con GITHUB, GITLAB y BITBUCKET para almacenamiento remoto de repositorios.

1.6.3.3 GITHUB

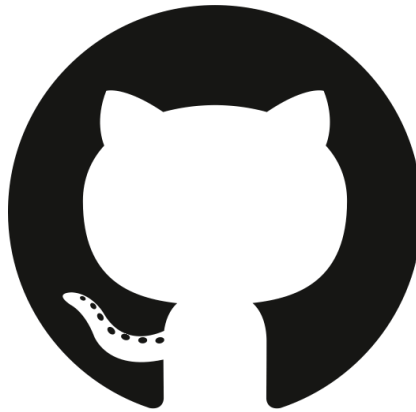


Ilustración 8 - Logo github

Plataforma basada en GIT que permite alojar y gestionar repositorios de código de manera remota y colaborativa.

Características principales:

- Repositorios públicos y privados con sistema de permisos.
- Gestión de *issues* y *pull requests*.
- Proporciona integración con CI/CD (*Continuous Integration/ Continuous Delivery*) para automatización de pruebas y despliegues.
- Histórico de cambios y aportaciones.

1.6.3.4 STUDIO 3T



Ilustración 9 - Logo Studio3T

STUDIO 3T es una herramienta GUI avanzada para la administración de bases de datos *MongoDB*, la cual proporciona herramientas de consulta y manipulación de datos sin necesidad de escribir comandos manuales a través de una terminal.

Características principales:

- Interfaz visual intuitiva para consultas y modificaciones.
- Importación y exportación de datos en varios formatos (JSON, CSV Y SQL).
- Optimización de consultas.

Capítulo 2: Estado del arte

En el capítulo de estado del arte se explicará la solución propuesta por FACTOR y el entorno de la aplicación. Se repasarán otras aplicaciones con enfoques similares, comparando los aspectos comunes y destacando las mejoras que presenta FACTOR con respecto al resto de aplicaciones analizadas.

2.1 Solución propuesta

FACTOR integra diferentes características clave compartidas por múltiples proyectos, como puede ser la divulgación de información, el estilo de “red social” y los mecanismos para la verificación del contenido. Sin embargo, es poco común encontrar un proyecto que combine estas tres características de manera efectiva.

Esta iniciativa impulsada por FACTOR pretende acercar contenido divulgativo de calidad a un público más amplio, ofreciendo una fuente segura de información. A diferencia de los repositorios tradicionales de *papers* que pueden resultar poco accesibles y/o atractivos, FACTOR presenta su red social con un diseño atractivo, sencillo e intuitivo. Su objetivo es hacer que el acceso al conocimiento no esté limitado a expertos o instituciones, sino que cualquiera, independientemente de su formación, pueda informarse de manera rigurosa y accesible.

2.2 Entorno de la aplicación

Con el fin de entender mejor la oferta de FACTOR en el mercado actual, compararemos sus características con las de otras aplicaciones con características similares, fundamentalmente su enfoque en la información verificada y su dinamismo y sistema de interacción de contenido.

2.2.1 Google Scholar

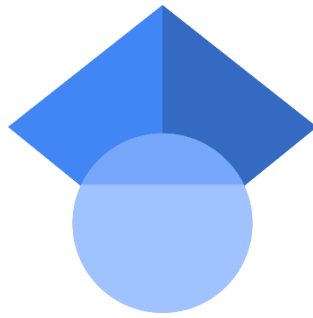


Ilustración 10 - Logo Google Scholar

Google Scholar es una de las principales herramientas para acceder a literatura científica y artículos publicados en revistas indexadas. Su sistema permite encontrar información de fuentes confiables, pero presenta varias limitaciones:

- Barreras de acceso, ya que muchos artículos están detrás de *paywalls* o requieren afiliación a una institución.
- Lenguaje técnico, porque está escrito por y para académicos e investigadores, pudiendo complicar mucho su comprensión para el público general.
- No permite realizar interacciones en los artículos publicados.

2.2.2 Reddit



Ilustración 11 - Logo Reddit

Reddit es una plataforma de foros (*subreddits*) donde los usuarios pueden publicar y votar contenido. Esta plataforma dispone de un sistema de votación comunitaria para destacar contenido, así como espacios de discusión donde los usuarios debaten sobre la información. Aun así, no dispone de un sistema estructurado y formal de verificación ni control en la calidad y cantidad de las publicaciones.

Si bien *Reddit* podría considerarse una de las principales competidoras de FACTOR, existen diferencias clave en su enfoque. Mientras que *Reddit* prioriza la flexibilidad y la diversidad de contenido sin restricciones temáticas, FACTOR se centra en ofrecer información precisa y verificada dentro del ámbito de la Informática, con la posibilidad

de expandirse a otras áreas del conocimiento. Esta especialización hace que FACTOR sea más riguroso y estructurado en comparación con la naturaleza más abierta y caótica de *Reddit*.

2.2.3 Comparativa

A continuación, se muestra una tabla comparativa de los dos proyectos mencionados en relación con FACTOR.

Característica	<i>Google Scholar</i>	<i>Reddit</i>	FACTOR
Enfoque	Repositorio de artículos científicos	Foros comunitarios con votación	Verificación y divulgación de información
Verificación	Basada en fuentes académicas	No estructurada (opiniones de usuarios)	Comunitaria y estructurada con sistema de reputación
Accesibilidad	Limitada por el lenguaje técnico y <i>paywalls</i>	Alta, pero con ruido informativo	Alta, con enfoque en accesibilidad y rigurosidad
Interacción	No permite comentarios ni votos	Votos y foros de discusión	Votación y categorización del contenido
Control de calidad	Alto, pero elitista y poco accesible	Bajo, depende de la comunidad	Balanceado, con control de calidad basado en reputación
Temáticas	Todas las áreas del conocimiento, con enfoque académico	Amplia variedad de temáticas sin restricciones	Inicialmente informática, pero escalable a otras áreas del conocimiento

Tabla 7 - Comparativa entre distintas aplicaciones

FACTOR se diferencia de los repositorios científicos al eliminar barreras de acceso y permitir una interacción más abierta. Al mismo tiempo, mejora la dinámica de las redes sociales al integrar un sistema de reputación y verificación, evitando la propagación de información falsa y se enfoca en esta etapa inicial en la rama de informática, lo que lo convierte en un espacio especializado, pero con gran potencial de crecimiento y escalabilidad.

Capítulo 3: Planificación y presupuestos

3.1 Metodología del trabajo

Para el desarrollo de este proyecto se ha escogido el modelo de desarrollo por prototipos, ya que se adapta perfectamente a la manera en la que ha sido construido el sistema. Este enfoque ha permitido el desarrollo y prueba de las funcionalidades clave de forma iterativa, permitiendo ajustes sobre la marcha y sin la necesidad de definir el sistema completo antes de su implementación.

¿Qué es el modelo de desarrollo por prototipos?

El modelo de desarrollo por prototipos es una metodología de desarrollo de software en la que se crea una versión inicial del sistema, la cual es mejorada progresivamente a través de iteraciones.

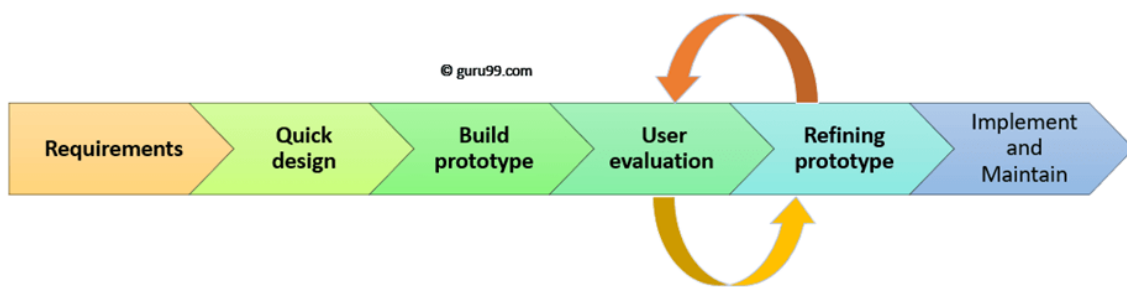


Ilustración 12 - Modelo de desarrollo por prototipos

Etapas y características del modelo de desarrollo por prototipos:

- En una primera etapa, se definen los requisitos iniciales, estableciendo los elementos esenciales del sistema, pero sin necesidad de profundizar en todos los detalles.
- La segunda etapa comprende la creación del prototipo inicial, es decir, una primera versión funcional con las características esenciales anteriormente definidas.
- En la tercera etapa se llevan a cabo las evaluaciones y ajustes necesarios, identificando áreas de mejora o funcionalidades adicionales, esta etapa se realiza de forma iterativa hasta la obtención del prototipo definitivo.
- Una vez refinado el sistema tras varias iteraciones, se implementa la versión final con todas las mejoras integradas.
- Por último, se realiza una fase de pruebas para asegurar el correcto comportamiento del sistema.

Esta metodología es ideal para este proyecto ya que permite tener flexibilidad en el diseño, sobre todo en etapas iniciales, además de permitir la validación temprana de las características principales. En etapas intermedias proporciona capacidad para realizar ajustes en los modelos y capacidad de desarrollo de funcionalidades en base a prioridades. Es por ello por lo que esta metodología fue la escogida para el proyecto, ya que aseguraba una integración gradual por prototipos, depurando el proyecto hasta su versión final.

3.2 Planificación Temporal y Seguimiento del Proyecto

En esta sección se describe la planificación temporal inicial del Trabajo Fin de Grado (TFG) junto con el seguimiento llevado a cabo durante su desarrollo. Este enfoque permite no solo detallar la estimación inicial de tiempos y recursos, sino también evaluar cómo se han ajustado estas previsiones en función del progreso real del proyecto.

Como se ha mencionado en capítulos anteriores, la metodología utilizada ha sido basada en prototipos. Esta metodología permite construir una versión funcional inicial del sistema que se refina progresivamente mediante iteraciones hasta alcanzar la implementación definitiva.

3.2.1 Planificación Temporal Inicial

Este TFG comenzó a inicios de septiembre de 2024, y se tiene como fecha prevista abril de 2025. La carga de trabajo inicial estimada es de 300 horas como está estipulado en el plan de estudios. Sin embargo, considerando posibles complicaciones y bloqueos, se ajustó la estimación a 385 horas. A continuación, se desglosa el tiempo inicialmente estimado para cada una de las fases del desarrollo:

- Diseño inicial y especificación de requisitos: 20 horas.
- Desarrollo del prototipo inicial y validación de funcionalidades esenciales: 110 horas.
- Iteraciones y mejoras progresivas: 150 horas.
- Pruebas finales y documentación: 105 horas.

Estas estimaciones sirvieron como base para la planificación general del proyecto y proporcionaron una hoja de ruta inicial sobre la que se organizó el trabajo.

3.2.2 Seguimiento del Proyecto

A medida que avanzaba el proyecto, se llevó a cabo un seguimiento continuo para comparar el progreso real con la planificación inicial. Esto permitió realizar ajustes necesarios en tiempo y esfuerzo para cumplir con los objetivos establecidos. En la tabla siguiente se presenta una comparativa entre las horas estimadas inicialmente y las horas reales empleadas en cada fase:

3.2.2.1 Diseño inicial y especificación de requisitos

Tarea	Horas est.	Horas reales	Desviación (%)	Observaciones
Definición objetivos TFG	2,00	1,00	-50,00	Objetivos Claros
Casos de uso iniciales	6,00	6,00	0,00	
Definición entidades	6,00	8,00	+25,00	Problemas al definir las relaciones.
Modelo de datos base	6,00	5,00	-16,66	Modelo de datos abierto hasta el desarrollo de las iteraciones.
Total	20,00	20,00	0,00	

Tabla 8 - Seguimiento Diseño inicial y especificación de requisitos

3.2.2.2 Desarrollo del prototipo inicial y validación de funcionalidades esenciales

Tarea	Horas est.	Horas reales	Desviación (%)	Observaciones
Configuración del entorno de desarrollo	6,00	8,00	+33,33	Problemas de dependencias y conexión con la bd.
Desarrollo y configuración estructura backend	8,00	8,00	0,00	Experiencia previa con la tecnología.
Desarrollo y configuración estructura frontend	6,00	6,00	0,00	Experiencia previa con la tecnología.

Desarrollo operaciones CRUD básicas	45,00	28,00	-37,77	Operaciones CRUD más sencillas en esta primera etapa de lo esperado.
Desarrollo de interfaces básicas y validación de formularios	25,00	15,00	-40,00	
Autenticación	20,00	23,00	+15,00	Bloqueos con las cabeceras y al establecer las cookies de sesión.
Total	110,00	88,00	-20,00	

Tabla 9 – Seguimiento Desarrollo del prototipo inicial y validación de funcionalidades esenciales

3.2.2.3 Iteraciones y mejoras progresivas

Tarea	Horas est.	Horas reales	Desviación (%)	Observaciones
Diseño del módulo de votaciones y reputación	10,00	15,00	+50,00	Bloqueo al decidir la evolución del peso por la reputación
Desarrollo del módulo de votaciones y reputación	40,00	40,00	0,00	
Evaluación del módulo de votaciones y reputación	4,00	4,00	0,00	El tiempo extra tomado en el diseño provocó que los resultados fueran los esperados.
Diseño del módulo de verificación colectiva y cambios en la reputación	10,00	15,00	+50,00	Esta fase se retrasó considerablemente con respecto a lo previsto debido a la complejidad del objetivo del módulo.
Desarrollo del módulo de verificación colectiva y cambios en la reputación	30,00	40,00	+33,33	Retraso debido a incompatibilidades en la integración con el módulo de la primera iteración.

Evaluación del módulo de verificación colectiva y cambios en la reputación	6,00	6,00	0,00	
Desarrollo del <i>feed</i> mediante el uso de los módulos integrados	5,00	5,00	0,00	
Configuración del estilo del frontal de la aplicación	45,00	45,00	0,00	
Total	150,00	170,00	+13,33	

Tabla 10 - Seguimiento Iteraciones y mejoras progresivas

3.2.2.4 Pruebas finales y documentación

Tarea	Horas est.	Horas reales	Desviación (%)	Observaciones
Búsqueda de artículos y llenado de la aplicación	5,00	5,00	0,00	
Pila de pruebas final	10,00	20,00	+100,00	Esta tarea se retrasó ya que aquí se refleja el tiempo invertido para solucionar varios GAP detectados en las pruebas.
Redacción de la memoria	90,00	100,00	+11,11	
Total	105,00	125,00	+19,04	

Tabla 11 - Seguimiento Pruebas finales y documentación

3.2.2.5 Resumen global y conclusiones

Fase	Horas est.	Horas reales	Desviación (%)	% del total
Diseño inicial y especificación de requisitos	20,00	20,00	0,00	4,96%
Desarrollo del prototipo inicial y validación de funcionalidades esenciales	110,00	88,00	-20,00	21,84%
Iteraciones y mejoras progresivas	150,00	170,00	+13,33	42,18%
Pruebas finales y documentación	105,00	125,00	+19,04	31,02%
Total	385,00	403,00	+4,67	100,00%

Tabla 12 - Seguimiento Resumen global y conclusiones

En resumen, la planificación temporal del proyecto ha resultado satisfactoria: de las 385 horas inicialmente previstas se han acabado invirtiendo 403, lo que supone una desviación global de apenas un 5 %. El esfuerzo empleado ha sido, de media, en torno a las 11 horas semanales, viéndose reducido en la fase de desarrollo del prototipo inicial e incrementado en las fases de iteración y de pruebas y redacción de la memoria. Este ajuste solo implicó un retraso de unas semanas, culminándose el proyecto en mayo en lugar de abril.

3.3 Presupuesto

En esta sección se lleva a cabo la estimación del presupuesto para FACTOR, para ello se tendrán tanto los recursos técnicos como los recursos humanos empleados a lo largo del ciclo de vida del software.

Esta estimación del presupuesto es esencial en cualquier proyecto, permitiendo dimensionar el esfuerzo requerido y facilitar la toma de decisiones estratégicas.

Para realizar esta estimación, se han aplicado dos métodos comúnmente utilizados en ingeniería de *software*: El método de *Albretch* o método de puntos de función y el modelo COCOMO.

3.3.1 Método de *Albretch*

El método de *Albretch* o método de puntos de función es un procedimiento conocido en ingeniería de software que permite estimar el tamaño del software en función de su funcionalidad percibida por el usuario, independientemente de la tecnología utilizada.

El método se basa en un conteo de funcionalidades perceptibles, divididas en 5 tipos cada tipo con peso asociado a su complejidad (baja, media y alta).

Funcionalidad	Descripción
Entradas Externas (EE)	Datos introducidos por el usuario en el sistema
Salidas Externas (SE)	Información proporcionada al usuario
Consultas Externas (CE)	Interacciones que implican entradas y salidas
Archivos Lógicos Internos (ALI)	Bases de datos o estructuras mantenidas por el sistema
Archivos de Interfaz Externa (AIE)	Datos utilizados por el sistema, pero mantenidos por otros

Una vez definido el método de *Albretch* se calculan los puntos de función no ajustados (PFNA), estos puntos representan el tamaño funcional del *software* sin tener en cuenta los factores de complejidad técnica o del entorno.

Para ello se utiliza una clasificación de cada componente según su complejidad y su número de datos elementales (DET), de archivos referenciados (FTR) o de subconjuntos lógicos (RET). asignándole un valor de puntos de función.

- **DET (*Data Element Type*):** Son tipos de datos elementales que el usuario puede ver o que afecte a la lógica del proceso, aplican a los EE, SE y CE.
- **FTR (*File Type Referenced*):** Son archivos lógicos leídos o referenciados en una transacción, se refieren a los ALI y AIE y aplican para los EE, SE y CE.
- **RET (*Record Element Type*):** Subconjunto lógico dentro de un ALI o un AIE.

Una vez explicados estos parámetros se puede realizar el cálculo de los puntos de función según los valores definidos en estas tablas (*Object Management Group*, 2013).

- **Entradas Externas:**

	DET		
FTR	1-4	5-15	>15
0-1	Baja	Baja	Media
2	Baja	Media	Alta
≥3	Media	Alta	Alta

Tabla 13 - Tabla grado complejidad para Entradas Externas

Aplicando la tabla anterior, se detalla a continuación una tabla con todas las entradas externas presentes en la aplicación, junto con sus grados de complejidad.

Entradas Externas	FTR	DET	Complejidad
Iniciar Sesión	0-1	1-4	Baja
Registro Usuario	0-1	5-15	Baja
Publicar Artículo	≥3	5-15	Alta
Modificar Perfil	2	5-15	Media
Votar Artículo	2	1-4	Baja
Eliminar Artículo	2	1-4	Baja
Eliminar Cuenta	2	1-4	Baja

Tabla 14 - Entradas Externas

- **Salidas Externas:**

	DET		
FTR	1-5	6-19	≥20
0-1	Baja	Baja	Media
2-3	Baja	Media	Alta
≥4	Media	Alta	Alta

Tabla 15 - Tabla grado de complejidad para Salidas Externas

En base a la tabla anterior se detallan todos los procesos de salidas externas presentes en la aplicación.

Salidas Externas	FTR	DET	Complejidad
Listado artículos <i>feed</i>	2-3	6-19	Media
Listado temáticas <i>feed</i>	2-3	1-5	Baja
Detalle artículo	2-3	6-19	Media
Listado actividad	2-3	6-19	Media
Listado artículos por autor	2-3	6-19	Media

Tabla 16 - Salidas Externas

- Consultas Externas

	DET		
FTR	1-4	5-15	≥15
0-1	Baja	Baja	Media
2	Baja	Media	Alta
≥3	Media	Alta	Alta

Tabla 17 – Tabla Grado de complejidad para Consultas Externas

Para el cálculo de los grados de complejidad de las consultas externas se utiliza la tabla anterior, en la siguiente tabla se detallan las complejidades calculadas para las CE de FACTOR.

Consultas Externas	FTR	DET	Complejidad
Explorar Artículos	≥3	5-15	Alta
Consultar perfil	0-1	5-15	Baja
Detalle artículo	≥3	5-15	Alta
Consultar privilegios	2	1-4	Baja

Tabla 18 - Consultas Externas

- Ficheros internos lógicos

	DET		
RET	1-19	20-50	≥50
1	Baja	Media	Alta
2-5	Media	Media	Alta
≥6	Alta	Alta	Alta

Tabla 19 - Tabla de grado de complejidad para ALI (Archivos internos) y AIE (Archivos externos)

ALI	RET	DET	Complejidad
Tabla artículos	3	20-50	Media
Tabla temas	2	1-19	Media
Tabla usuarios	2	1-19	Media

Tabla 20 – ALI (Archivos internos) y AIE (Archivos externos)

- Ficheros lógicos externos

En este proyecto no se ha hecho uso de ficheros lógicos externos por lo que no se tendrán en cuenta para el cálculo de los puntos de función no ajustados.

Componente	Complejidad	Peso	Total	Total x Peso
Entradas Externas	Baja	3	5	15
	Media	4	1	4
	Alta	6	1	6
Salidas Externas	Baja	4	1	4
	Media	5	4	20
	Alta	7	0	0
Consultas Externas	Baja	3	2	6
	Media	4	0	0
	Alta	6	2	12
Archivos Lógicos Internos	Baja	7	0	0
	Media	10	3	30
	Alta	15	0	0
Total				97

Tabla 21 - Puntos de Función sin Ajustar

Una vez se tienen los **PFNA**, el siguiente paso es aplicar el **Factor de Ajuste (FA)** para obtener los **Puntos de Función Ajustados (PFA)**. Este factor es calculado en base a 14 características generales del sistema evaluados en una escala de 0 a 5 según la siguiente tabla:

Valor	Descripción
0	Sin influencia
1	Influencia mínima
2	Influencia moderada
3	Influencia apreciable
4	Influencia significativa
5	Influencia muy fuerte

Tabla 22 - Ponderación Factor de Ajuste

Evaluando las características generales del sistema en base a esta escala resultaría en los siguientes grados de influencia:

Característica	Grado de influencia
Comunicación de datos	4
Procesamiento distribuido	3
Rendimiento	2
Configuración altamente utilizada	3
Tasa de transacciones	3
Entrada de datos en línea	4
Eficiencia del usuario final	3
Actualización en línea	4
Complejidad del procesamiento	4
Reusabilidad	3
Facilidad de instalación	2
Facilidad de operación	2
Múltiples localizaciones	0
Facilidad de cambios	3
Total	40

Tabla 23 - Características para el cálculo del Factor de Ajuste

Con los PFNA y la suma total de influencia (**TGI**) podemos aplicar la fórmula para calcular el **Factor de Ajuste (FA)** utilizando la siguiente fórmula:

$$FA = (0,01 \times TGI) + 0,65$$

Aplicando los valores calculados resultaría:

$$FA = (0,01 \times 40) + 0,65 = 1,05$$

Por último, solo tendríamos que calcular los **Puntos de Función Ajustados (PFA)** con la siguiente fórmula:

$$PFA = PFNA \times FA$$

Aplicando los datos calculados obtendríamos:

$$PFA = 97 \times 1,05 = 101,85$$

Estimación de la duración:

Una vez que tenemos los Puntos de Función Ajustados (PFA) podemos calcular el tiempo estimado para completar el proyecto.

Según datos recopilados de proyectos en la industria, un valor promedio de esfuerzo corresponde con 7 horas/persona por punto de función, en el caso estudiado nos quedarían:

$$\text{Esfuerzo total} = 101,85 \times 7 = 712,95 \text{ horas/persona}$$

Para estimar la duración total consideraremos una jornada laboral estándar de 8 horas y una disponibilidad de 21 días laborables al mes. Con estos datos se obtiene la duración total del proyecto en meses:

$$\text{Duración en meses} = 712,95 / (8 \times 21) \approx 4,24 \text{ meses}$$

Con esto, se determina una estimación para la duración del proyecto de aproximadamente 4,24 meses.

Estimación económica:

Para la estimación económica se tendrán en cuenta tanto los recursos técnicos como recursos humanos necesarios para el desarrollo del proyecto.

- Recursos humanos:

En base a diversas fuentes consultadas, el salario medio de un desarrollador de software en España en 2024 se sitúa en 2.613 €/mes, ya que se ha estimado una duración del proyecto de 4,24 meses el coste humano del proyecto resultaría en:

$$\text{Coste humano} = 2.613 \times 4,24 = 11.079,12\text{€}$$

- Recursos técnicos:

La parte de recursos técnicos comprende todas las herramientas tanto *software* como *hardware* requeridas para la elaboración del proyecto.

En cuanto al *software* utilizado no aporta ningún coste al proyecto ya que todo el *software* utilizado es de código abierto o gratuito.

El *hardware* utilizado para el desarrollo se detalla en la siguiente tabla:

Componente	Precio	Vida Útil(años)	Porcentaje de uso	Coste real
Ordenador	1050 €	5	7,00%	73,50 €
Ratón	40 €	3	11,78%	4,71 €
Teclado	55 €	4	8,34%	4,58 €
Total				82,79 €

Tabla 24 – Costes de Componentes Hardware

Por último, habría que tener en cuenta los costes operativos asociados, fundamentalmente la luz y la conexión a internet. Con datos de abril del 2025 y aplicados a los 4,24 meses estimados obtendríamos un coste de 130,21 € euros en luz y 84,38 € asociados a una tarifa económica con fibra óptica promedio.

Por tanto, la estimación económica final del proyecto nos quedaría de la siguiente manera:

Categoría	Coste Estimado
Recursos humanos	11.079,12 €
Recursos técnicos	82,79 €
Costes operativos	214,59 €
Total	11.376,50 €

Tabla 25 - Estimación Económica Final por el Método de Albretch

3.3.2 Estimación por COCOMO

El modelo COCOMO (*Constructive Cost Model*) es otro método muy utilizado en ingeniería de software basado en la estimación de esfuerzo, tiempo y coste. COCOMO se aplica sobre el tamaño del software en términos de líneas de código fuente o KLOC.

El modelo define tres modos de desarrollo, cada uno con sus constantes propias:

- **Modo Orgánico:** Proyectos pequeños, normalmente con menos de 50 KLOC (50.000 líneas de código), con tecnología conocida y requisitos bien definidos además de pocos riesgos técnicos y equipos pequeños, pero con experiencia.
- **Modo Semiacoplado:** Referido a proyectos de tamaño medio-grande (50-300 KLOC), con requisitos menos definidos o en constante evolución y que hacen uso de nuevas tecnologías y arquitecturas mixtas. Los equipos son mixtos incluyendo a personal experimentado y con menos experiencia.
- **Modo Empotrado:** Aplica a proyectos de alta complejidad (más de 300 KLOC), con requisitos muy estrictos y un entorno técnico muy restringido. Lo conforman equipos complejos y multidisciplinarios.

En el caso de FACTOR, el modo más adecuado es el orgánico debido a las siguientes razones:

1. Proyecto pequeño e individual.
2. Tecnología común.
3. Requisitos bien definidos.
4. Carece de integración con servicios externos.
5. Carece de restricciones externas y de requerimientos técnicos especiales.

Tras la elección del modo de desarrollo el siguiente paso consiste en estimar la cantidad de líneas de código necesarias para el desarrollo del proyecto, según las tecnologías utilizadas podemos realizar la conversión de PFA a KLOC por lenguaje:

Tecnología	LOC por PFA
HTML/CSS	15-30
<i>TypeScript</i>	50-55
<i>Node</i>	50-55

Tabla 26 - Equivalencia de Líneas de Código por Puntos de Función según la tecnología utilizada

Suponiendo que el 70% del esfuerzo recae en *TypeScript* y *Node* y el 30% restante en HTML obtenemos que:

$$LOC/PFA = (0,7 \times 52) + (0,3 \times 25) = 36,4 + 7,5 = 43,9 LOC/PFA$$

Que aproxima 43,9 líneas de código por punto de función, aplicando los PFNA calculados anteriormente obtenemos los KLOC estimados:

$$LOC = 97 \times 43,9 = 4.258 \text{ LOC} = 4,26 \text{ KLOC}$$

Una vez calculados los KLOC necesarios se evalúan los factores de coste en base a su peso en el proyecto según la siguiente tabla:

Factor	Descripción	Muy Baja	Baja	Nominal	Alta	Muy Alta	Extra Alta
RELY	Fiabilidad requerida del sistema	0,75	0,88	1,00	1,15	1,40	N/A
DATA	Tamaño de la base de datos en relación con el código	N/A	0,94	1,00	1,08	1,16	N/A
CPLX	Complejidad lógica del sistema	0,70	0,85	1,00	1,15	1,30	1,65
TIME	Restricciones de tiempo de ejecución	N/A	N/A	1,00	1,11	1,30	1,66
STOR	Restricciones de almacenamiento/memoria	N/A	N/A	1,00	1,06	1,21	1,56
VIRT	Volatilidad del entorno de ejecución	N/A	0,87	1,00	1,15	1,30	N/A
TURN	Tiempo de respuesta requerido del sistema	N/A	0,87	1,00	1,07	1,15	N/A
ACAP	Capacidad del analista	1,46	1,19	1,00	0,86	0,71	N/A
AEXP	Experiencia en el dominio de la aplicación	1,29	1,13	1,00	0,91	0,82	N/A
PCAP	Habilidad del programador	1,42	1,17	1,00	0,86	0,70	N/A
VEXP	Experiencia con el entorno (SO, librerías)	1,21	1,10	1,00	0,90	N/A	N/A
LEXP	Experiencia con el lenguaje de programación	1,14	1,07	1,00	0,95	N/A	N/A
TOOL	Herramientas de desarrollo utilizadas	1,24	1,10	1,00	0,91	0,83	N/A
MODP	Metodología y prácticas de desarrollo	1,24	1,10	1,00	0,91	0,82	N/A
SCED	Restricción de cronograma/plazos	1,23	1,08	1,00	1,04	1,10	N/A

Tabla 27 - Valor de los Factores

En el caso de FACTOR obtenemos los siguientes valores de los factores de ajuste:

Factor	Evaluación	Valor FA
RELY	Alta	1,15
DATA	Nominal	1,00
CPLX	Alta	1,15
TIME	Baja	N/A
STOR	Baja	N/A
VIRT	Muy Baja	N/A

TURN	Alta	1,07
ACAP	Nominal	1,00
AEXP	Nominal	1,00
PCAP	Nominal	1,00
VEXP	Nominal	1,00
LEXP	Alta	N/A
TOOL	Alta	0,83
MODP	Baja	1,10
SCED	Nominal	1,00

Tabla 28 - Cálculo del Factor de Ajuste COCOMO

Tras haber evaluado los factores de ajuste se calcula el Factor de ajuste del esfuerzo como el producto de los multiplicadores correspondientes a cada uno de los valores asignados:

$$EAF = 1,15 \times 1 \times 1,15 \times 1,07 \times 1 \times 1 \times 1 \times 1 \times 0,83 \times 1,10 \times 1 = 1,29$$

Calculado el EAF ya se pueden estimar los tiempos y costes utilizando las constantes del modelo orgánico:

Modelo	a	b	c	d
Orgánico	3,2	1,05	2,5	0,38

Tabla 29 - Constantes Modelo Orgánico

Esfuerzo estimado (persona/mes):

$$E = a \times (KLOC)^b \times EAF$$

$$E = 3,2 \times (4,26)^{1,05} \times 1,29 = 18,91 \text{ personas/mes}$$

Duración estimada (meses):

$$D = c \times E^d$$

$$D = 2,5 \times 18,91^{0,38} = 7,64 \text{ meses}$$

Personal estimado

$$P = \frac{E}{D} = \frac{18,91}{7,64} = 2,47 \text{ personas}$$

3.3.3 Comparativa de modelos de estimación

Realizando una comparativa del coste humano obtenido en ambas estimaciones obtenemos la siguiente tabla:

Modelo	Personas	Meses	Total Coste Humano
<i>Albretch</i>	1,00	4,24	$4,24 \times 2.613 = 11.079,12$
COCOMO	2,47	7,64	$18,91 \times 2.613 = 49.414,83$

Tabla 30 - Comparativa Modelos de Estimación

La conclusión comparativa que se puede extraer tras analizar ambos modelos es que el modelo de *Albretch* ofrece una estimación más razonable para el proyecto, teniendo en cuenta el contexto de que es un trabajo de fin de grado individual y con una duración y costes moderados.

Por otro lado, el modelo de COCOMO sería más adecuado para un entorno empresarial y real, incluyendo en los factores de ajuste múltiples variables humanas, técnicas y organizativas.

Capítulo 4: Análisis del sistema

Este capítulo recoge un análisis profundo de los requisitos y casos de uso de FACTOR, agrupando todos los requisitos especificados para el prototipo inicial y los añadidos en las posteriores fases iterativas.

4.1 Requisitos de Usuario

Los requisitos de usuario se definen como las necesidades, expectativas y objetivos que un usuario final tiene con respecto a un sistema o aplicación. Especifican qué debe poder hacer un usuario en el sistema y cómo debe ser la experiencia de uso. Los requisitos de usuario están formulados con un lenguaje natural y desde la perspectiva del usuario, sin profundizar en detalles o lenguaje más técnico.

A continuación, se muestra la tabla de requisitos de usuario para FACTOR.

ID Requisito	Descripción
RU-01	El usuario debe poder registrarse en la plataforma.
RU-02	El usuario debe poder iniciar sesión en la plataforma.
RU-03	El usuario debe poder cerrar sesión en la aplicación una vez iniciada sesión.
RU-04	El usuario debe poder publicar artículos en formato PDF con título, descripción, temática y fuente.
RU-05	El usuario debe poder ver un <i>feed</i> con los artículos y temáticas recomendados por la aplicación.
RU-06	El usuario debe poder visualizar el detalle de los artículos publicados.
RU-07	El usuario debe poder votar positiva y negativamente artículos
RU-08	El usuario debe poder realizar una búsqueda de artículos por temática.
RU-09	El usuario debe poder ordenar la búsqueda de artículos por fecha y por veracidad.
RU-10	El usuario debe poder ver su información detallada en su perfil.
RY-11	El usuario debe poder ver la información básica de otros usuarios en sus respectivos perfiles.
RU-12	El usuario debe poder editar la imagen, nombre, apellidos y acreditaciones de su perfil.
RU-13	El usuario debe poder ver sus artículos publicados.

RU-14	El usuario debe poder eliminar sus artículos publicados en su perfil.
RU-15	El usuario debe poder ver los artículos publicados por otros usuarios en sus respectivos perfiles.
RU-16	El usuario debe poder ver su actividad en su perfil.
RU-17	El usuario debe poder ver la actividad de otros usuarios en sus respectivos perfiles.
RU-18	El usuario debe poder ver en tiempo real las actualizaciones en los indicadores al votar un artículo.
RU-19	El usuario debe poder consultar en su perfil de los privilegios de votos y publicaciones de los que dispone.
RU-20	El usuario debe poder eliminar su cuenta.
RU-21	El usuario debe poder eliminar su cuenta manteniendo sus artículos en la aplicación.

Tabla 31 - Requisitos de Usuario

4.2 Casos de Uso

Una vez definidos los requisitos de usuario, es necesario representar cómo estos interactúan con el sistema para lograr sus objetivos, para ello hacemos uso de los casos de uso.

Se define un caso de uso como una descripción detallada de cómo un usuario interactúa con un sistema para lograr un objetivo específico. Son útiles para definir los distintos escenarios en los que el usuario puede utilizar la aplicación y sus variaciones, además de las respuestas que debe ofrecer el sistema antes estas interacciones.

En la siguiente sección se presentan los casos de uso de FACTOR a través de un diagrama de casos de uso UML y documentación sobre cada caso en concreto:

4.2.1 Listado de Casos de Uso

A continuación, se detallan los diferentes casos de uso definidos para la aplicación.

ID	Caso de Uso	Descripción
CU-01	Registrarse en la aplicación	El sistema permite al usuario registrarse en la aplicación.

CU-02	Iniciar sesión	El sistema permite al usuario iniciar sesión en la aplicación.
CU-03	Cerrar sesión	El sistema permite al usuario cerrar la sesión una vez la haya iniciado.
CU-04	Publicar artículo	El sistema permite al usuario publicar un artículo.
CU-05	Ver <i>feed</i>	El sistema permite al usuario ver el <i>feed</i> con los artículos y temáticas recomendadas.
CU-06	Ver artículo	El sistema permite al usuario ver el detalle del artículo seleccionado.
CU-07	Votar artículo	El sistema permite al usuario votar positiva o negativamente un artículo y visualizar en tiempo real la repercusión de su voto.
CU-08	Buscar artículos	El sistema permite al usuario buscar artículos por temáticas y ordenarlos por puntuación o fecha de publicación.
CU-09	Ver perfil	El sistema permite al usuario visualizar su perfil y el de otros usuarios.
CU-10	Ver artículos publicados	El sistema permite al usuario visualizar el listado de artículos publicados de cualquier usuario.
CU-11	Ver actividad	El sistema permite al usuario visualizar la actividad de cualquier usuario.
CU-12	Editar perfil	El sistema permite al usuario editar información de su perfil.
CU-13	Eliminar artículos	El sistema permite al usuario eliminar sus artículos publicados.
CU-14	Consultar privilegios	El sistema permite al usuario consultar sus privilegios de voto y publicación.

CU-15	Eliminar cuenta	El sistema debe permitir al usuario la eliminación de su cuenta.
CU-16	Mantener artículos	El sistema debe permitir al usuario mantener sus artículos en la aplicación si así lo desea.

Tabla 32 - Casos de Uso

Para visualizar una comparativa en tres las funcionalidades especificadas por los casos de uso y los requisitos de usuario, se muestra a continuación una tabla con sus relaciones:

ID-RU	Requisito de Usuario	Caso de Uso	ID-CU
RU-01	El usuario debe poder registrarse en la plataforma.	Registrarse en la aplicación	CU-01
RU-02	El usuario debe poder iniciar sesión en la plataforma.	Iniciar sesión en la aplicación	CU-02
RU-03	El usuario debe poder cerrar sesión en la aplicación una vez iniciada sesión.	Cerrar sesión	CU-03
RU-04	El usuario debe poder publicar artículos en formato PDF con título, descripción, temática y fuente.	Publicar artículo	CU-04
RU-05	El usuario debe poder ver un <i>feed</i> con los artículos y temáticas recomendados por la aplicación.	Ver <i>feed</i>	CU-05
RU-06	El usuario debe poder visualizar el detalle de los artículos publicados.	Ver artículo	CU-06
RU-07	El usuario debe poder votar positiva y negativamente artículos	Votar artículo	CU-07
RU-18	El usuario debe poder ver en tiempo real las actualizaciones en los indicadores al votar un artículo.		
RU-08	El usuario debe poder realizar una búsqueda de artículos por temática.	Buscar artículos	CU-08

RU-09	El usuario debe poder ordenar la búsqueda de artículos por fecha y por veracidad.		
RU-12	El usuario debe poder editar la imagen, nombre, apellidos y acreditaciones de su perfil.	Editar perfil	CU-12
RU-14	El usuario debe poder eliminar sus artículos publicados en su perfil.	Eliminar artículos	CU-13
RU-10	El usuario debe poder ver su información detallada en su perfil.	Ver perfil	CU-09
RU-11	El usuario debe poder ver la información básica de otros usuarios en sus respectivos perfiles.		
RU-13	El usuario debe poder ver el listado de sus artículos publicados.	Ver artículos publicados	CU-10
RU-15	El usuario debe poder ver el listado de artículos publicados por otros usuarios en sus respectivos perfiles.		
RU-16	El usuario debe poder ver su actividad en su perfil.	Ver actividad	CU-11
RU-17	El usuario debe poder ver la actividad de otros usuarios en sus respectivos perfiles.		
RU-19	El usuario debe poder consultar en su perfil los privilegios de votos y publicaciones de los que dispone.	Consultar privilegios	CU-14
RU-20	El usuario debe poder eliminar su cuenta.	Eliminar cuenta	CU-15
RU-21	El usuario debe poder eliminar su cuenta manteniendo sus artículos en la aplicación.	Mantener artículos	CU-16

Tabla 33 - Relación entre Casos de Uso y Requisitos de Usuario

4.2.2 Diagrama de Casos de Uso

En la siguiente sección se muestra el diagrama de casos de uso.

4.2.3 Especificación de Casos de Uso



Ilustración 13 -Diagrama de Casos de Uso

En la siguiente sección se detallan las especificaciones de los casos de uso identificados en la sección anterior.

CU-01	Registrarse en la aplicación
Versión	1.0
Autor	Mario Olmos Manso
Actor	Usuario
Requisito asociado	RU-01
Precondición	El usuario no debe tener una cuenta existente en la aplicación.
Descripción	El sistema permite al usuario registrarse en la aplicación.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de registrarse. 2. El usuario rellena los datos del formulario 3. El sistema valida los datos introducidos. 4. Si la validación es exitosa, el sistema crea una cuenta de usuario.
Postcondiciones	El sistema redirige al usuario al formulario de login.
Excepciones	<p>3.1 El usuario introduce datos con formato erróneo en el formulario.</p> <ul style="list-style-type: none"> • El campo que sea incorrecto mostrará un mensaje de error hasta que se introduzca correctamente. <p>4.1 El usuario es menor de edad.</p> <ul style="list-style-type: none"> • El sistema mostrará al usuario un mensaje de error indicando que debe ser mayor de edad para registrarse. <p>4.2 El usuario introduce un correo electrónico ya existente en la aplicación.</p> <ul style="list-style-type: none"> • El sistema mostrará un mensaje de error indicando que el correo introducido ya está en uso por otro usuario en la aplicación.
Importancia	Alta

Tabla 34 - Especificación CU-01

CU-02	Iniciar Sesión
Versión	1.0
Autor	Mario Olmos Manso

Actor	Usuario
Requisito asociado	RU-02
Precondición	El usuario debe tener una cuenta registrada en la aplicación.
Descripción	El sistema permite al usuario iniciar sesión en la aplicación.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de iniciar sesión 2. El usuario ingresa sus credenciales. 3. El sistema valida las credenciales. 4. Si son validadas, el sistema establece la cookie de sesión permitiendo el acceso a la cuenta.
Postcondiciones	El sistema redirige al usuario a la página de home.
Excepciones	3.1 El usuario introduce unas credenciales incorrectas. <ul style="list-style-type: none"> • Se mostrará un mensaje de error dando la opción de volver a introducir las credenciales.
Importancia	Alta

Tabla 35 - Especificación CU-02

CU-03	Cerrar Sesión
Versión	1.0
Autor	Mario Olmos Manso
Actor	Usuario
Requisito asociado	RU-03
Precondición	El usuario debe estar autenticado en la aplicación.
Descripción	El sistema permite al usuario cerrar la sesión una vez la haya iniciado.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de cerrar sesión. 2. El sistema termina la sesión del usuario.
Postcondiciones	El sistema redirige al usuario a la página de inicio de sesión.
Excepciones	N/A
Importancia	Alta

Tabla 36 - Especificación CU-03

CU-04	Publicar Artículo
Versión	1.0
Autor	Mario Olmos Manso
Actor	Usuario
Requisito asociado	RU-04
Precondición	El usuario debe estar autenticado en la aplicación.
Descripción	El sistema permite al usuario publicar un artículo.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario marca la opción de subir artículo. 2. El sistema le redirige al formulario para publicar un artículo. 3. El usuario completa los datos del formulario y lo envía. 4. El sistema valida los datos introducidos y los privilegios del usuario. 5. El sistema da de alta el artículo.
Postcondiciones	El sistema limpia el formulario e indica al usuario que su artículo ha sido dado de alta correctamente.
Excepciones	<p>3.1 El usuario introduce datos con formato erróneo en el formulario.</p> <ul style="list-style-type: none"> • El campo que sea incorrecto mostrará un mensaje de error hasta que se introduzca correctamente. <p>4.1 El usuario no dispone del privilegio para publicar un artículo.</p> <ul style="list-style-type: none"> • Se muestra un mensaje de error al usuario indicándole que no tiene privilegios suficientes. <p>4.2 El usuario ha superado su límite de publicaciones mensuales.</p> <ul style="list-style-type: none"> • Se muestra un mensaje de error al usuario indicándole que ha superado su número máximo de publicaciones mensuales.
Importancia	Alta

Tabla 37 - Especificación CU-04

CU-05	Ver Feed
Versión	1.0
Autor	Mario Olmos Manso
Actor	Usuario

Requisito asociado	RU-05
Precondición	El usuario debe estar autenticado en la aplicación.
Descripción	El sistema permite al usuario ver el <i>feed</i> con los artículos y temáticas recomendadas,
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de home. 2. El sistema muestra una selección de temáticas y artículos recomendados.
Postcondiciones	N/A
Excepciones	N/A
Importancia	Alta

Tabla 38 - Especificación CU-05

CU-06	Ver Artículo
Versión	1.0
Autor	Mario Olmos Manso
Actor	Usuario
Requisito asociado	RU-06
Precondición	El usuario debe estar autenticado en la aplicación.
Descripción	El sistema permite al usuario ver el detalle del artículo seleccionado.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario elige un artículo de los mostrados en página home, explorador, actividad o artículos del autor. 2. El usuario pulsa el botón ver artículo. 3. El sistema muestra la página de detalle del artículo.
Postcondiciones	N/A
Excepciones	N/A
Importancia	Alta

Tabla 39 - Especificación CU-06

CU-07	Votar Artículo
Versión	1.0
Autor	Mario Olmos Manso
Actor	Usuario
Requisito asociado	RU-07, RU-18
Precondición	El usuario debe estar autenticado en la aplicación.
Descripción	El sistema permite al usuario votar positiva o negativamente un artículo y visualizar en tiempo real la repercusión de su voto.
Flujo Normal	1. El usuario vota positiva o negativamente un artículo.
Postcondiciones	El valor del voto, así como su repercusión en los distintos indicadores de calidad será actualizado en tiempo real.
Excepciones	<p>1.1 El usuario no dispone del privilegio para votar un artículo.</p> <ul style="list-style-type: none"> Se muestra un mensaje de error al usuario indicándole que no tiene privilegios suficientes. <p>1.2 El usuario ha superado su límite de votaciones semanales.</p> <ul style="list-style-type: none"> Se muestra un mensaje de error al usuario indicándole que ha alcanzado el límite de votaciones semanales.
Importancia	Alta

Tabla 40 - Especificación CU-07

CU-08	Buscar Artículos
Versión	1.0
Autor	Mario Olmos Manso
Actor	Usuario
Requisito asociado	RU-08, RU-09
Precondición	El usuario debe estar autenticado en la aplicación.
Descripción	El sistema permite al usuario buscar artículos por temáticas y ordenarlos por puntuación o fecha de publicación.
Flujo Normal	<p>1. El usuario selecciona la página explorador.</p> <p>2. El usuario establece los filtros para los artículos.</p> <p>3. El usuario pulsa el botón buscar.</p>

	4. El sistema muestra la lista de artículos filtrados y ordenados.
Postcondiciones	N/A
Excepciones	<p>1.1 El usuario selecciona una temática recomendada en la página de <i>feed</i>.</p> <ul style="list-style-type: none"> El sistema redirigirá al usuario a la página explorador con la temática precargada en el filtro de temática. <p>2.1 El usuario establece filtros de ordenación.</p> <ul style="list-style-type: none"> El sistema solo permitirá filtrar por puntuación o por fecha, pero no por ambos. <p>4.1 El usuario establece una temática sin artículos asociados.</p> <ul style="list-style-type: none"> El sistema mostrará un mensaje indicando que la búsqueda no obtuvo resultados.
Importancia	Media

Tabla 41 - Especificación CU-07

CU-09	Ver Perfil
Versión	1.0
Autor	Mario Olmos Manso
Actor	Usuario
Requisito asociado	RU-10, RU-11
Precondición	El usuario debe estar autenticado en la aplicación.
Descripción	El sistema permite al usuario visualizar su perfil y el de otros usuarios.
Flujo Normal	<ol style="list-style-type: none"> El usuario selecciona la opción de ver perfil, o visualizando otro artículo selecciona la opción de ver perfil de su autor. El sistema carga el perfil propio o el del usuario seleccionado.
Postcondiciones	N/A
Excepciones	<p>1.1 El usuario quiere visualizar el perfil de un autor que ha eliminado su cuenta manteniendo sus artículos.</p> <ul style="list-style-type: none"> El sistema mostrará un mensaje en rojo en lugar del botón para ver el perfil del autor indicando que la cuenta del autor ha sido eliminada. <p>2.1 El usuario quiere visualizar un perfil ajeno.</p>

	<ul style="list-style-type: none"> El sistema mostrará información reducida del usuario visitado, así como bloquea las acciones exclusivas del propietario del perfil.
Importancia	Alta

Tabla 42 - Especificación CU-09

CU-10	Ver Artículos Publicados
Versión	1.0
Autor	Mario Olmos Manso
Actor	Usuario
Requisito asociado	RU-13, RU-15
Precondición	El usuario debe estar autenticado en la aplicación.
Descripción	El sistema permite al usuario visualizar el listado de artículos publicados de cualquier usuario.
Flujo Normal	<ol style="list-style-type: none"> El usuario selecciona la opción de ver perfil, o visualizando otro artículo selecciona la opción de ver perfil de su autor. El usuario selecciona la pestaña de artículos. El sistema muestra un listado de los artículos publicados por el dueño del perfil.
Postcondiciones	N/A
Excepciones	<p>2.1 El usuario quiere visualizar el listado de artículos de un usuario que todavía no ha publicado.</p> <ul style="list-style-type: none"> El sistema mostrará un mensaje indicando que el usuario visitado aún no tiene artículos publicados.
Importancia	Media

Tabla 43 - Especificación CU-10

CU-11	Ver Actividad
Versión	1.0
Autor	Mario Olmos Manso
Actor	Usuario
Requisito asociado	RU-16, RU-17

Precondición	El usuario debe estar autenticado en la aplicación.
Descripción	El sistema permite al usuario visualizar la actividad de cualquier usuario.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de ver perfil, o visualizando otro artículo selecciona la opción de ver perfil de su autor. 2. El usuario selecciona la pestaña de información. 3. El sistema muestra un listado de los artículos con los que ha interactuado el propietario del perfil.
Postcondiciones	N/A
Excepciones	<ol style="list-style-type: none"> 2.1 El usuario quiere visualizar la actividad de un usuario que todavía no ha votado. <ul style="list-style-type: none"> • El sistema mostrará un mensaje indicando que el usuario visitado aún no ha votado ningún artículo.
Importancia	Media

Tabla 44 - Especificación CU-11

CU-12	Editar Perfil
Versión	1.0
Autor	Mario Olmos Manso
Actor	Usuario
Requisito asociado	RU-12
Precondición	El usuario debe estar autenticado en la aplicación y solo puede editar su propio perfil.
Descripción	El sistema permite al usuario editar información de su perfil.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de ver perfil. 2. El usuario selecciona la pestaña de información. 3. El sistema muestra un formulario con los datos del perfil que se permiten editar. 4. El usuario realiza las ediciones que desee. 5. El sistema valida los cambios realizados. 6. El sistema actualiza la información de usuario.
Postcondiciones	El sistema muestra el perfil del usuario con la información actualizada.
Excepciones	<ol style="list-style-type: none"> 4.1 El usuario introduce datos con formato erróneo en el formulario. <ul style="list-style-type: none"> • El campo que sea incorrecto mostrará un mensaje de error hasta que se introduzca correctamente.

	<p>4.2 El usuario intenta editar la fecha de nacimiento por una incorrecta.</p> <ul style="list-style-type: none"> El sistema mostrará al usuario un mensaje de error indicando que debe ser mayor de edad para utilizar la aplicación.
Importancia	Media

Tabla 45 - Especificación CU-12

CU-13	Eliminar Artículos
Versión	1.0
Autor	Mario Olmos Manso
Actor	Usuario
Requisito asociado	RU-14
Precondición	El usuario debe estar autenticado en la aplicación y haber publicado al menos un artículo.
Descripción	El sistema permite al usuario eliminar sus artículos publicados.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de ver perfil. 2. El usuario selecciona la pestaña de artículos. 3. El sistema muestra un listado de los artículos publicados por el usuario. 4. El usuario presiona el botón de eliminar de uno de sus artículos.
Postcondiciones	El sistema muestra el perfil del usuario con la información actualizada.
Excepciones	N/A
Importancia	Media

Tabla 46 - Especificación CU-13

CU-14	Consultar Privilegios
Versión	1.0
Autor	Mario Olmos Manso
Actor	Usuario

Requisito asociado	RU-19
Precondición	El usuario debe estar autenticado en la aplicación.
Descripción	El sistema permite al usuario consultar sus privilegios de voto y publicación.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de ver perfil. 2. El usuario selecciona la ruleta de acciones y la opción de consultar privilegios. 3. El sistema muestra los privilegios del usuario.
Postcondiciones	N/A
Excepciones	N/A
Importancia	Baja

Tabla 47 - Especificación CU-14

CU-15	Eliminar Cuenta
Versión	1.0
Autor	Mario Olmos Manso
Actor	Usuario
Requisito asociado	RU-20
Precondición	El usuario debe estar autenticado en la aplicación.
Descripción	El sistema debe permitir al usuario la eliminación de su cuenta.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de ver perfil. 2. El usuario selecciona la ruleta de acciones y la opción de eliminar cuenta. 3. El usuario decide entre mantener o eliminar sus artículos.
Postcondiciones	N/A
Excepciones	N/A
Importancia	Alta

Tabla 48 - Especificación CU-15

CU-16	Eliminar Cuenta
Versión	1.0
Autor	Mario Olmos Manso
Actor	Usuario
Requisito asociado	RU-21
Precondición	El usuario debe estar autenticado en la aplicación y haber publicado artículos.
Descripción	El sistema debe permitir al usuario mantener sus artículos en la aplicación si así lo desea.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de ver perfil. 2. El usuario selecciona la ruleta de acciones y la opción de eliminar cuenta. 3. El usuario decide seleccionar la opción de mantener artículos.
Postcondiciones	Los artículos que quedan en la aplicación contendrán información del autor.
Excepciones	N/A
Importancia	Media

Tabla 49 - Especificación CU-16

4.3 Requisitos funcionales

Los requisitos funcionales describen el comportamiento y características que debe tener el sistema para cumplir con los requisitos de usuario, es decir, qué funcionalidades debe ofrecer el sistema y su comportamiento ante determinados eventos o acciones del usuario.

En esta sección se detallan los requisitos funcionales analizados en FACTOR:

RU-ID	RF-ID	Descripción del Requisito Funcional
RU-01	RF-01	El sistema solicitará al usuario los datos necesarios en el formulario de registro para crear una cuenta de usuario.
	RF-02	El sistema comprobará que el correo electrónico introducido es único en la aplicación.
	RF-03	El sistema comprobará a partir de la fecha de nacimiento que el usuario es mayor de dieciocho años.

RU-02	RF-04	El sistema solicitará al usuario su email y contraseña para iniciar sesión en la aplicación.
	RF-05	El sistema verificará las credenciales y los volverá a solicitar en caso de ser erróneos.
	RF-06	De ser las credenciales correctas el sistema establecerá la <i>cookie</i> de sesión y redirigirá al usuario autenticado a la aplicación.
RU-03	RF-07	El sistema permitirá al usuario autenticado terminar su sesión actual en cualquier momento mediante un botón en la barra de navegación y un botón en los ajustes del perfil.
RU-04	RF-08	El sistema solicitará al usuario los datos necesarios en el formulario de alta de artículo para publicar un nuevo artículo.
	RF-09	El sistema verificará la obligatoriedad y tamaño de los datos introducidos.
	RF-10	El sistema comprobará que el archivo introducido es en formato pdf.
	RF-11	El sistema verificará que el usuario tiene privilegios para publicar.
	RF-12	El sistema verificará que el usuario no haya sobrepasado su límite de publicaciones mensuales.
RU-05	RF-13	El sistema mostrará al usuario un <i>feed</i> con las temáticas donde se han publicado artículos recientemente.
	RF-14	El sistema mostrará al usuario un <i>feed</i> con los artículos recomendados en base a un valor de peso calculado por el algoritmo.
	RF-15	El sistema permitirá al usuario seleccionar una temática del <i>feed</i> y redirigirle a la página del buscador con la temática como parámetro de filtro.
RU-06	RF-16	El sistema permitirá visualizar el detalle de un artículo en cualquier página en la que se liste.
	RF-17	El sistema cargará en cada artículo todos los indicadores de calidad asociados.
	RF-18	El sistema cargará la rama jerárquica de temáticas a la que pertenece el artículo.
	RF-19	El sistema cargará el número de votos positivos y negativos del artículo, así como el voto del usuario en caso de existir.
	RF-20	El sistema mostrará la información básica del autor y un botón para visualizar su perfil completo.
	RF-21	El sistema mostrará un enlace a la fuente del artículo.
	RF-22	El sistema permitirá establecer en pantalla completa el visor de pdf.
RU-07	RF-23	El sistema permitirá votar positiva y negativamente artículos mediante dos iconos situados en cada artículo.
	RF-24	El sistema calculará el peso del voto en la puntuación del artículo en base a la reputación del votante.

	RF-25	El sistema recompensará o castigará a los usuarios votantes cuando se categorice un artículo con el que han interactuado
	RF-26	El sistema recompensará o castigará al autor de un artículo cuando este se categorice.
RU-08	RF-27	El sistema permitirá realizar una búsqueda de artículos filtrando por temática mediante un desplegable en la barra de filtrado.
	RF-28	El sistema permitirá seleccionar temáticas de cualquier nivel, pudiendo buscar temáticas genéricas y específicas.
RU-09	RF-29	El sistema permitirá ordenar los artículos filtrados por puntuación ascendente o descendente.
	RF-30	El sistema permitirá ordenar los artículos filtrados por fecha de publicación ascendente o descendente.
	RF-31	El sistema solo permitirá seleccionar un parámetro de ordenación, deshabilitando el otro parámetro.
RU-10	RF-32	El sistema permitirá visualizar directamente el perfil personal del usuario mediante un botón en la barra de navegación.
	RF-33	El sistema permitirá al usuario ver la información detallada en su propio perfil.
	RF-34	El sistema habilitará funciones adicionales en caso en el perfil personal tales como la edición de información, menú de acciones y botón de eliminar artículos.
RU-11	RF-35	El sistema permitirá visualizar el perfil de otros usuarios a través de sus artículos publicados.
	RF-36	El sistema mostrará información básica del perfil de otros usuarios.
RU-12	RF-37	El sistema permitirá al usuario modificar su imagen de perfil, nombre, apellidos y fecha de nacimiento.
	RF-38	El sistema permitirá al usuario añadir, editar y eliminar una o varias acreditaciones.
	RF-39	El sistema validará que los campos editados no son nulos antes de guardarlos.
	RF-40	El sistema validará en caso de modificarse que la nueva fecha de nacimiento no implique un usuario menor de edad.
	RF-41	El sistema permitirá guardar o cancelar los cambios realizados en la pestaña de edición.
RU-13	RF-42	El sistema permitirá al usuario visualizar en su perfil el listado de sus artículos publicados.
RU-14	RF-43	El sistema permitirá al usuario eliminar sus artículos publicados mediante un botón de eliminar.
	RF-44	El sistema solicitará al usuario una confirmación a través de una ventana emergente para eliminar un artículo.
RU-15	RF-45	El sistema permitirá visualizar en el perfil de otro usuario el listado de sus artículos publicados en la pestaña artículos.

RU-16	RF-46	El sistema permitirá visualizar en el perfil personal, en la pestaña de información, un listado de sus artículos votados.
RU-17	RF-47	El sistema permitirá visualizar en el perfil de otro usuario, en la pestaña información, el listado de los artículos que ha votado.
RU-18	RF-48	El sistema mostrará al usuario en tiempo real la actualización del icono de votación cuando vote.
	RF-49	El sistema mostrará al usuario en tiempo real la actualización del contador de votos cuando vote.
	RF-50	El sistema mostrará al usuario en tiempo real la actualización de la puntuación del artículo cuando vote.
RU-19	RF-51	El sistema permitirá visualizar en el perfil personal los límites de publicaciones mensuales y el límite de votos semanales de los que dispone el usuario.
	RF-52	El sistema permitirá visualizar en el perfil personal las publicaciones realizadas por el usuario el último mes y las votaciones realizadas la última semana.
RU-20	RF-53	El sistema permitirá al usuario mediante un botón en la barra de acciones del perfil personal eliminar su cuenta.
RU-21	RF-54	El sistema permitirá al usuario mediante una ventana emergente mantener sus artículos en la aplicación antes de eliminar su cuenta.

Tabla 50 - Requisitos de Usuario

4.3.1 Requisitos de información

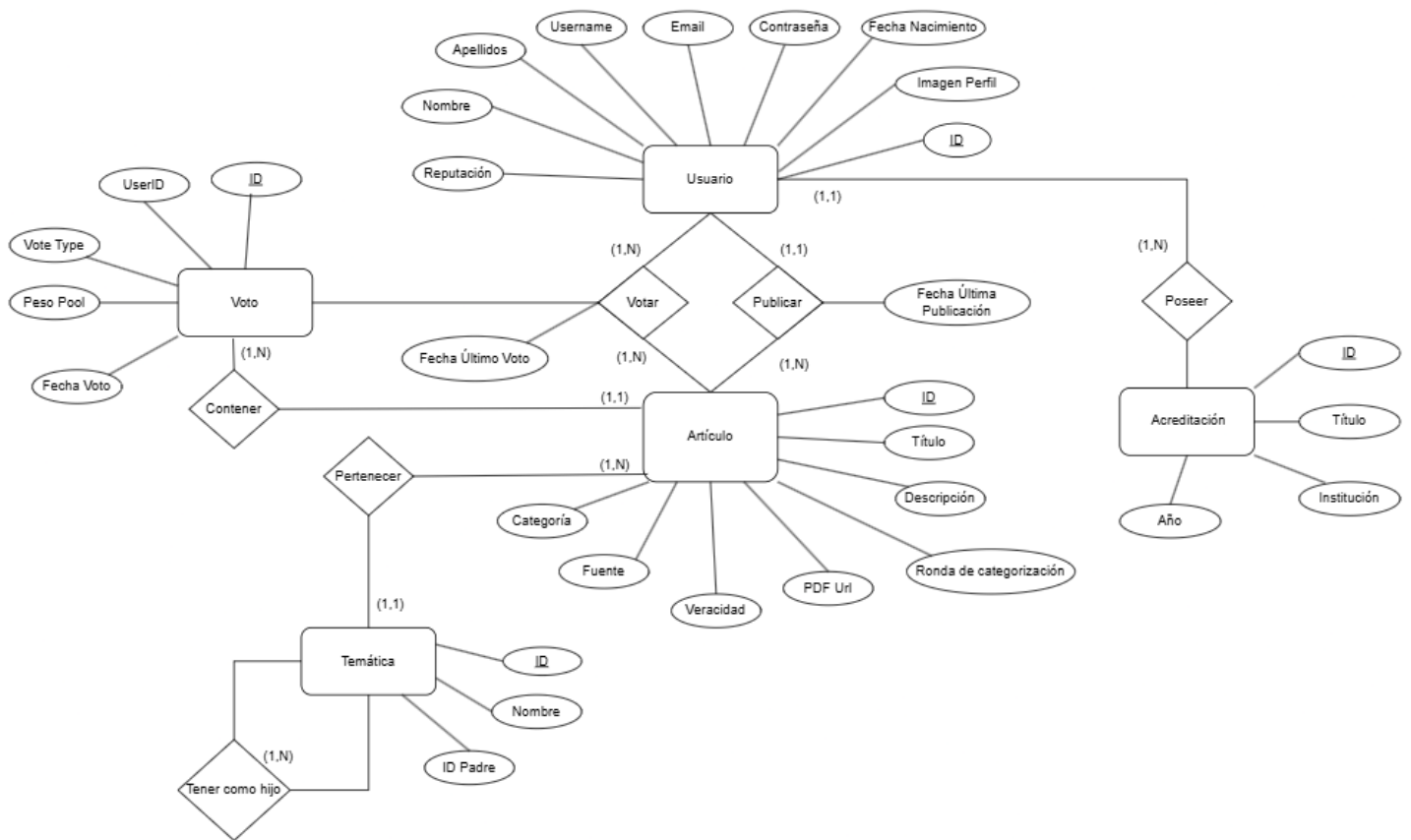
Esta sección detalla los requisitos de almacenamiento de información identificados para la aplicación. Estos requisitos definen la información a almacenar por el sistema para satisfacer las funcionalidades descritas en los casos de uso del sistema y los requisitos funcionales.

ID	Descripción
RI-01	Los datos requeridos en el formulario de registro son el nombre, apellidos, correo electrónico, fecha de nacimiento y contraseña.
RI-02	Cuando un usuario realice una votación se guardará la fecha de su última votación.
RI-03	Cuando un usuario realice una publicación se guardará la fecha de su última publicación.
RI-04	Los datos requeridos para añadir una acreditación son título, institución y año de la titulación.

RI-05	En la edición del perfil el usuario podrá modificar su nombre, apellidos, fecha de nacimiento, imagen de perfil y acreditaciones.
RI-06	El sistema guardará de cada artículo su id, título, descripción, id del autor, temática, puntuación de veracidad, fuente, fecha de creación, array de votos, categoría evaluada, ronda de categorización, información del autor y booleano de borrado.
RI-07	Cada artículo guardará información del autor en su campo “ <i>authorInfo</i> ” en el caso de que el autor elimine su cuenta manteniendo sus artículos, guardando su nombre, apellidos, email, reputación, fecha de nacimiento y acreditaciones.
RI-08	Cada voto del array de votos contendrá información del usuario que ha realizado el voto, el tipo de voto, el peso en el “pool” y la fecha de votación.
RI-09	El sistema guardará de cada temática su nombre, su temática padre y el array de sus subtemáticas.
RI-10	El sistema restringirá el tamaño de los pdf a un máximo de veinte MB y su formato a pdf.
RI-11	El sistema restringirá el tamaño de las imágenes de perfil a un máximo de 5MB.
RI-12	El sistema debe eliminar el archivo pdf cuando un usuario elimina el artículo asociado.
RI-13	El sistema debe eliminar todos los archivos asociados a los artículos de un usuario cuando este elija eliminar su cuenta y sus artículos.
RI-14	El sistema debe eliminar la foto de perfil del usuario en caso de que exista si el usuario actualiza su foto.

Tabla 51 - Requisitos de Información

Para modelar los requisitos de información identificados previamente, se ha diseñado el siguiente Modelo Entidad-Relación (M-ER), el cual refleja las entidades, atributos y relaciones necesarias para el sistema.



A partir este modelo, se desarrollará un diccionario de datos en el capítulo de diseño. Este diccionario detalla las estructuras específicas de cada entidad, proporcionando una base sólida para la implementación técnica del sistema

4.4 Requisitos no funcionales

Los requisitos no funcionales describen el comportamiento que debe tener el sistema en términos de rendimiento, seguridad, accesibilidad, escalabilidad, fiabilidad y demás aspectos de calidad. No describen funcionalidades si no cómo debe comportarse el sistema para garantizar un correcto funcionamiento técnico y una buena experiencia de usuario.

ID	Tipo	Descripción
RNF-01	Rendimiento	El sistema debe tener un tiempo de respuesta para cargar cualquier página de la aplicación inferior a 4 segundos.
RNF-02	Rendimiento	El sistema debe poder manejar como mínimo 500 solicitudes concurrentes sin que el rendimiento se vea afectado.
RNF-03	Rendimiento	Los artículos cargados en el <i>feed</i> se cargarán progresivamente de 10 en 10 para mejorar el tiempo de carga y la experiencia de usuario.
RNF-04	Seguridad	El sistema debe encriptar la contraseña de los usuarios mediante encriptado usando el algoritmo SHA-256.
RNF-05	Seguridad	El sistema implementara autenticación y autorización mediante una cookie basada en JWT para todas las operaciones que requieran de estos privilegios.
RNF-06	Seguridad	El sistema devolverá únicamente la información necesaria evitando la exposición de datos sensibles y de los id de los usuarios utilizados por la base de datos.
RNF-07	Seguridad	El sistema configurará para la cookie el campo httpOnly como true evitando que el contenido de la cookie sea accesible desde JavaScript en el navegador y ataques Cross-Site Scripting.
RNF-08	Usabilidad	La interfaz de usuario deberá ser intuitiva y fácil de utilizar, pudiendo realizar todas las funcionalidades en 3 clicks como máximo.
RNF-09	Usabilidad	El sistema deberá proporcionar mensajes de éxito y error claros y útiles.
RNF-10	Escalabilidad	El sistema debe ser escalable horizontalmente para añadir nuevas temáticas y verticalmente para añadir nuevos niveles de subtemáticas.
RNF-11	Escalabilidad	El sistema debe poder soportar un aumento en el volumen de datos y en el flujo de usuarios sin realizar cambios en la arquitectura.
RNF-12	Mantenibilidad	El código del sistema debe estar correctamente estructurado y comentado para su mantenimiento.
RNF-13	Mantenibilidad	Los parámetros del algoritmo de veracidad, así como otros parámetros de la aplicación deben ser fácilmente modificables para responder a un crecimiento de usuarios.
RNF-14	Fiabilidad	Si durante la publicación de un artículo el sistema falla, el sistema debe revertir todos los cambios realizados para evitar datos inconsistentes.
RNF-15	Compatibilidad	El sistema debe ser compatible con los navegadores más utilizados.

Tabla 52 - Requisitos no Funcionales

Capítulo 5: Diseño

5.1 Arquitectura lógica

La arquitectura lógica define la estructura de los diferentes componentes de software y la forma en que interactúan entre sí.

El sistema se basa en la pila MEAN (*MongoDB*, *Express*, *Angular* y *Node.js*), lo que permite una integración fluida y eficiente, ya que utiliza JavaScript en todas sus capas, tanto en el *frontend* como en el *backend*.

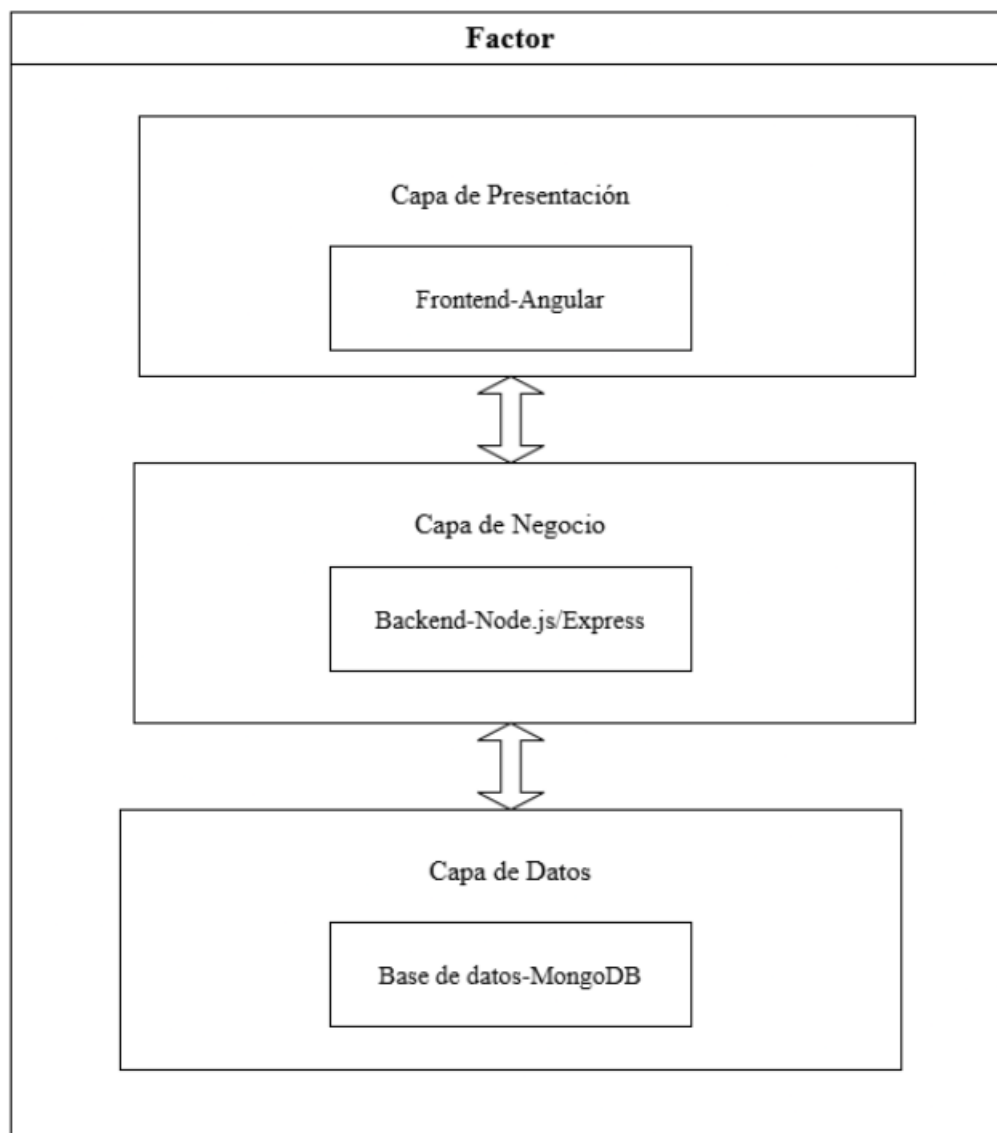


Ilustración 14 - Diagrama de la Arquitectura Lógica

- Capa de presentación:

En la primera capa se encuentran todas las interfaces de la aplicación, son el conjunto de las pantallas con las que interactúa el usuario.

En esta capa también se implementa lógica de enrutamiento, las validaciones de formularios anteriores a enviar los datos al *backend* y se consume las APIs del *backend* a través de peticiones HTTP.

Esta capa utiliza como tecnologías *Angular*, *TypeScript*, *HTML* y *CSS*.

- Capa de Negocio:

Esta capa gestiona todas las reglas de negocio y operaciones de la aplicación.

En la capa de negocio se encuentran todas las rutas y controladores de la API REST, implementando las diferentes operaciones y algoritmos que afectan a los usuarios y a los artículos de la aplicación, además de las operaciones de autenticación y seguridad mediante el *middleware* de JWT y los *middlewares* para la subida de archivos pdf a través de *multer* y de imágenes.

Esta capa utiliza como tecnologías *Node.js* y su *framework Express.js*.

- Capa de Datos:

Esta capa almacena y gestiona los datos de la aplicación.

Almacena todos los artículos, usuarios y temáticas presentes en FACTOR.

Esta capa utiliza *MongoDB* con MONGOOSE

5.2 Arquitectura física

La arquitectura física especifica como se distribuyen los componentes *software* del sistema sobre la infraestructura *hardware* y las redes.

FACTOR se ha llevado a cabo en un entorno de desarrollo local, con la posibilidad de ser desplegado en un futuro en un servidor en la nube para su acceso público. Su arquitectura se basa en una infraestructura común para las aplicaciones web modernas, donde actualmente los distintos componentes (*frontend* y *backend*) se encuentran en una misma máquina a excepción de la base de datos, siendo lo ideal en un futuro alojar estos componentes en uno o varios servidores web.

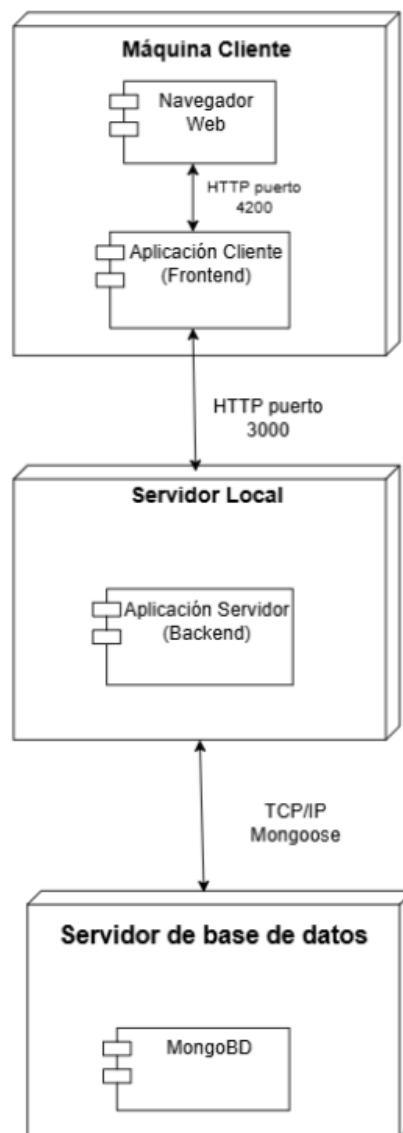


Ilustración 15 - Diagrama de la Arquitectura Física

5.3 Modelo de datos

En la siguiente sección se especifica el modelo de datos diseñado, que define como se estructuran, almacenan y gestionan los datos dentro de la aplicación FACTOR.

Dado que se ha utilizado *MongoDB* como base de datos, el modelo sigue un enfoque NoSQL donde la información se organiza en colecciones y documentos JSON.

En esta sección, se presenta el diagrama de entidad-relación (ERD) del sistema y una explicación de las colecciones principales, detallando los atributos y relaciones entre los diferentes tipos de datos.

5.3.1 Modelo lógico

En esta sección, se presenta el modelo lógico de la base de datos, donde se detallan las colecciones y documentos, sus atributos clave, y las relaciones entre ellas.

USUARIOS (*ObjectId*, Nombre, Apellidos, *Username*, Email, Contraseña, Fecha_Nacimiento, Reputación, Fecha_Última_Publicación, Fecha_Último_voto, Acreditaciones [], Imagen_Perfil)

Acreditaciones []: ACREDITACIÓN (*ObjectId*)

ACREDITACIÓN (*ObjectId*, Título, Institución, Año)

Ilustración 16 - Diagrama Entidad-Relación

TEMA (*ObjectId*, Nombre, ID_Padre, Subtemas [])

Subtemas []: TEMA (*ObjectId*)

VOTO (*ObjectId*, UserId, VoteType, Peso_Pool, Fecha_Voto)

UserId: USUARIOS (*ObjectId*)

ARTÍCULO (ObjectId, Título, Descripción, Pdf_Url, Autor, Tema, Veracidad, Fuente, Fecha_Creación, Votos [], Categoría, Ronda_Categorización)

Autor: USUARIOS (*ObjectId*)

Tema: TEMA (*ObjectId*)

Votos []: VOTO (*ObjectId*)

5.3.2 Diccionario de datos

El diccionario de datos describe la estructura de almacenamiento de la información en la base de datos FACTOR, detallando cada entidad, sus atributos, tipos de datos y propósito dentro del sistema.

Si bien en el modelo lógico se representaban las entidades Votos y Acreditaciones como entidades independientes para facilitar la comprensión del diseño, en el modelo físico estas no existen como colecciones separadas. En su lugar:

- Los votos están embebidos dentro de los documentos de la colección "Artículos".
- Las acreditaciones están embebidas dentro de los documentos de la colección "Usuarios".

Dado este enfoque, el diccionario de datos a nivel físico se presenta con las entidades Usuarios, Artículos y Temas, reflejando la estructura real de almacenamiento en *MongoDB*.

Atributo	Tipo de Dato	Descripción	Restricciones
username	String	Nombre de usuario único	Único, No nulo
email	String	Correo electrónico único del usuario	Único, No nulo
password	String (hashed)	Contraseña cifrada del usuario	No nulo (Cifrado)
reputacion	Number	Nivel de reputación del usuario en la plataforma	No nulo (Valor por defecto: 30)

imagenPerfil	String (URL)	URL de la imagen de perfil del usuario	Opcional
nombre	String	Nombre del usuario	No nulo
apellidos	String	Apellidos del usuario	No nulo
fechaNacimiento	Date	Fecha de nacimiento del usuario	No nulo
fechaUltimaPublicacion	Date	Fecha de la última publicación del usuario	Opcional
fechaUltimoVoto	Date	Fecha del último voto realizado por el usuario	Opcional
acreditaciones.title	String	Título de la acreditación	No nulo
acreditaciones.institution	String	Institución que otorgó la acreditación	No nulo
acreditaciones.year	Number	Año en que se obtuvo la acreditación	No nulo

Tabla 53 - Diccionario de Datos - Usuario

Atributo	Tipo de Dato	Descripción	Restricciones
title	String	Título del artículo	No nulo
description	String	Descripción breve del artículo	Opcional
pdfUrl	String (URL)	URL donde se almacena el archivo PDF del artículo	No nulo
veracity	Number	Puntuación de veracidad del artículo	No nulo (Valor por defecto: 0)
theme	ObjectId (Theme)	Referencia a la temática a la que pertenece el artículo	No nulo
source	String	Fuente de donde proviene el artículo	No nulo
createdAt	Date	Fecha en que se creó el artículo	No nulo (Por defecto: Fecha actual)
evaluated	String (Enum: 'verificado', 'neutro', 'pocoFiable')	Estado de evaluación del artículo	Opcional (Valores: 'verificado', 'neutro', 'pocoFiable')

categorizationRounds	Number	Número de rondas de categorización	No nulo (Por defecto: 0)
deleted	Boolean	Indica si el artículo ha sido eliminado	No nulo (Por defecto: false)
votes.user	ObjectId (User)	Usuario que realizó el voto (referencia)	No nulo
votes.voteType	String (Enum: 'upvote', 'downvote')	Tipo de voto: positivo o negativo	No nulo (Valores: 'upvote', 'downvote')
votes.pesoPool	Number	Peso del voto en la reputación del artículo	No nulo
votes.votedAt	Date	Fecha en que se realizó el voto	No nulo
author	ObjectId (User)	Referencia al autor del artículo	No nulo
title	String	Título del artículo	No nulo
description	String	Descripción breve del artículo	Opcional
pdfUrl	String (URL)	URL donde se almacena el archivo PDF del artículo	No nulo
veracity	Number	Puntuación de veracidad del artículo	No nulo (Valor por defecto: 0)
theme	ObjectId (Theme)	Referencia a la temática a la que pertenece el artículo	No nulo
source	String	Fuente de donde proviene el artículo	No nulo
createdAt	Date	Fecha en que se creó el artículo	No nulo (Por defecto: Fecha actual)

Tabla 54 - Diccionario de Datos - Artículo

Atributo	Tipo de Dato	Descripción	Restricciones
name	String	Nombre de la temática	No nulo
parentTheme	ObjectId (Theme)	Referencia al tema padre en caso de ser subcategoría	Opcional (Puede ser <i>null</i> para categorías raíz)
subthemes	Array de ObjectId (Theme)	Lista de subtemas relacionados	Opcional

Tabla 55 - Diccionario de Datos - Temática

5.3.3 Consideraciones adicionales

En la etapa del diseño físico del modelo de datos se han tomado una serie de decisiones para satisfacer las funcionalidades analizadas. A continuación, se especifican estas decisiones.

En primer lugar y como ya se ha mostrado en el diccionario de datos, las entidades de acreditaciones y votos se han embebido en las entidades usuarios y artículos respectivamente. La razón de esta decisión ha sido la eficiencia en las consultas a la base de datos, en el caso de las acreditaciones son inherentes al usuario y no tenían el peso suficiente como para necesitar ser una entidad independiente. De una manera similar cada voto debería estar asociado con un único artículo, aunque ya que en la totalidad de consultas realizadas a la colección de artículos se necesita acceder al *array* de votos se tomó la decisión de definirlo como un atributo más de cada artículo ya que presentaba más ventajas en términos de eficiencia que desventajas en términos de tamaño de documentos.

Otra decisión de diseño importante vino dada por la necesidad de persistir los datos de un autor en caso de que este decida eliminar su cuenta, y quiera mantener sus artículos en la aplicación. Para satisfacer este caso de uso, se diseñó un método el cual en el momento en el que el autor decide eliminar su cuenta el sistema inserta en cada uno de sus artículos un bloque de datos que persistirá en la base de datos. Este bloque de datos consta de la siguiente información:

Atributo	Tipo de Dato	Descripción	Restricciones
authroInfo.nombre	String	Nombre del autor	No nulo
authroInfo.apellidos	String	Apellidos del autor	No nulo
authroInfo.email	String	Email del autor	No nulo
authroInfo.reputacion	Number	Reputación del autor al momento de eliminar la cuenta	No Nulo
authroInfo.fechaNacimiento	Date	Fecha de nacimiento del autor	No nulo
authroInfo.acreditaciones.title	String	Título de la acreditación	No nulo
authroInfo.acreditaciones.institution	String	Institución del título	No nulo
authroInfo.acreditaciones.year	Number	Año de expedición del título	No nulo
deleted	Boolean	Flag de cuenta eliminada	True

Tabla 56 - Bloque de datos authorInfo

En último lugar, un comentario sobre el almacenamiento de los documentos y las imágenes del perfil ya que estos archivos no se guardan en la base datos. Tanto los pdf como las imágenes son almacenadas en directorios separados en el servidor y según el caso, son actualizados o eliminados en paralelo a su referencia asociada en la base de datos.

5.4 Categorización de artículos según su veracidad

En esta sección se van a detallar las cuestiones de diseño más relevantes llevadas a cabo en FACTOR, explicando el funcionamiento de la reputación y la puntuación y el algoritmo que gestiona la categorización de artículos. Por último, se detallará el sistema de *feed*.

5.4.1 Parámetros de veracidad.

Previo a entrar en el algoritmo de categorización es imprescindible entender los parámetros de veracidad asociados a los artículos y a los usuarios, la puntuación y la reputación.

5.4.1.1 Reputación

La reputación es un parámetro asociado a cada usuario y toma valores entre 0 y 100, se trata como un indicador de qué tan buen comportamiento está teniendo en la plataforma y puede disminuir o aumentar en función de las acciones que tome el usuario. La reputación también es utilizada para delimitar las acciones del usuario en el sistema, restringiendo el número de veces que un usuario puede publicar artículos al mes y el número de votos que el usuario puede realizar a la semana.

Reputación	Publicaciones por mes
(70, 100]	Ilimitadas
(50, 70]	4
(30, 50]	2
(15, 30]	1
[0, 15]	0

Tabla 57 - Reglas de publicación

Reputación	Votos por semana
(70, 100]	Ilimitados
(50, 70]	6
(30, 50]	4
(15, 30]	2
[0, 15]	0

Tabla 58 - Reglas de voto

Como se puede observar en las tablas anteriores se disponen de dos listas con las diferentes reglas de publicación y de votación, otorgando a los usuarios con una reputación mayor de 70, publicaciones y votos ilimitados y prohibiendo publicaciones y limitando a un voto semanal a los usuarios con una reputación inferior a 15 puntos.

Por último, la reputación también es utilizada para calcular el peso que tiene el voto de un usuario en otro artículo, como veremos en el siguiente apartado.

5.4.1.2 Puntuación

El parámetro de puntuación es un indicador asociado a cada artículo, toma valores entre 0 y 10 y es un valor numérico trata de evaluar lo que ha gustado un artículo a la comunidad. En primera instancia cuando un autor publica un artículo, este le otorga una puntuación inicial en base a su reputación hasta un máximo de 6 puntos, de esta manera los artículos de autores con mayor reputación serán recompensados con un empuje inicial.

La puntuación de cada artículo variará con cada voto que reciba y en función de la reputación del usuario que lo vote. Así la puntuación asociada a cada voto pondera diferente en función de la reputación, siendo lineal en reputaciones menores, pero viéndose incrementada en mayor medida si la reputación del votante es alta. Para modelar esta relación entre reputación y peso del voto, se exploraron y compararon distintas funciones matemáticas, generando sus gráficas correspondientes mediante la herramienta *GeoGebra Graphing Calculator*. A continuación, se describen las tres opciones evaluadas:

1. Función original (lineal + raíz cuadrada)

Esta versión consistía en un crecimiento lineal desde la reputación 0 hasta 50, partiendo de un peso mínimo de 0.01 y alcanzando 0.3. A partir de ahí, se introducía un crecimiento no lineal mediante una función de raíz cuadrada, acelerando el incremento del peso del voto hasta alcanzar el valor máximo de 1.

Aunque esta opción era sencilla y reflejaba bien la intención de premiar la reputación alta, presentaba un salto brusco en el punto de transición (50–51), lo cual generaba una discontinuidad poco natural y potencialmente confusa.

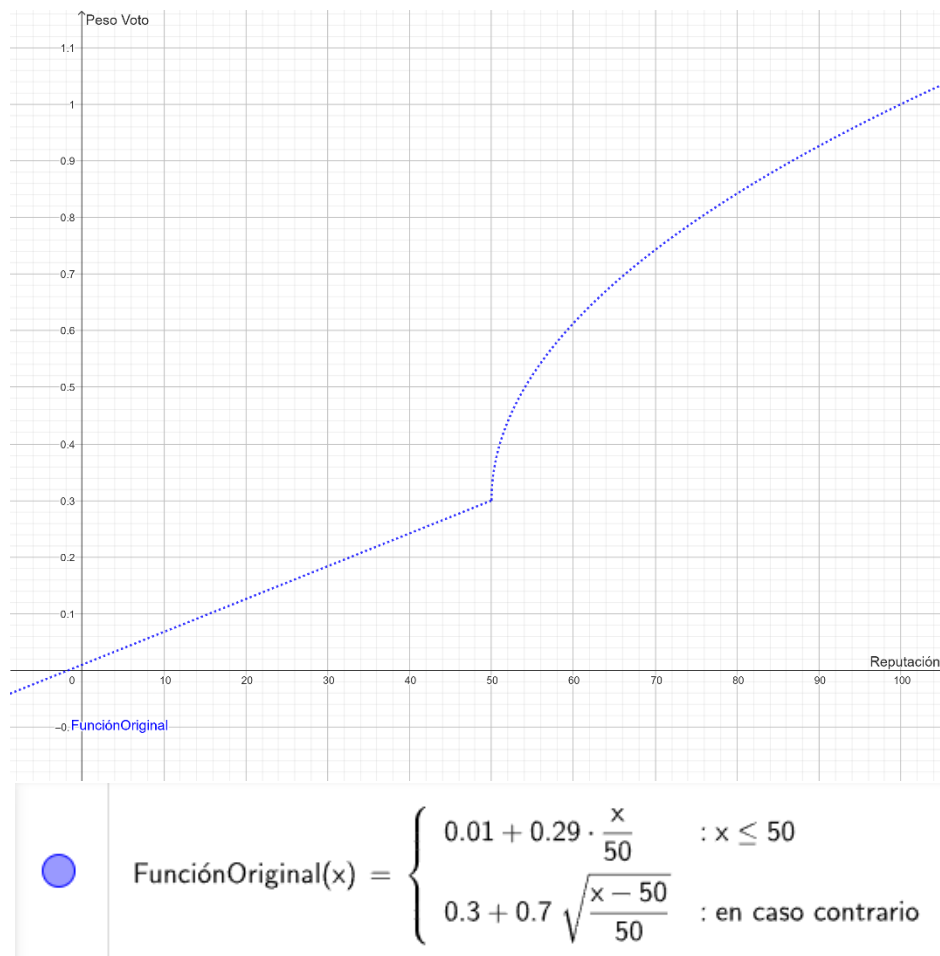


Ilustración 17 - Función Original (Lineal + Raíz Cuadrada)

2. Función suavizada dividiendo por 100

Para atenuar el cambio abrupto, se propuso suavizar la raíz cuadrada dividiendo por 100 en lugar de 50, lo que ralentiza el crecimiento en los primeros puntos después de 50. Esta solución reduce el salto, pero también ralentiza demasiado el crecimiento, lo que restaba valor a los usuarios intermedios y altos.

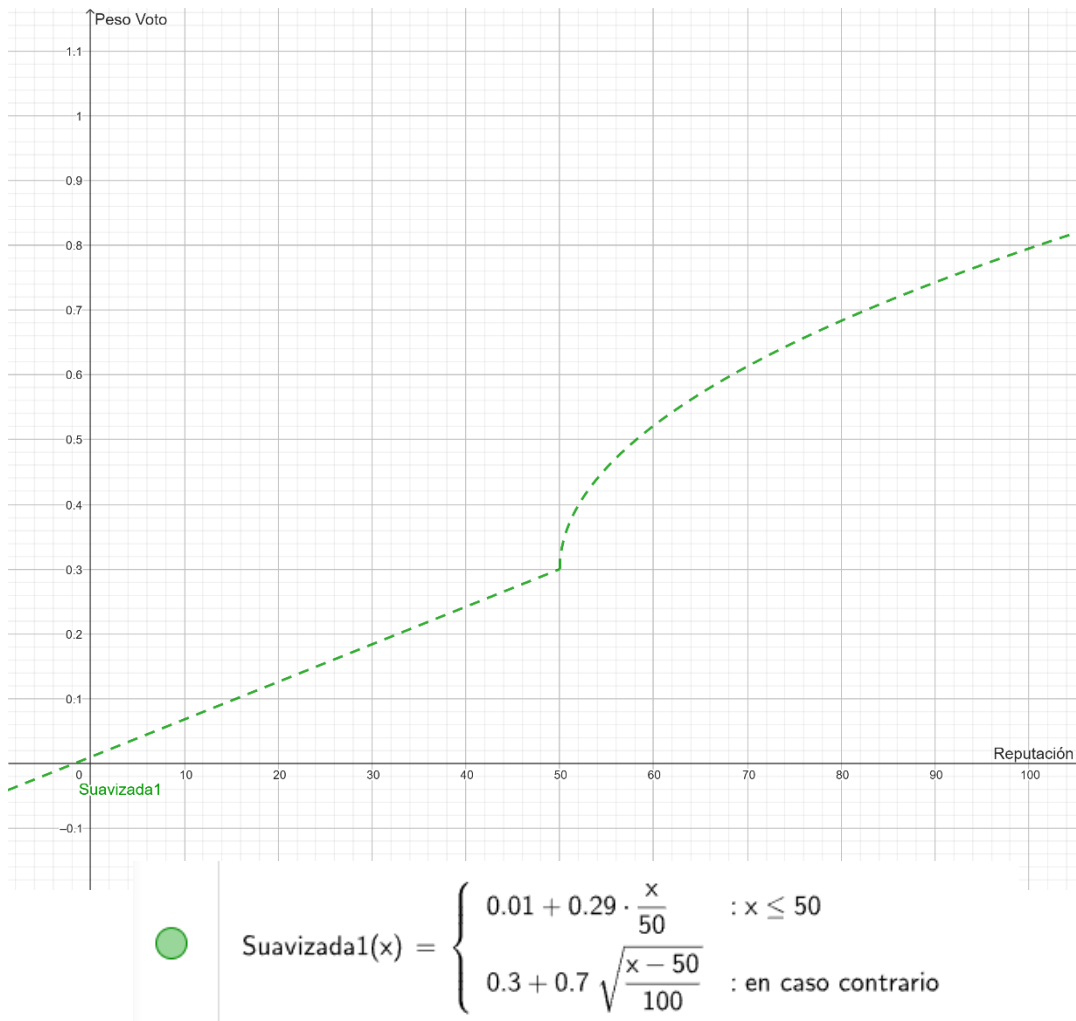


Ilustración 18 - Función Suavizada Dividiendo por 100

3. Función con transición suave (50–70)

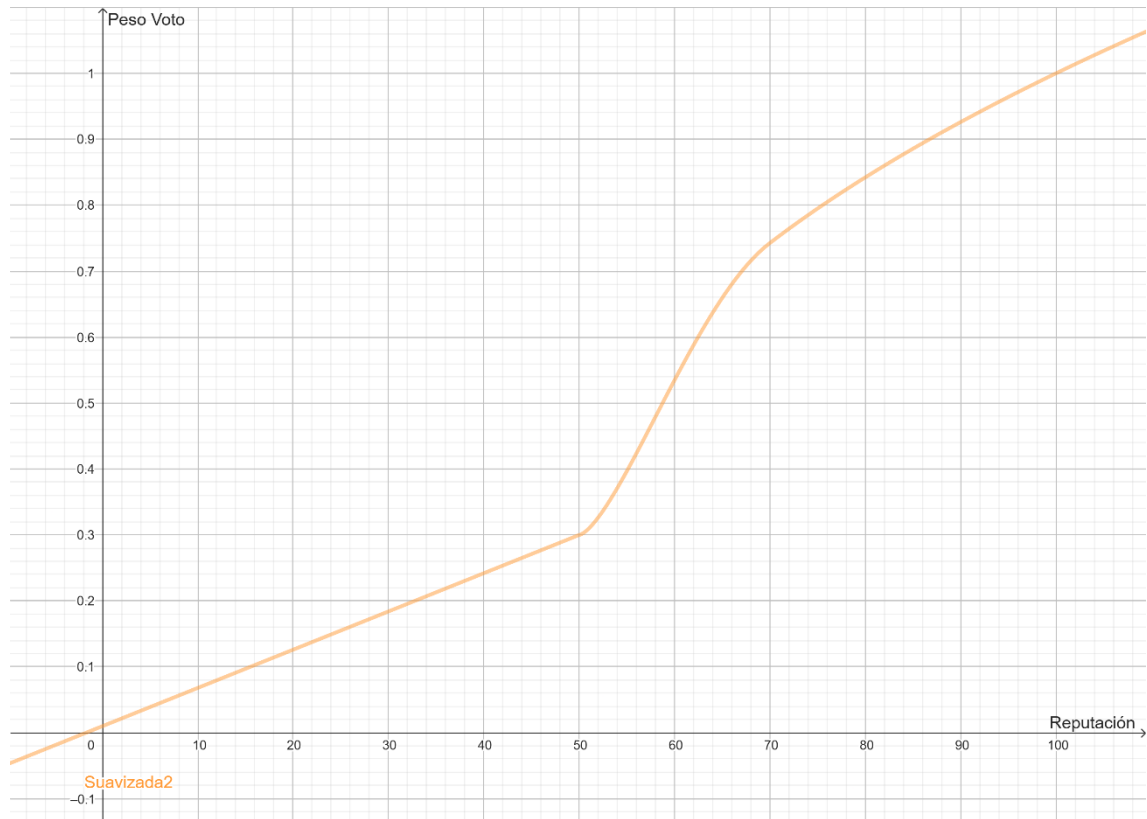
La solución finalmente adoptada fue una función definida por tres tramos:

Tramo 1 (0–50): crecimiento lineal desde 0.01 hasta 0.3.

Tramo 2 (50–70): transición suave entre el crecimiento lineal y el curvo, mediante una interpolación entre ambos.

Tramo 3 (>70): crecimiento mediante raíz cuadrada, manteniendo el valor máximo en 1.

Esta estructura permite mantener la progresión inicial gradual, evitar el salto drástico de la primera opción, y acelerar el impacto del voto conforme aumenta la reputación



$$\text{Suavizada2}(x) = \begin{cases} 0.01 + 0.29 \cdot \frac{x}{50} & : x \leq 50 \\ \left(1 - \frac{x-50}{20}\right)^2 \cdot 0.3 + \left(2 \left(1 - \frac{x-50}{20}\right) \frac{x-50}{20} + \left(\frac{x-50}{20}\right)^2\right) \left(0.3 + 0.7 \sqrt{\frac{x-50}{50}}\right) & : 50 < x \leq 70 \\ 0.3 + 0.7 \sqrt{\frac{x-50}{50}} & : \text{en caso contrario} \end{cases}$$

Ilustración 19 - Función con Transición Suave (50 -70)

Las tres opciones comparadas en un mismo gráfico quedarían de la siguiente manera:

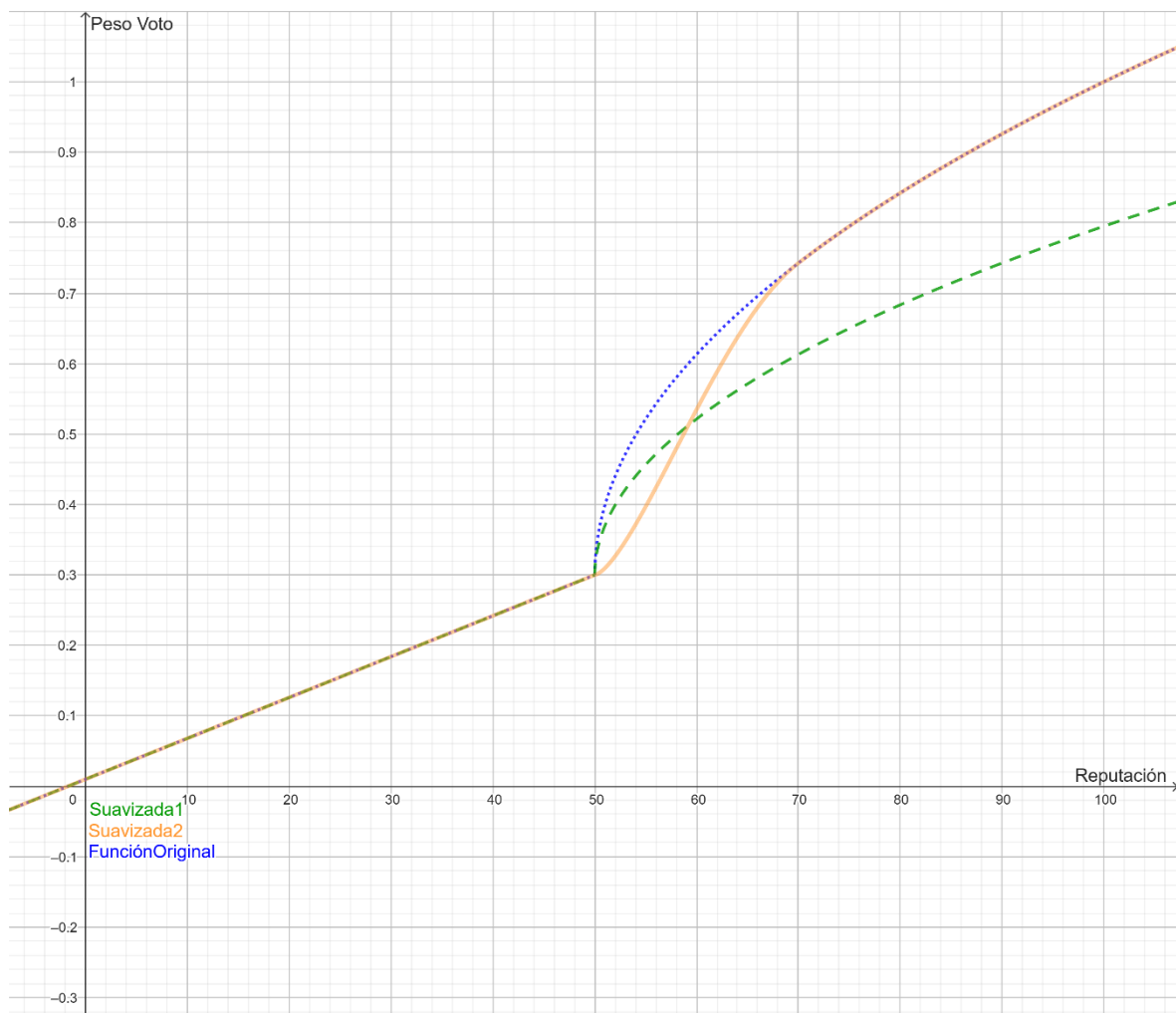


Ilustración 20 - Comparativa de Funciones

Por último, se muestra una tabla con los muestreos realizados para la elección del tipo de función, contemplando los puntos críticos de reputación y el peso del voto que se calcularía (truncado a dos decimales para su mejor comparativa).

Reputación	Función Original (opción 1)	Función Final (opción 3)	Función Suavizada (opción 2)
10	0.06	0.06	0.06
20	0.12	0.12	0.12
30	0.18	0.18	0.18
40	0.24	0.24	0.24

50	0.30	0.30	0.30
52	0.43	0.32	0.39
55	0.52	0.39	0.45
57	0.56	0.45	0.48
60	0.61	0.53	0.52
62	0.64	0.58	0.54
65	0.68	0.65	0.57
67	0.70	0.69	0.58
70	0.74	0.74	0.61
80	0.84	0.84	0.68
90	0.92	0.92	0.74
100	1.00	1.00	0.80

Tabla 59 - Comparativa entre funciones de peso de voto

5.4.2 Sistema de categorización de artículos y de aumento/disminución de la reputación

Una vez explicados los distintos parámetros de veracidad diseñados para la aplicación, se puede explicar el sistema de categorización de artículos y aumento/disminución de la reputación. Este algoritmo es la última pieza del engranaje del sistema, siendo el encargado de categorizar la información publicada en base a las opiniones de la comunidad y recompensar o castigar a los usuarios por las acciones que han tomado.

Para comenzar la explicación es necesario mencionar que, al momento de publicar un artículo, dicho artículo solo se verá afectado por la reputación de su autor, resultando así impulsada su puntuación inicial y siendo categorizado como ‘Sin evaluar’. Una vez ha sido publicado, dicho artículo será accesible por el resto de los usuarios de la aplicación, los cuales lo podrán votar, tanto positiva como negativamente.

Los votos realizados por los usuarios tendrán un impacto inmediato en la puntuación del artículo, como se ha comentado en la sección anterior, sin embargo, al depositarse el voto

también se realiza un segundo cálculo en base a la reputación del autor, este segundo cálculo se corresponde con el que tendrá el voto en el “*pool* de votos”.

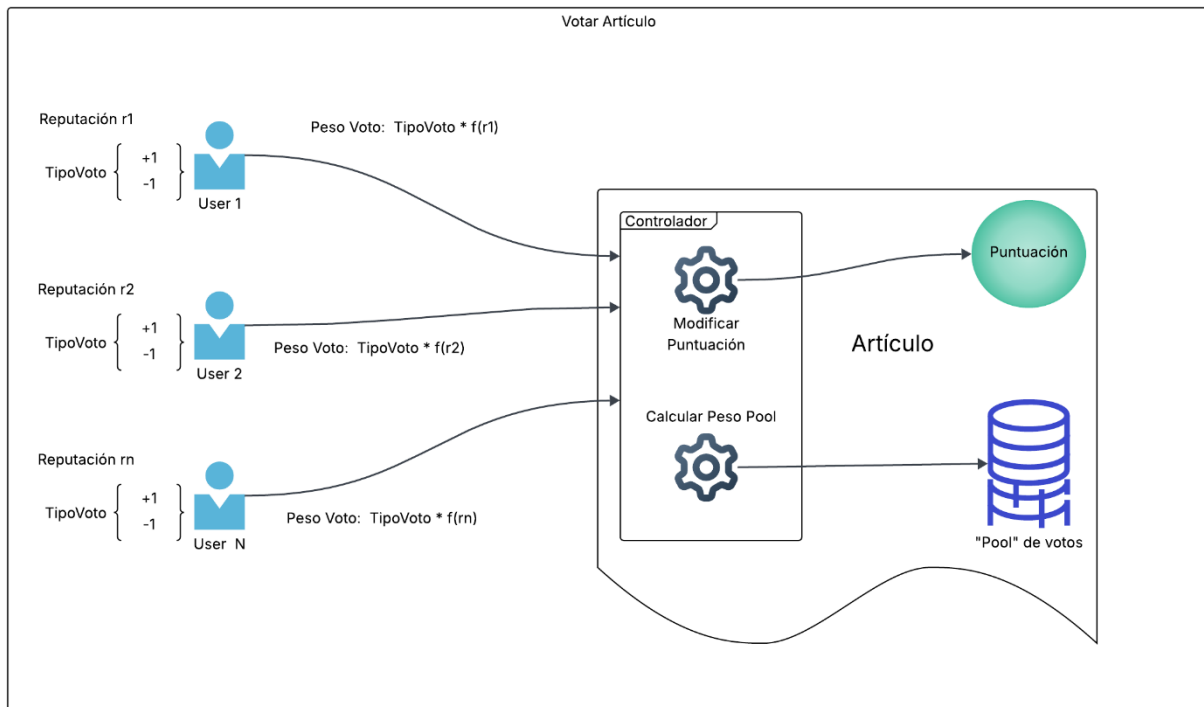


Ilustración 21 - Sistema de categorización de artículos y de aumento/disminución de la reputación I

Este “*pool* de votos” funciona como un almacén donde, en función de la reputación de usuario, se almacenarán más o menos entradas. Un ejemplo, utilizando la captura anterior es imaginar que el usuario 1 tiene una reputación de 90 puntos, cuando este usuario vote el artículo, el sistema añadirá al *pool*, 3 votos provenientes de este usuario, registrando también el tipo de voto (negativo o positivo). Por otro lado, imaginemos que los usuarios 2 y 3 tienen una reputación de 25 y 32 puntos respectivamente, cuando el sistema reciba sus votos sólo añadirá un voto de cada usuario al *pool* de votos.

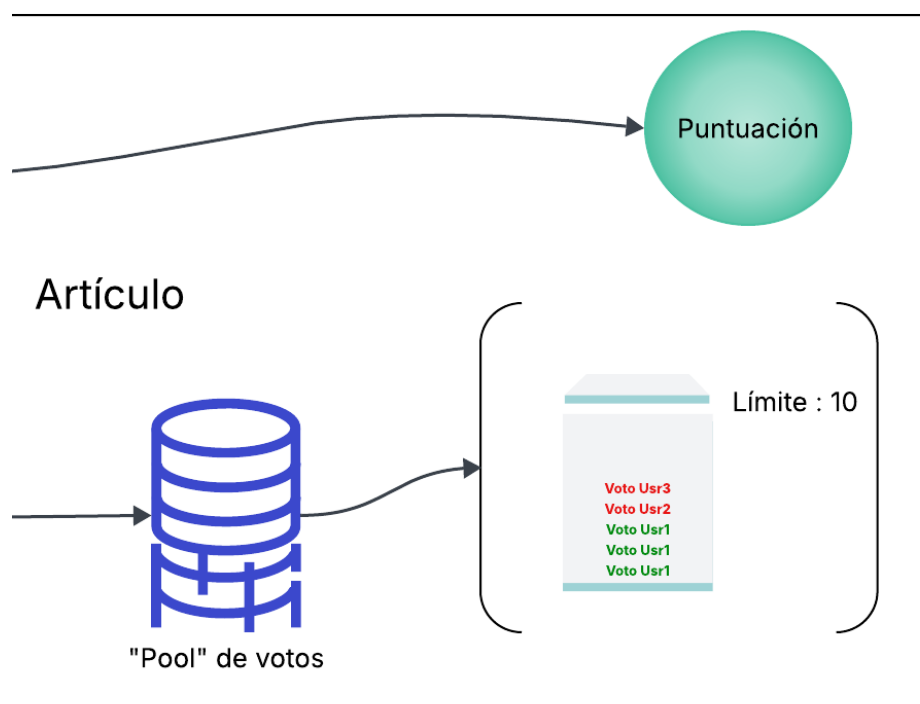


Ilustración 22 - Sistema de categorización de artículos y de aumento/disminución de la reputación II

Como se puede observar en la captura anterior el sistema va añadiendo votos en el *pool* en función de la reputación de los votantes, hasta que llega al límite de 10 votos en el *pool*. En este momento, el sistema detectará que el *pool* está lleno y evaluará los votos registrados para registrar el artículo. Imaginemos para nuestro ejemplo anterior que el artículo ha recibido los siguientes votos:

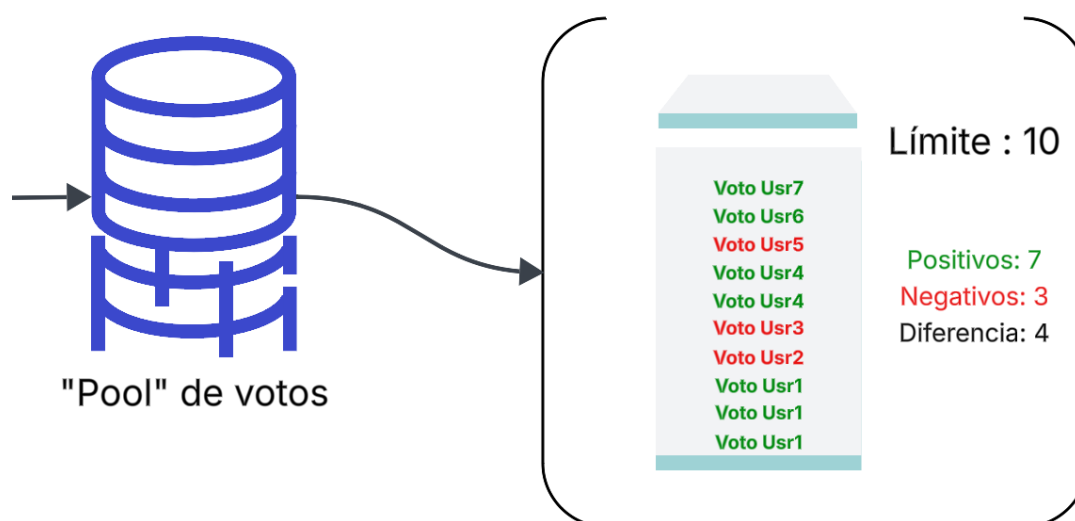


Ilustración 23 - Sistema de categorización de artículos y de aumento/disminución de la reputación III

En este caso, el artículo ha recibido un voto positivo del usuario 4, el cual, por su reputación, aporta dos votos al *pool* y tres votos más de usuarios que han aportado un solo voto. Una vez llegado el límite el sistema realiza un recuento de los votos positivos y los votos negativos del *pool*, la diferencia del resultado obtenido será utilizada por el sistema para categorizar la calidad del artículo de entre tres categorías:

Diferencia calculada	Categoría del artículo
(1.5, 10]	Verificado
[-1.5, 1.5]	Neutro
(-1.5, -10]	Poco fiable

Tabla 60 - Categorías de un artículo

En este punto, llegamos a la primera parte del ciclo de vida del algoritmo, en el momento en el que es categorizado en base al *pool* de votos. De esta manera la información es evaluada en base a la opinión de la comunidad, donde la opinión de aquellos autores con una mayor reputación, tendrá más peso a la hora de evaluar la información.

La segunda parte del ciclo de vida del algoritmo consiste en recompensar o castigar con reputación al autor del artículo y a los usuarios que lo han votado, de esta manera si un autor publica un artículo que es categorizado como “Verificado”, verá su reputación incrementada, al igual que los usuarios que votaron a favor del artículo.

Repercusión Autor (Reputación)	Categoría	Acción Votante	Repercusión Votante (Reputación)
+5	Verificado	Voto a favor	+2
		Voto en contra	-2
N/A	Neutro	Voto a favor	N/A
		Voto en contra	
-5	Poco Fiable	Voto a favor	-2
		Voto en contra	+2

Tabla 61 - Repercusiones en la reputación

Esto en nuestro ejemplo anterior se vería de la siguiente manera:

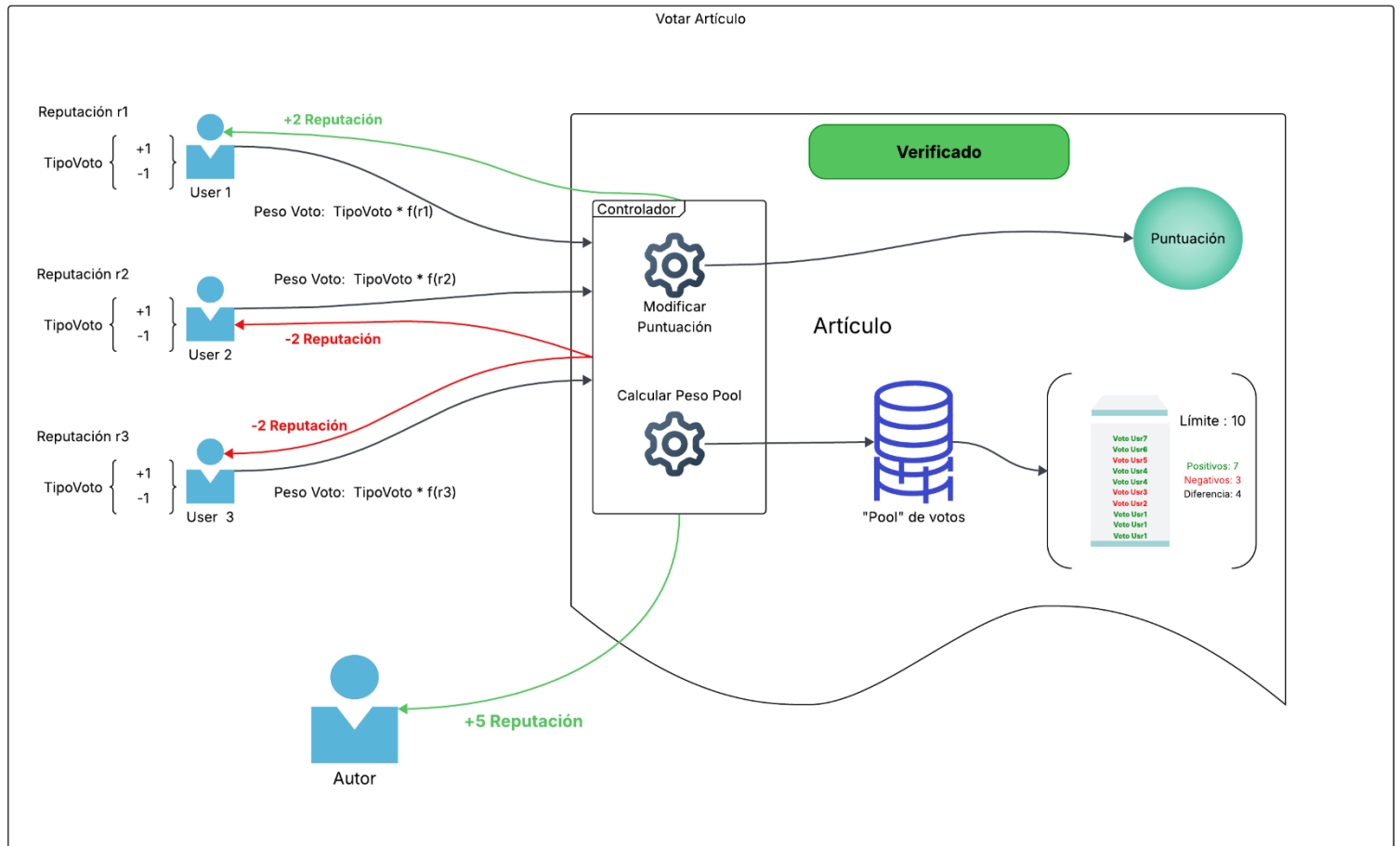


Ilustración 24 - Sistema de categorización de artículos y de aumento/disminución de la reputación IV

Como última consideración para tener en cuenta, sería injusto que una vez categorizado un artículo este permaneciese en esa categoría para siempre, ya que igual, con un muestreo de votos más amplio, la comunidad lo recolocaría en otra categoría. Es por este motivo por lo que están las rondas de categorización. Las rondas de categorización son un sistema que permite que un artículo sea recategorizado. En la versión inicial de FACTOR se han establecido dos rondas de categorización, una cuando el *pool* se llena con 10 votos y otra cuando ese mismo *pool* alcanza los 30 votos. Con este sistema escalable, nos aseguramos una categorización de la información cada vez más precisa, basándonos en muestreos de votos más grandes.

5.4.3 Sistema de *feed*

Una vez explicados los diferentes parámetros de la aplicación y el sistema de veracidad la última implementación reseñable por comentar es el sistema de *feed*. Este sistema

muestra en la página principal de la aplicación una selección de artículos y temáticas en base a diferentes parámetros.

Por el lado de las temáticas, el sistema simplemente mostrará las últimas 6 temáticas de las que se haya publicado un artículo.

Por el lado de los artículos el sistema los listará en el *feed* en base a tres criterios:

Frescura: Es un indicador que mide lo reciente que ha sido publicado un artículo, cuanto más reciente mayor relevancia tendrá el artículo para salir en el *feed*.

Puntuación: La puntuación es el parámetro de veracidad explicado anteriormente, cuanto mayor puntuación mayor relevancia tendrá el artículo para salir en el *feed*.

Categoría: Por último, está la categoría, este parámetro suma a la puntuación del artículo si está categorizado como “verificado”, resta si es “poco fiable” y no modifica la puntuación si es “neutro”.

La unión de estos tres parámetros da lugar a un indicador que muestra los artículos recientes, mejor puntuados y verificados por la comunidad, sin dejar de dar cabida a nuevos autores con menor repercusión, ya que la frescura y categoría otorgan mayor peso al indicador.

5.5 Diagramas de Secuencia

En la siguiente sección se van a mostrar los diagramas de secuencia de las interacciones más relevantes.

Los diagramas de secuencia son una herramienta clave en el diseño de sistemas, ya que permiten visualizar cómo interactúan los diferentes actores y componentes de una aplicación en un proceso específico.

A continuación, se muestran los diagramas de secuencia del proceso de iniciar sesión y publicar un artículo.

5.5.1 Diagrama de secuencia iniciar sesión.

Como aspecto destacable a la hora de iniciar sesión, cabe mencionar que se realiza a través del establecimiento de una cookie en el navegador configurada con la cabecera

httpOnly a *true*, de esta manera solo se puede establecer esta cookie si el *backend* valida las credenciales.

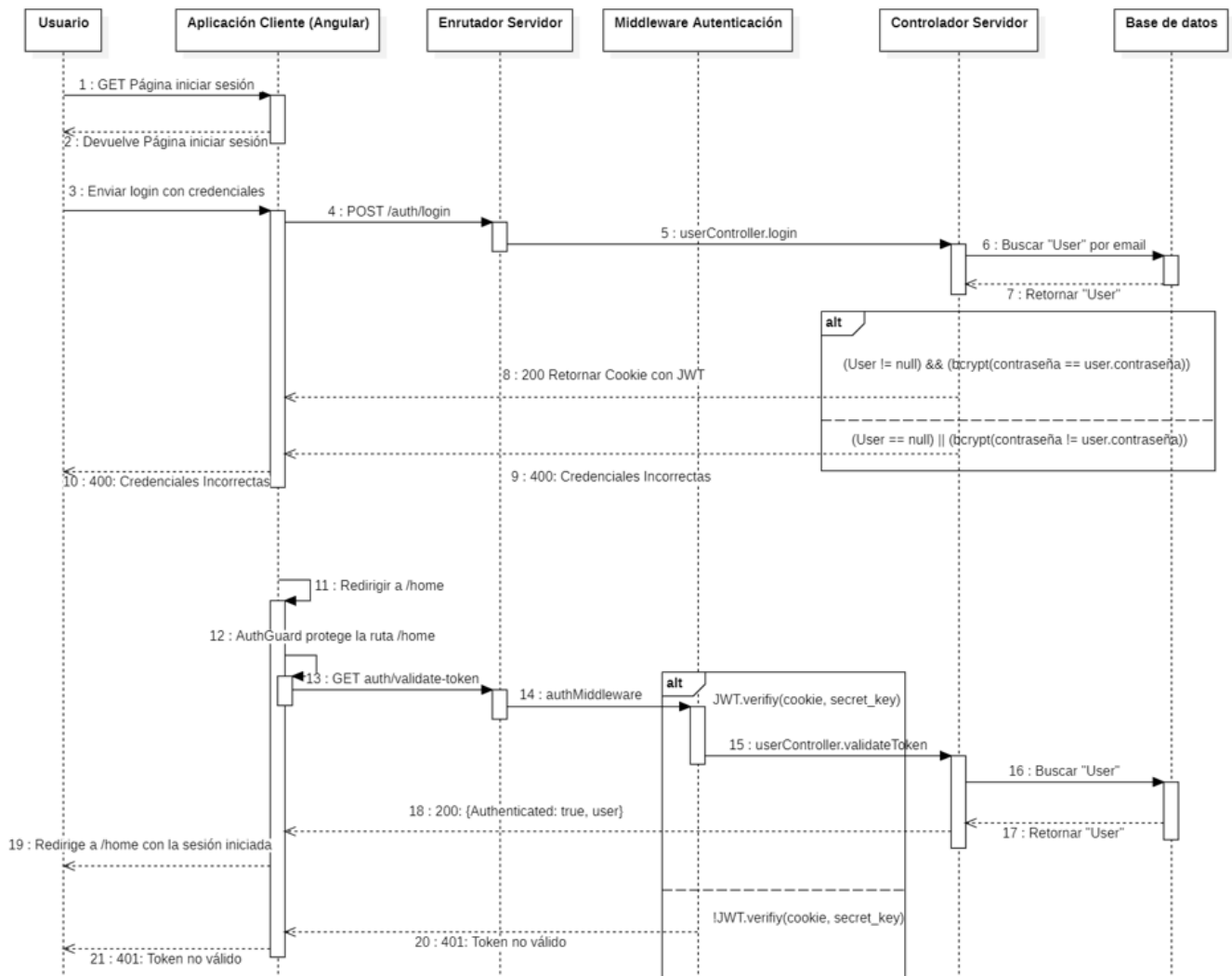


Ilustración 25 -Diagrama de Secuencia Iniciar Sesión

5.5.2 Diagrama de secuencia publicar artículo

En este diagrama de secuencia se muestra el flujo por el cual el usuario publica un artículo, evaluando las validaciones y obligatoriedad del formulario de creación, las restricciones del formato y tamaño del archivo a pdf y 20MB. Por último, el sistema verifica que el usuario posea la reputación mínima requerida para publicar y que no haya excedido su límite de publicaciones mensuales.

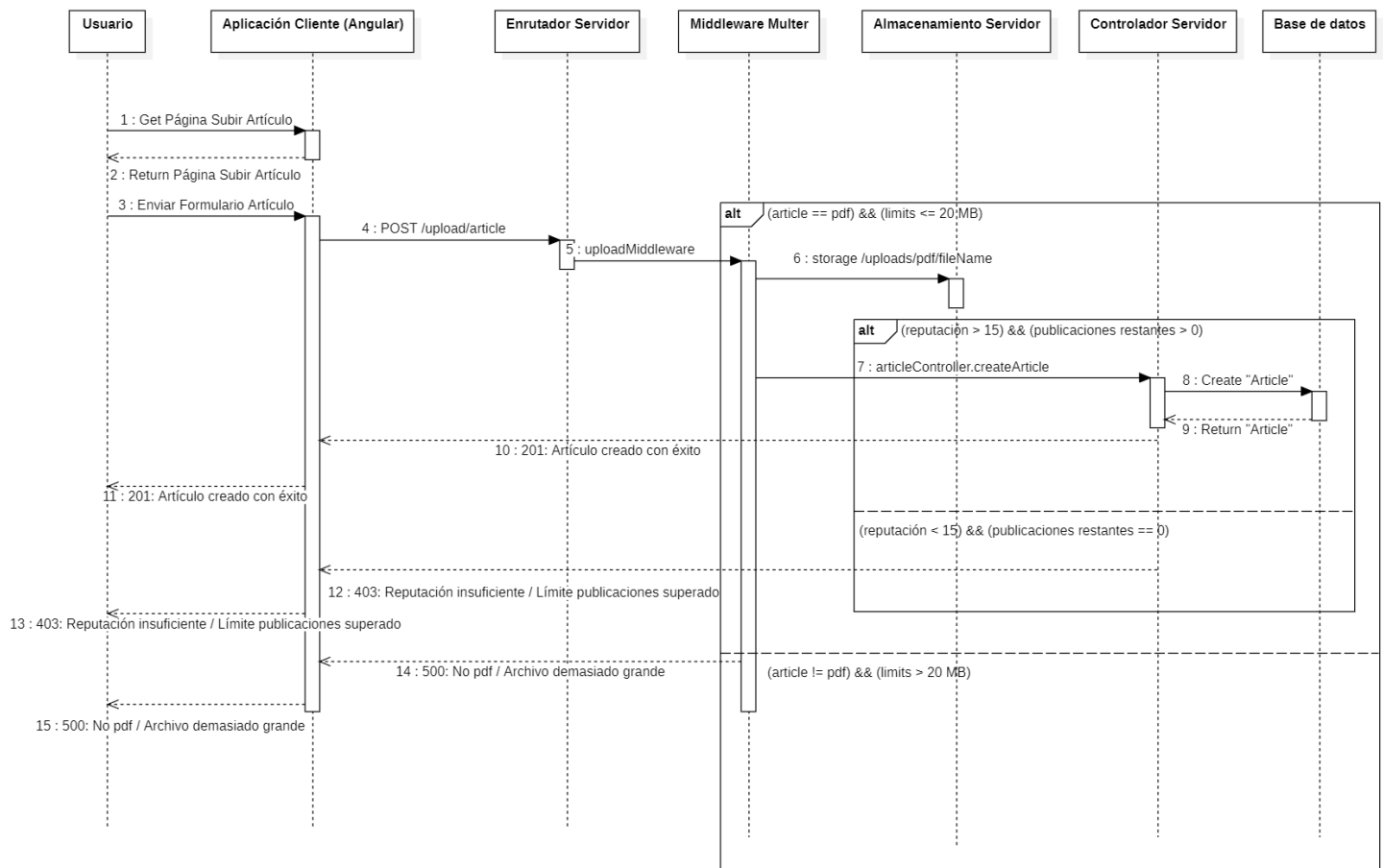


Ilustración 26 - Diagrama de Secuencia Publicar Artículo

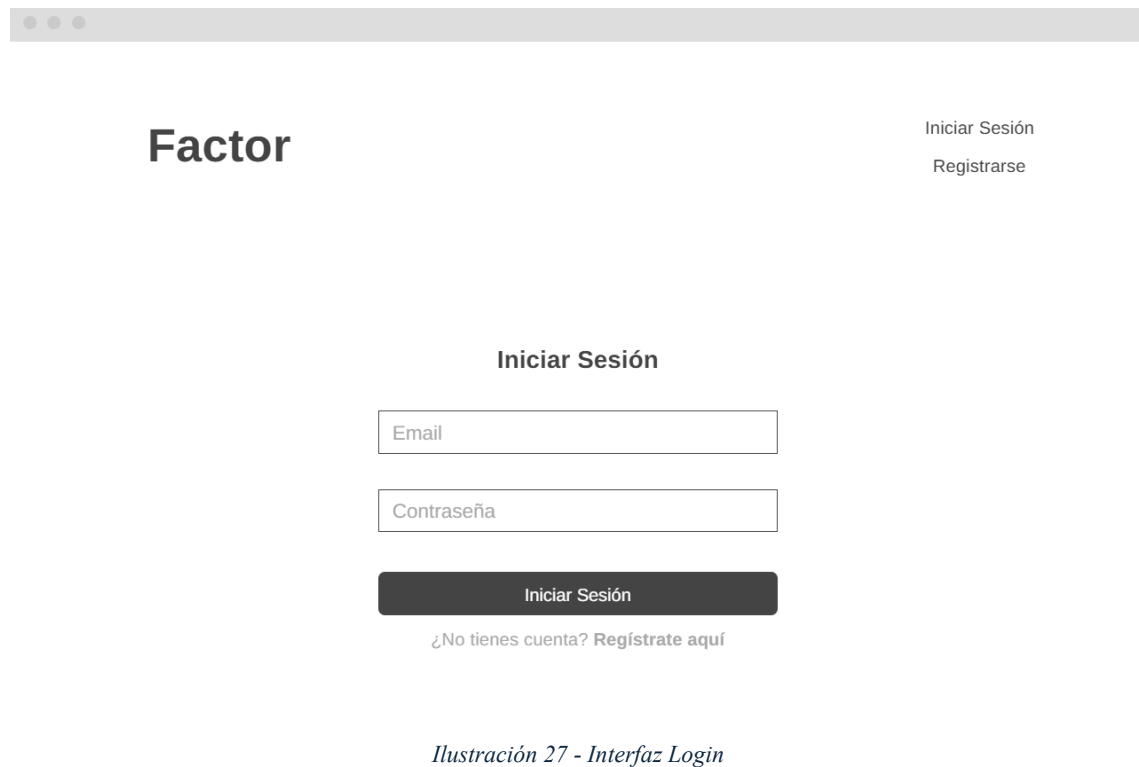
5.6 Diseño de interfaces de usuario

La última sección del capítulo de diseño corresponde a los bocetos iniciales elaborados para cada una de las vistas de la aplicación. Estos bocetos han servido como una guía para la implementación de las principales interfaces de la aplicación.

Para su creación utilicé la aplicación WIREFRAME.CC que permite crear bocetos a partir de componentes de manera rápida y sencilla.

Interfaz *Login*:

La primera vista que aparece al arrancar la aplicación se corresponde con el Login. El formulario presenta dos campos de email y contraseña de un usuario ya existente y un botón para iniciar sesión.



The image shows a login interface for a web application. At the top, there is a header with the name 'Mario Olmos Manso' on the left and 'Factor' on the right. Below the header, there is a navigation bar with three dots. The main content area has a large 'Factor' logo on the left and links for 'Iniciar Sesión' and 'Registrarse' on the right. The login form is centered and consists of a title 'Iniciar Sesión', two input fields for 'Email' and 'Contraseña', a dark 'Iniciar Sesión' button, and a link '¿No tienes cuenta? Regístrate aquí'.

Factor

Iniciar Sesión
Registrarse

Iniciar Sesión

Email

Contraseña

Iniciar Sesión

¿No tienes cuenta? Regístrate aquí

Ilustración 27 - Interfaz Login

Interfaz Registro

Si el usuario no dispone de una cuenta creada en la aplicación deberá ir a la pestaña de registro, disponible tanto en el menú del *navbar* como en un enlace debajo del botón de iniciar sesión. El formulario de registro consta de los campos de nombre, apellidos, email, contraseña y fecha de nacimiento.



The image shows a registration interface for a web application. At the top, there is a header with the name 'Mario Olmos Manso' on the left and 'Factor' on the right. Below the header, there is a navigation bar with three dots. The main content area has a large 'Factor' logo on the left and links for 'Iniciar Sesión' and 'Registrarse' on the right. The registration form is centered and consists of a title 'Registro de Usuario', five input fields for 'Nombre', 'Apellidos', 'Email', 'Contraseña', and 'Fecha Nacimiento', a dark 'Iniciar Sesión' button, and a link '¿Ya tienes cuenta? Inicia Sesión'.

Factor

Iniciar Sesión
Registrarse

Registro de Usuario

Nombre

Apellidos

Email

Contraseña

Fecha Nacimiento

Iniciar Sesión

¿Ya tienes cuenta? Inicia Sesión

Ilustración 28 - Interfaz Registro

Interfaz *Feed*

Una vez el usuario inicia sesión la aplicación le redirigirá a la página home, al iniciar sesión la aplicación también mostrará el menú en el *navbar* para usuarios registrados. La página *home* está dividida por los temas populares, los cuales se pueden pulsar lo que redirigirá al usuario a la página explorador con el tema seleccionado como filtro. Debajo de los temas populares encontraremos los artículos populares sobre los que podremos votar, ver su detalle y el perfil de su autor.

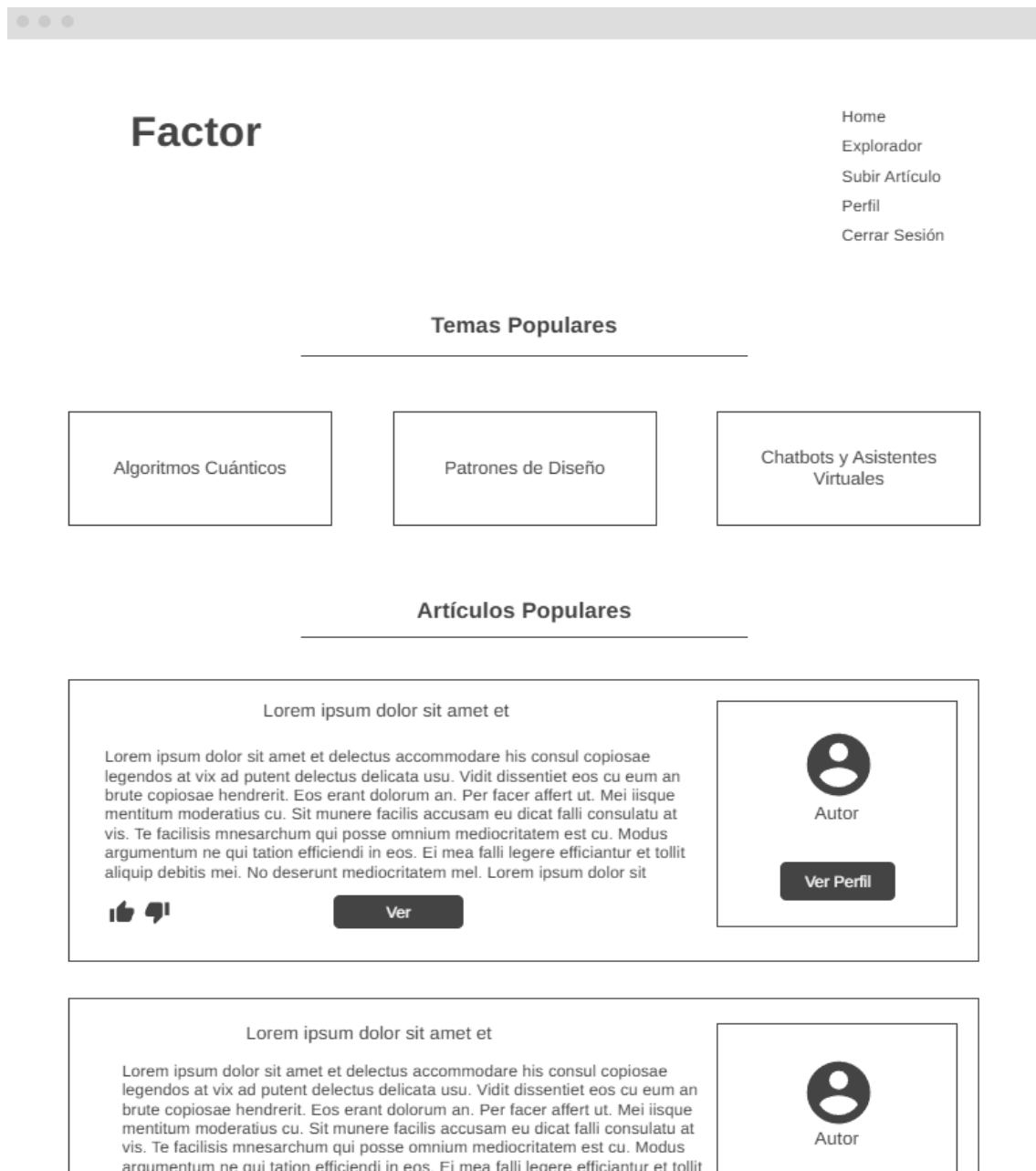


Ilustración 29 - Interfaz *Feed*

Interfaz Explorador

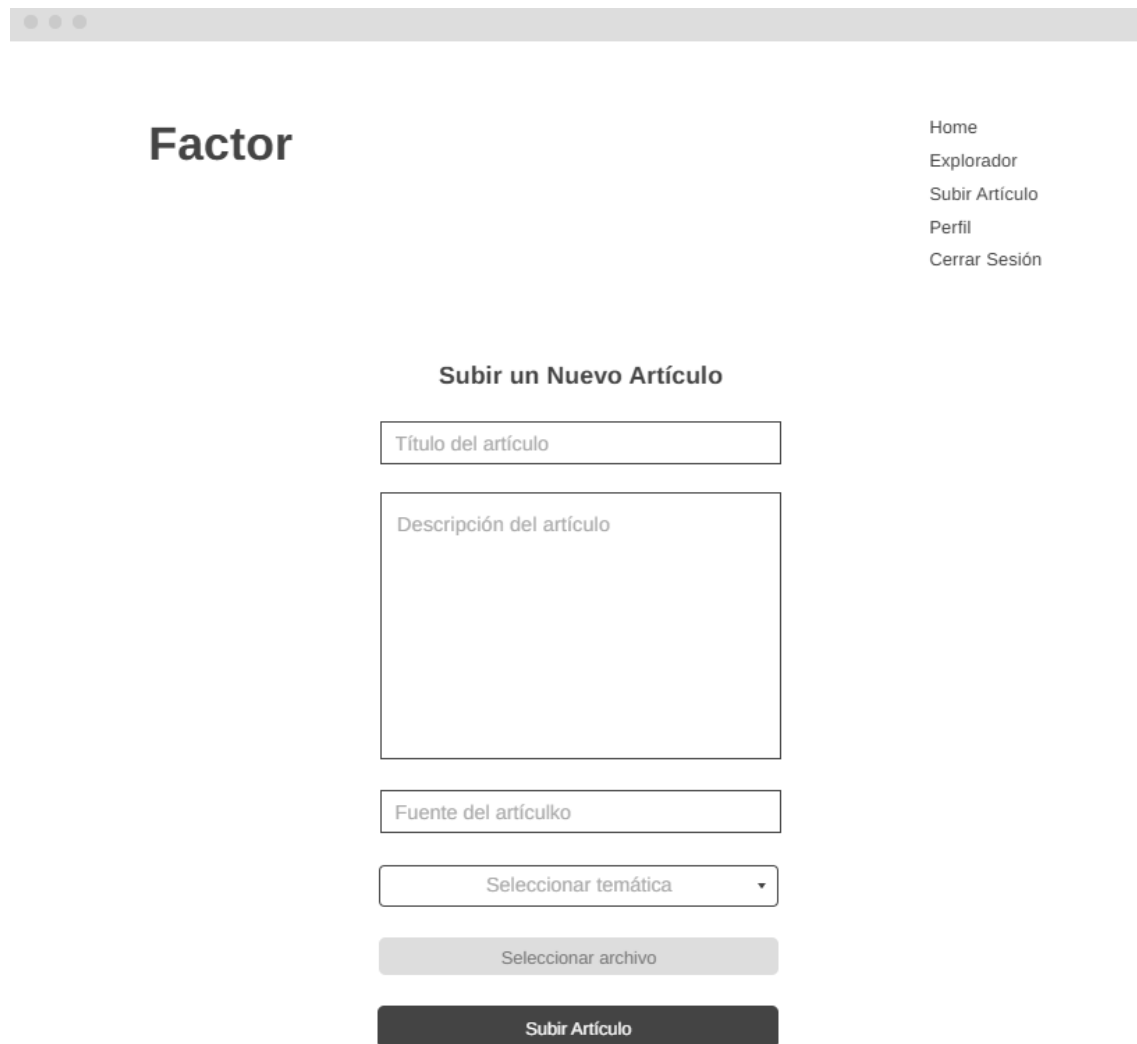
La página del explorador es accedida a través del *navbar* o bien al clicar una temática en la página *home*. Esta vista nos permite realizar una búsqueda por temática y ordenar los resultados por puntuación del artículo o por su fecha de publicación. El explorador también dispone de las mismas funcionalidades que el *home*, permitiendo votar artículos, ver su detalle y el perfil de sus autores



Ilustración 30 - Interfaz Explorador

Interfaz Subir Artículo

La siguiente página es el formulario para publicar un artículo, se accede a través del *navbar* y despliega un formulario con los datos necesarios para crear un artículo.



The screenshot shows a web interface for uploading a new article. At the top, there is a header with the 'Factor' logo on the left and a navigation menu on the right containing links: Home, Explorador, Subir Artículo, Perfil, and Cerrar Sesión. Below the header, the main content area is titled 'Subir un Nuevo Artículo'. This section contains a form with the following elements:

- A text input field labeled 'Título del artículo'.
- A larger text area labeled 'Descripción del artículo'.
- A text input field labeled 'Fuente del artículo'.
- A dropdown menu labeled 'Seleccionar temática' with a downward arrow.
- A button labeled 'Seleccionar archivo'.
- A dark button labeled 'Subir Artículo'.

Ilustración 31 - Interfaz Subir Artículo

Interfaz detalle artículo

La vista del detalle de un artículo muestra una versión extendida de la información del propio artículo y de su autor, así como un visor del pdf con el artículo completo. Esta vista es accedida desde cualquier parte de la aplicación donde aparece un artículo y nos permite votar, ver el perfil de su autor o consultar la fuente original del artículo.



Ilustración 32 - Interfaz Detalle Artículo

Interfaz perfil ajeno – información

La vista del perfil ajeno nos muestra la información de un autor diferente al usuario que ha iniciado sesión en la aplicación, podemos acceder a ella a través de cualquier artículo de ese autor. Una vez accedido nos enseña la información de ese autor, con sus datos y acreditaciones su rol en la aplicación y su actividad reciente.

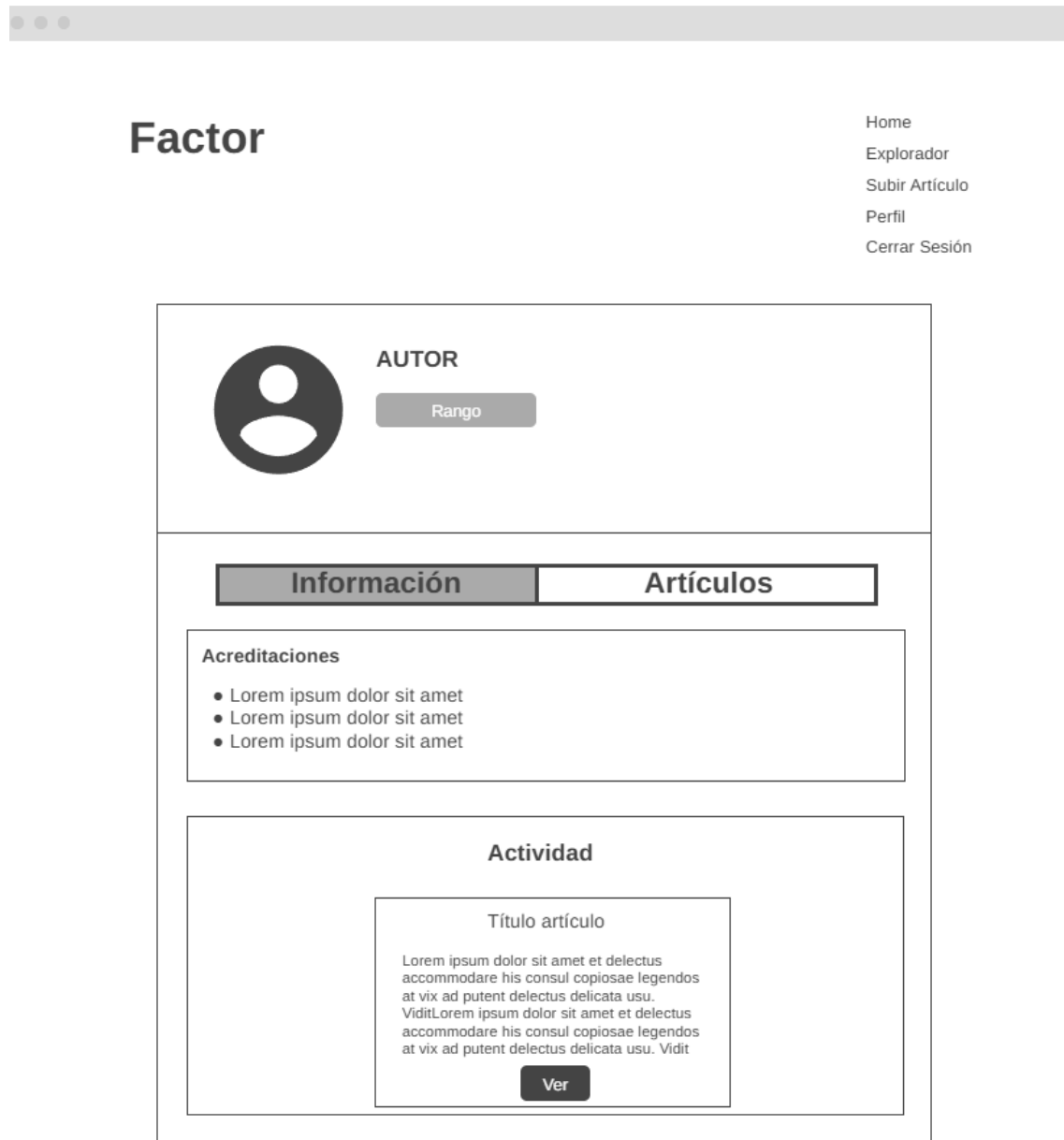


Ilustración 33 - Interfaz Perfil Ajeno - Información

Interfaz perfil ajeno – artículos

En la misma vista que la interfaz de información encontramos la pestaña de artículos, la cual despliega el listado de artículos publicados por ese autor.

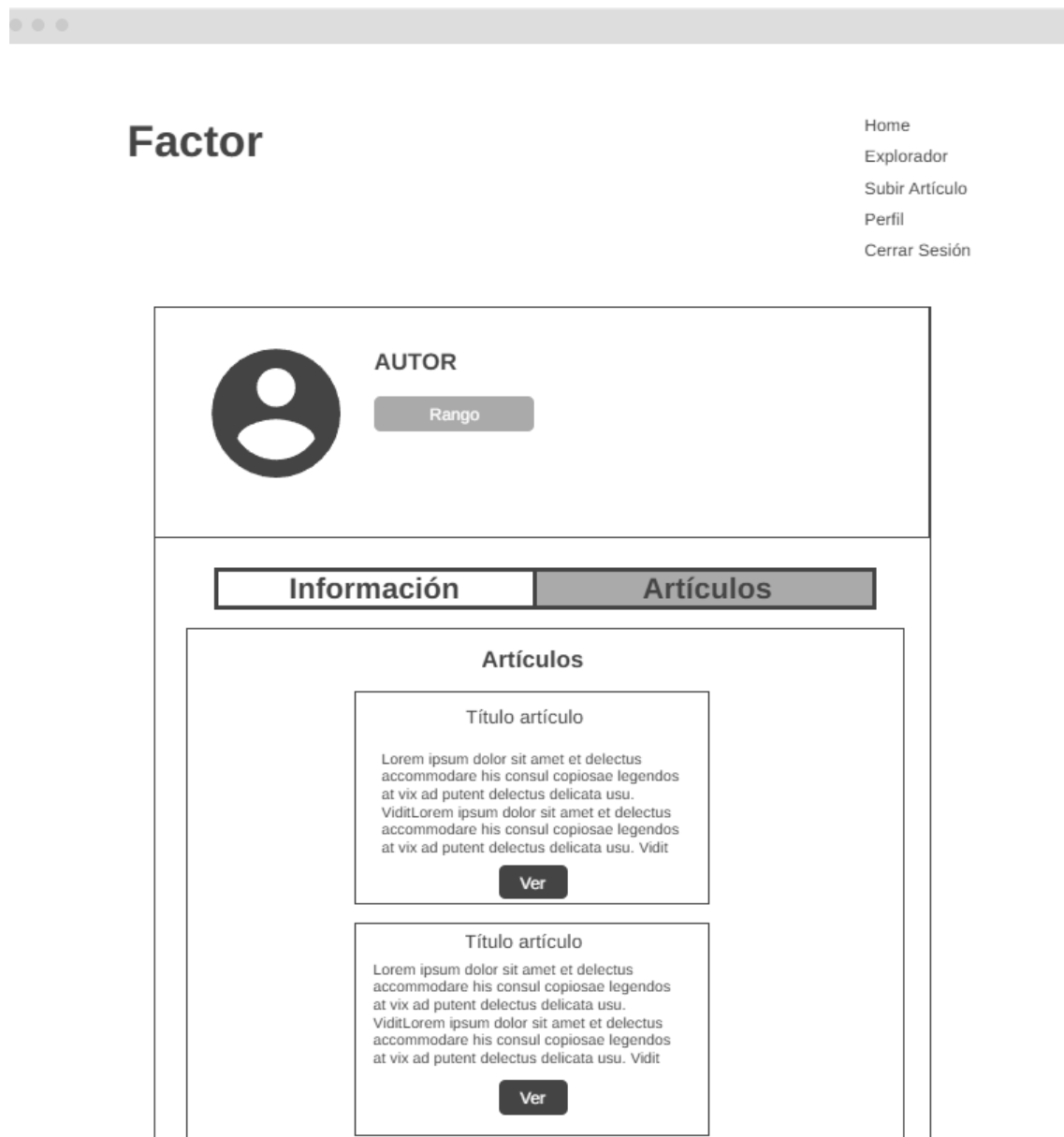


Ilustración 34 - Interfaz Perfil Ajeno - Artículos

Interfaz perfil personal – información

En la vista de perfil personal encontramos el perfil del usuario que ha iniciado sesión, se accede a ella a través del *navbar* o si presionamos el botón del perfil en un artículo propio. La pestaña de información es la misma que en un perfil ajeno con la diferencia de la ruleta de acciones, la cual contiene las funcionalidades de eliminar cuenta y consultar privilegios, también contiene información personal y el botón para editar el perfil.

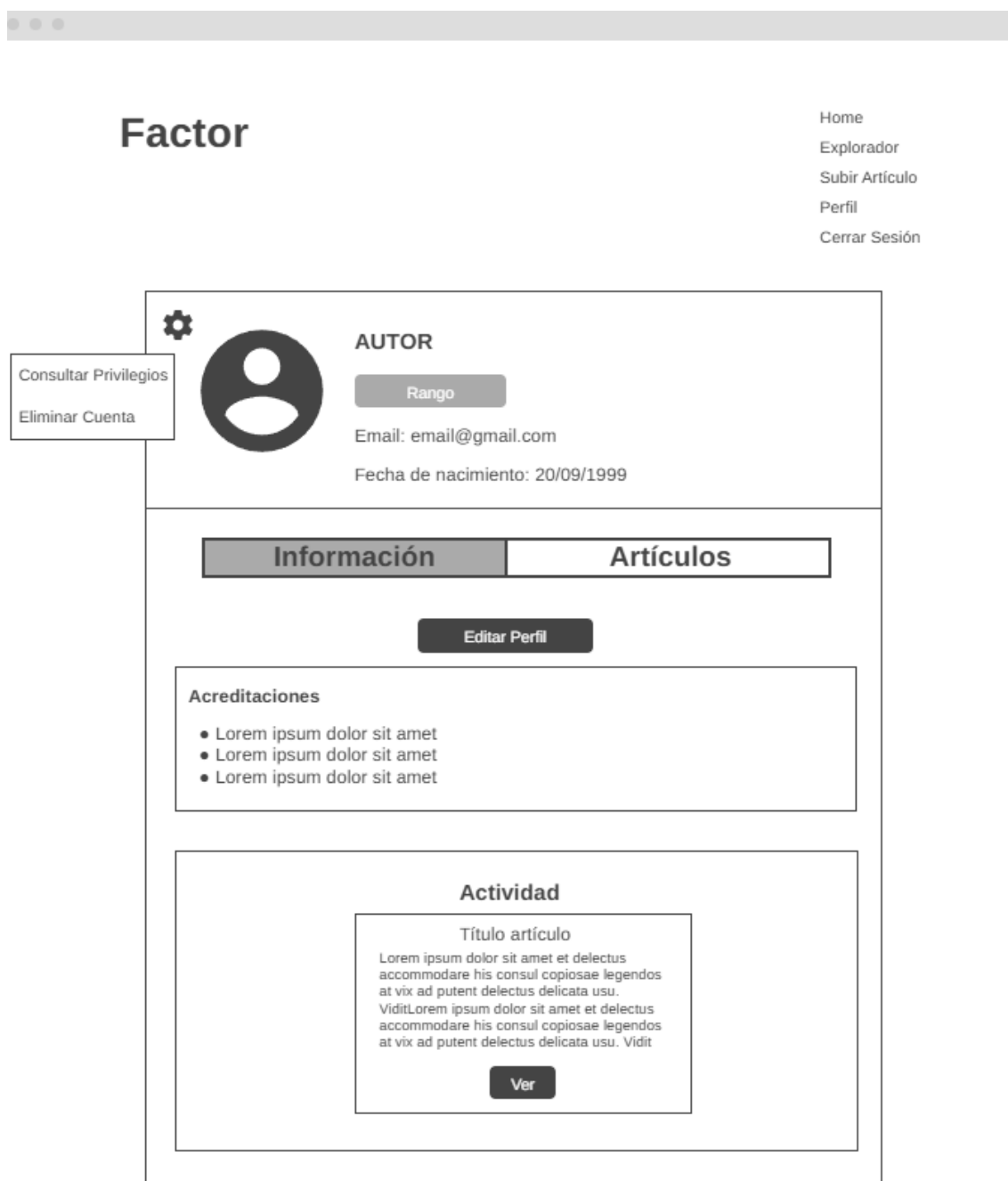


Ilustración 35 - Interfaz Perfil Personal - Información

Interfaz perfil personal – artículos

En el perfil personal en su pestaña de artículo encontramos la misma información que en un perfil ajeno, con la diferencia que al ser artículos del propio autor estará habilitado un botón para eliminarlos.

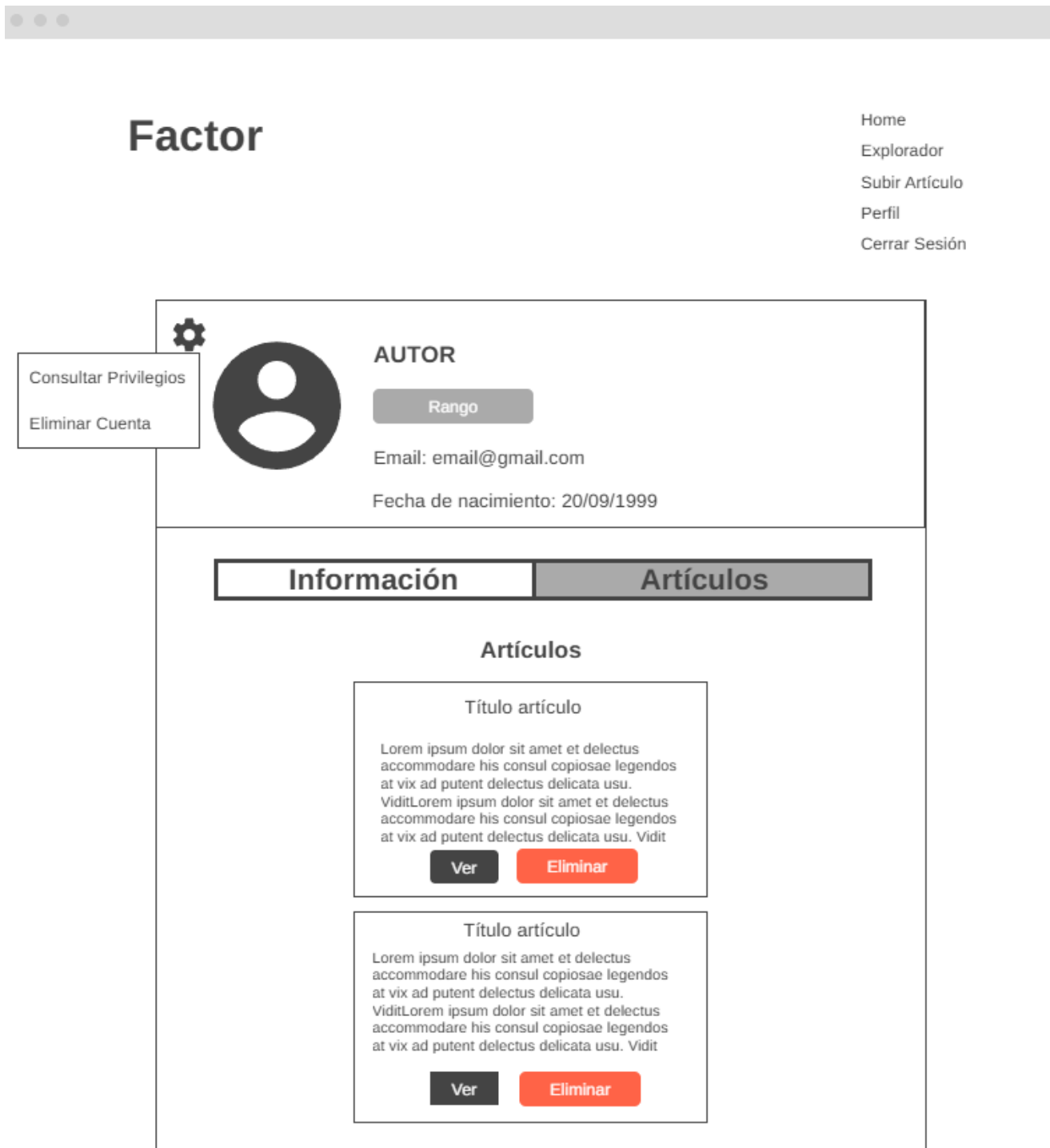


Ilustración 36 - Interfaz Perfil Personal - Artículos

Capítulo 6: Implementación y pruebas

En este capítulo se detalla el proceso de implementación del proyecto FACTOR, detallando la estructura del proyecto, las principales implementaciones desarrolladas y las pruebas realizadas para verificar su correcto funcionamiento.

6.1 Cuestiones de implementación

En esta sección se presenta la estructura del proyecto dividida en la parte de *frontend* y en la parte de *backend* ya que estos dos componentes funcionan de manera completamente independiente.

6.1.1 Backend

Como ya se comentó en la sección de entorno y tecnologías utilizadas (Sección 1.6) el *backend* se ha desarrollado con *node.js* y su *framework express* junto con la base de datos no relacional *MongoDB*.

En este componente podemos distinguir la siguiente estructura:

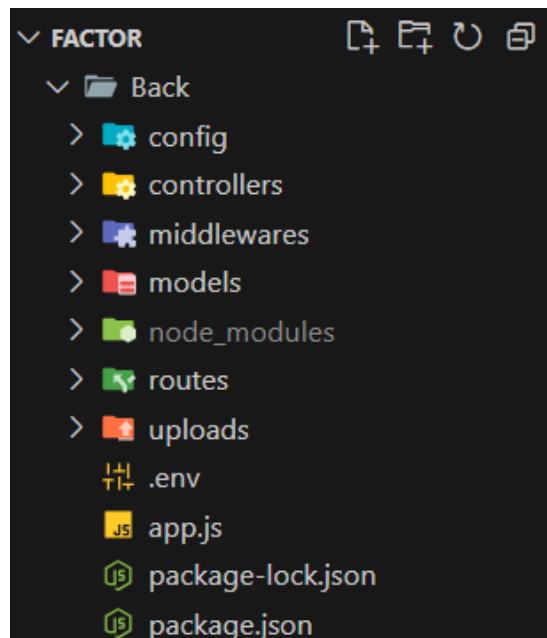


Ilustración 37 - Estructura Backend

En la captura anterior podemos observar los siguientes elementos:

- **Carpeta `config`:** Contiene un archivo llamado `db.js` con la configuración para la conexión con la base de datos.
- **Carpeta `controllers`:** Directorio con la configuración de los distintos controladores.
- **Carpeta `middlewares`:** Directorio con la configuración de los *middlewares*.
- **Carpeta `models`:** Directorio con los modelos de las entidades.
- **Carpeta `routes`:** Directorio que contiene las rutas a los diferentes *endpoints* de la API REST.
- **Carpeta `uploads`:** Almacena los archivos `.pdf` y las imágenes de la aplicación.
- **Archivo `env`:** Contiene las diferentes variables de entorno tales como la clave secreta, las credenciales para la conexión con la base de datos o la configuración de los puertos.
- **Archivo `app.js`:** Archivo esencial en *node.js*, es el archivo principal del *backend* y en él se definen las características principales tales como el uso de *express*, *cors*, *cookies* o la ruta donde se acceden a los archivos estáticos.

Una vez enumerados los diferentes archivos y directorios en el primer nivel vamos a profundizar en los directorios de los controladores, los *middlewares*, los modelos y las rutas. Estos cuatro directorios contienen todos los archivos con la lógica de la aplicación.

6.1.1.1 Controladores

En este subdirectorio encontramos todos los archivos JavaScript correspondientes a los diferentes controladores del *backend*, en estos controladores se declaran todas las operaciones CRUD (*Create, Read, Update and Delete*), así como el resto de las operaciones que establecen la lógica del *backend*.

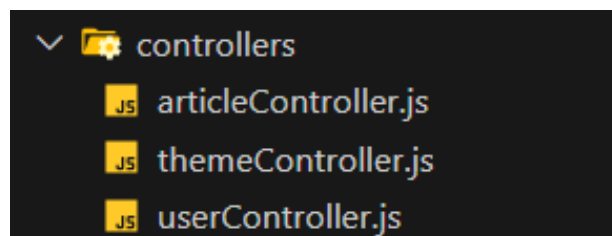


Ilustración 38 - Controladores

Como podemos observar se ha creado un controlador para cada una de las entidades principales definidas, de esta manera se intenta desacoplar la lógica de cada una de las

entidades y sus respectivas operaciones, resultando un código mucho más limpio, robusto y reutilizable.

6.1.1.2 *Middlewares*

En este subdirectorio encontramos los *middlewares* de la aplicación, sirven como intermediarios para tareas transversales ejecutándose antes de pasar al controlador final. De esta manera se ahorra el repetir estos fragmentos de código en cada controlador. Además, en FACTOR también se han establecido *middlewares* de subida de archivos tanto para gestionar la subida de pdf como de imágenes, centralizando todo el flujo de procesamiento y gestionando posibles errores antes de que la petición llegue al controlador final.

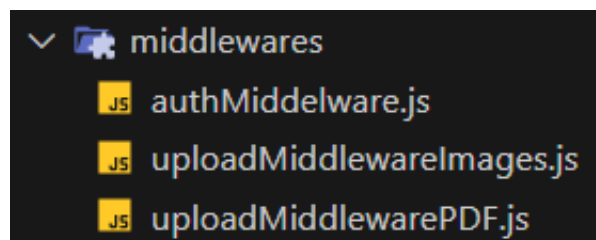


Ilustración 39 - Middlewares

6.1.1.3 Modelos

En el subdirectorio de los modelos están definidos los diferentes esquemas y definiciones de todas las entidades y subentidades almacenadas en la base de datos. Cada archivo describe los campos y tipos de datos que se almacenan y en algunos casos validaciones específicas como valores por defecto, valores posibles para ese dato, intervalos...

De esta manera se encapsula toda la lógica relevante a la estructura, formato y validaciones de tus datos en un único lugar.

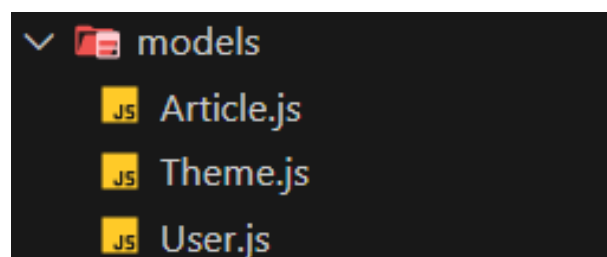


Ilustración 40 – Modelos Backend

6.1.1.4 Rutas

Por último, el directorio de `routes` define todos los *endpoints* que conforman la API, es decir, las direcciones y métodos HTTP que los clientes pueden invocar. En cada archivo de rutas se establece qué *middlewares* deben ejecutarse antes de llegar al controlador en caso de haberlos y cuál de los controladores (y su método) gestionará la petición. Así se mantiene separada la definición de las rutas de la lógica de negocio, facilitando la comprensión y organización del flujo de la aplicación.

En la siguiente tabla se enumeran todos los *endpoints* de la aplicación, enumerando la ruta genérica establecida, la ruta completa, el tipo de operación y una descripción de los *middlewares* y archivos que participan en esa ruta.

Router	Ruta	Operación	Descripción
Auth	/api/auth/register	POST	Crea un nuevo usuario.
	/api/auth/login	POST	Inicia sesión y establece una <i>cookie</i> de sesión.
	/api/auth/validate-token	GET	Valida la <i>cookie</i> establecida y devuelve al objeto usuario que ha iniciado sesión.
	/api/auth/logout	POST	Valida la <i>cookie</i> de sesión y la elimina.
Profile	/api/profile/getUser/:username	GET	Devuelve la información del perfil de un usuario.
	/api/profile/delete	DELETE	Verifica la <i>cookie</i> de sesión y elimina la cuenta del usuario.
	/api/profile/update	PUT	Verifica la <i>cookie</i> de sesión, sube la imagen de perfil en caso de ser cambiada y actualiza la información de un usuario.
	/api/profile/privileges	GET	Verifica la <i>cookie</i> de sesión y devuelve los privilegios de un usuario.
Theme	/api/theme/getThemes	GET	Devuelve toda la jerarquía de temáticas almacenada en la base de datos.
	/api/theme/getTrendyThemes	GET	Devuelve las temáticas populares utilizadas en el <i>feed</i> .
Upload	/api/upload	POST	Verifica la <i>cookie</i> de sesión, llama al

			<i>middleware</i> para la subida del archivo pdf y crea el artículo.
Articles	/api/articles/getArticles	GET	Verifica la <i>cookie</i> de sesión y devuelve los artículos mostrados en el <i>feed</i> , o si recibe parámetros de filtro también se utiliza en la página explorador.
	/api/articles/vote	POST	Verifica la <i>cookie</i> de sesión y registra un voto a un artículo, realizando las operaciones lógicas que conlleva en la reputación y puntuación de las entidades que afecta.
	/api/articles/getArticles-ByUser	GET	Verifica la <i>cookie</i> de sesión y devuelve los artículos de un autor en concreto.
	/api/articles/getActivity-ByUser	GET	Verifica la <i>cookie</i> de sesión y devuelve los artículos con los que ha interactuado un usuario determinado.
	/api/articles/deleteArticle	DELETE	Verifica la <i>cookie</i> de sesión y elimina un artículo.
	/api/articles/:id	GET	Verifica la <i>cookie</i> de sesión y devuelve el detalle de un artículo.
Uploads	/api/uploads/pdf/:name	STATIC	Ruta estática para acceder a los pdf
	/api/uploads/image/:name	STATIC	Ruta estática para acceder a las imágenes

6.1.2 Frontend

En esta sección se detalla la estructura y organización del *frontend* de FACTOR, esta estructura basada en el *framework Angular* facilita la organización, reutilización y mantenimiento del código perteneciente al *frontend*.

En los siguientes puntos se describen los componentes y archivos principales

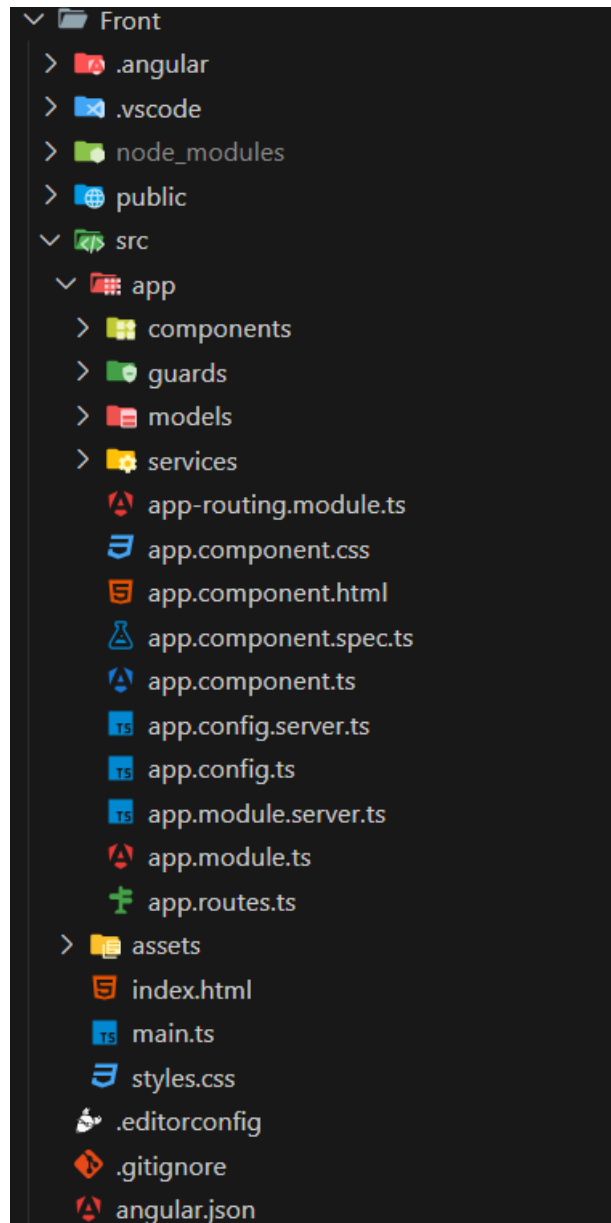


Ilustración 41 - Estructura Frontend

En la captura anterior podemos destacar los siguientes directorios y archivos:

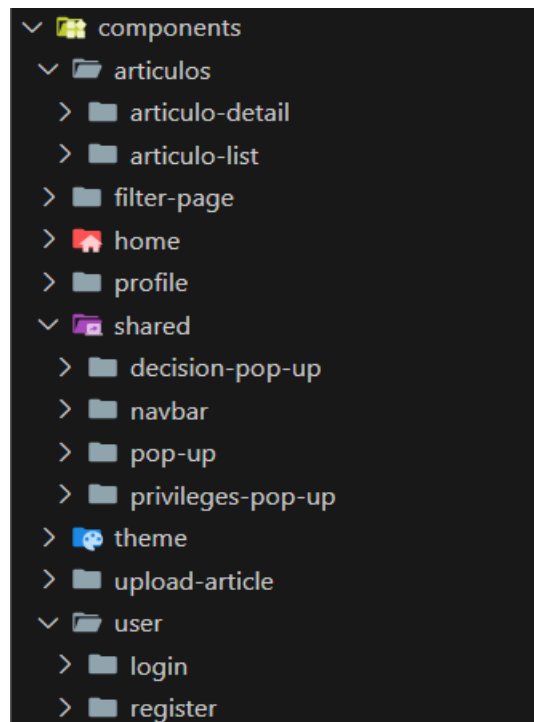
- **Carpeta `components`:** Contiene todos los componentes de la aplicación. Estos componentes se han diseñado para su reutilización en diferentes partes de la aplicación y cada componente contiene un archivo *typescript*, un archivo *css* y uno *html*.

- **Carpeta `services`:** Esta carpeta contiene los servicios, son clases diseñadas para compartir lógica y datos entre diferentes componentes y es donde se realizan las llamadas a la API.
- **Carpeta `guards`:** Contiene el guardián de las rutas. Estos guardianes son clases diseñadas para proteger el acceso a las diferentes páginas de la aplicación.
- **Carpeta `models`:** Contiene las interfaces que definen la forma de los datos utilizada en la app, facilitando el tipado y evitando errores.
- **Archivo `app-routing-module.ts`:** El módulo de enrutamiento, en este archivo se declaran las distintas rutas de la aplicación, los componentes que se cargan en cada ruta y el guardián que las protege.
- **Archivo `app-module.ts`:** Es el módulo raíz de *Angular* y esencial para que funcione la aplicación. Aquí se recogen las declaraciones de todos los módulos, componentes y servicios utilizados en la aplicación.
- **Archivo `index.html`:** Archivo *html* principal, utilizado para definir metadatos, fuentes y estilos.
- **Archivo `style.css`:** Archivo de estilo global para la aplicación, en este archivo se han declarado las reglas de estilo globales, así como el estilo de clases *html* compartidas entre varios componentes. De esta manera que en los archivos *css* propios de cada componente solo estarán definidas las reglas de estilo correspondientes a las clases *html* únicas en ese componente, así como reglas específicas de otras clases compartidas.
- **Archivo `angular.json`:** Archivo de configuración del proyecto de *Angular*. Se puede configurar reglas de compilado, que archivos se incluyen, *assets*, etc.

Una vez enumerados los principales archivos que componen el *frontend* vamos a profundizar en los componentes, servicios y modelos.

6.1.2.1 Componentes

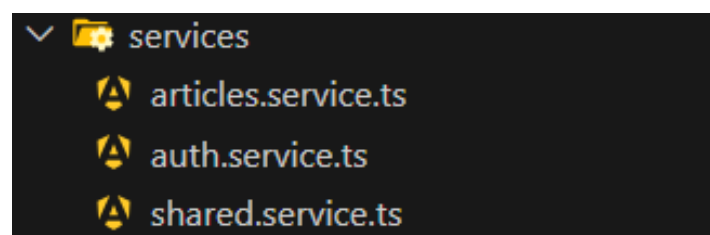
Este subdirectorio contiene todos los componentes utilizados en FACTOR, organizados en carpetas según el tipo de componente y cada uno con sus archivos *typescript*, *html* y *css*.

*Ilustración 42 - Componentes*

Como se puede observar en la captura anterior la aplicación sigue una estrategia de componentización que facilita la organización y mantenimiento del código. Algunos componentes, como los *pop-ups* o la barra de navegación, han sido diseñados para poder utilizados en todas las interfaces, por otro lado, el componente `articulo-list` también es reutilizado en múltiples páginas donde, además, las reglas específicas de *css* de cada componente lo adaptan a la interfaz según se necesite. Esto permite integrarlos fácilmente en diversas plantillas *html* mediante su selector. Otros, en cambio, están pensados para cumplir funciones específicas dentro de una única vista, como los formularios y el componente de perfil.

6.1.2.2 Servicios

Este subdirectorío contiene los tres servicios definidos en el proyecto, un servicio para la lógica de usuarios, otro para tratar la lógica de los artículos y por último un servicio *shared* que contiene la lógica compartida por varios componentes y que no requiere de llamadas a la API.

*Ilustración 43 - Servicios*

6.1.2.3 Modelos

El subdirectorio de modelos contiene las diferentes interfaces de datos utilizadas por el *frontend* con el fin de facilitar el tipado y controlar la información que se recibe de la API, así como donde se muestra.

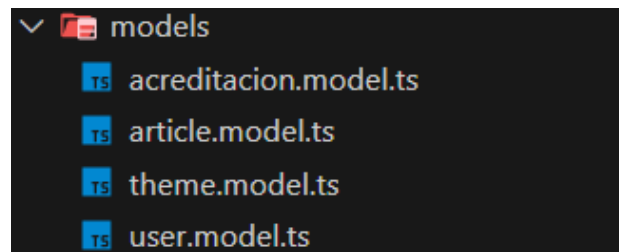


Ilustración 44 - Modelos Frontend

Un ejemplo representativo de esta implementación es el uso de dos interfaces distintas para los objetos de tipo usuario. La aplicación necesita acceder a toda la información del usuario autenticado, incluyendo datos sensibles que no deben ser visibles para otros. Por este motivo, se definieron dos interfaces diferenciadas: una destinada al usuario que ha iniciado sesión, con acceso completo a sus datos, y otra para representar al resto de usuarios, limitando la exposición de información privada.

```
import { Acreditacion } from './acreditacion.model';

export interface UserProfile {
  username: string;
  reputacion: number;
  nombre: string;
  apellidos: string;
  imagenPerfil?: string;
  email: string;
  fechaNacimiento: Date;
  acreditaciones: Acreditacion[]
}

export interface UserArticle {
  username: string;
  reputacion: number;
  nombre: string;
  apellidos: string;
  imagenPerfil?: string;
  acreditaciones: Acreditacion[]
}
```

Ilustración 45 -Modelo UserProfile y User.Article

6.2 Pruebas

En la siguiente sección se describen y analizan las pruebas llevadas a cabo para garantizar la calidad y correcto funcionamiento del sistema FACTOR. Para abordarlo, se ha dividido la fase de pruebas en pruebas de caja blanca y pruebas de caja negra.

6.2.1 Pruebas de caja blanca

Las pruebas de caja blanca se enfocan en la lógica interna del código fuente, comprobando las estructuras de control, condiciones, caminos de ejecución y el resto de la lógica interna.

A continuación, se describen las diferentes pruebas de caja blanca realizadas en la fase de pruebas:

1. Validación de *cookies* y gestión de sesiones: Se verifica el correcto funcionamiento de las *cookies* y el *token* de sesión, evaluando los tiempos de expiración y que el usuario no pueda modificar manualmente el token.
2. Validación de control de acceso a rutas protegidas: Se comprueban las restricciones de acceso a rutas protegidas a usuarios autenticados y no autenticados
3. Validación de conexión con la base de datos.
4. Prueba de creación de datos: Se verifica la correcta creación de los distintos tipos de datos.
5. Prueba de eliminación de datos: Se verifican los mecanismos de eliminación de datos, así como del resto de archivos asociados (documentos e imágenes del servidor).
6. Prueba de actualización de datos: Se verifica la correcta actualización de las entidades en la base de datos.
7. Prueba de seguridad e integridad en la API: Se verifica que las APIs operan únicamente con los datos necesarios.
8. Pruebas de autenticación: Se verifica el correcto flujo y comportamiento de los procesos de registro e inicio de sesión.

9. Pruebas de automatización: Se verifica el correcto comportamiento automático del sistema a la hora de categorizar un artículo cuando su *pool* de votos llega al límite, así como los cálculos de los diferentes parámetros de veracidad y la asignación de recompensas a los usuarios autores y votantes.
10. Manejo de excepciones y errores: Se verifican las diferentes excepciones tratadas en el sistema y se comprueban los mensajes de éxito o error.

6.2.2 Pruebas de caja negra

Las pruebas de caja negra, a diferencia de las de caja blanca se enfocan a validar comportamiento externo del sistema, abstrayéndose de su implementación interna.

A través de estas pruebas se evalúan las respuestas del sistema a diferentes entradas, satisfaciendo los requisitos funcionales analizados desde el punto de vista del usuario.

En las siguientes tablas se detallan las pruebas de caja negra realizadas, así como su resultado.

PCN-01 Crear cuenta de usuario	
Objetivo	Crear una nueva cuenta de usuario con los datos introducidos.
Precondiciones	N/A
Datos de entrada	Nombre, apellidos, email y contraseña
Resultado esperado	Mensaje “Registro exitoso” (201).
Resultado obtenido	Correcto

Tabla 62 - PCN-01 Crear cuenta de usuario

PCN-02 Inicio de sesión	
Objetivo	Ingresar en la aplicación con las credenciales del usuario.
Precondiciones	Las credenciales introducidas deben corresponder con una cuenta registrada en el sistema
Datos de entrada	Email, contraseña
Resultado esperado	Mensaje “Inicio de sesión exitoso” (200).
Resultado obtenido	Correcto

Tabla 63- PCN-02 - Inicio de Sesión

PCN-03 Verificar sesión	
Objetivo	Verificar la <i>cookie</i> de sesión del usuario.
Precondiciones	El usuario debe haber iniciado sesión.
Datos de entrada	Cookie de sesión.
Resultado esperado	Información de usuario (200).
Resultado obtenido	Correcto

Tabla 64 - PCN-03 - Verificar sesión

PCN-04 Listar artículos del <i>feed</i>	
Objetivo	Listar los artículos del feed.
Precondiciones	El usuario debe estar autenticado.
Datos de entrada	Cookie de sesión.
Resultado esperado	Listado de artículos (200).
Resultado obtenido	Correcto

Tabla 65 - PCN-04 - Listar artículos del feed

PCN-05 Listar temáticas del <i>feed</i>	
Objetivo	Listar las temáticas del feed.
Precondiciones	El usuario debe estar autenticado.
Datos de entrada	Cookie de sesión.
Resultado esperado	Listado de temáticas (200).
Resultado obtenido	Correcto

Tabla 66 - PCN-05 - Listar temáticas del feed

PCN-06 Ver temática	
Objetivo	Buscar artículos por temática del feed.
Precondiciones	El usuario debe estar autenticado.
Datos de entrada	Cookie de sesión.

Resultado esperado	Al presionar una temática el sistema redirige al usuario a la página explorador habiendo filtrado por la temática seleccionada y ordenado por defecto por fecha de publicación (200).
Resultado obtenido	Correcto

Tabla 67 - PCN-06 - Ver temática

PCN-07 Visualizar detalle artículo	
Objetivo	Ver detalle de un artículo.
Precondiciones	El usuario debe estar autenticado.
Datos de entrada	Cookie de sesión.
Resultado esperado	Al presionar el botón ver de un artículo el sistema redirige al usuario a la página del detalle del artículo (200).
Resultado obtenido	Correcto

Tabla 68 - PCN-07 - Visualizar detalle artículo

PCN-08 Visualizar perfil	
Objetivo	Ver perfil de un usuario.
Precondiciones	El usuario debe estar autenticado.
Datos de entrada	Cookie de sesión.
Resultado esperado	Al presionar el botón ver perfil de un artículo el sistema redirige al usuario a la página del perfil del autor (200).
Resultado obtenido	Correcto

Tabla 69 - PCN-08 - Visualizar perfil

PCN-09 Listar actividad perfil	
Objetivo	Lista la actividad de un usuario.
Precondiciones	El usuario debe estar autenticado.
Datos de entrada	Cookie de sesión.
Resultado esperado	Listado de artículos (200).
Resultado obtenido	Correcto

Tabla 70 - PCN-09 - Listar actividad perfil

PCN-10 Listar artículos por autor	
Objetivo	Lista los artículos de un usuario determinado.
Precondiciones	El usuario debe estar autenticado.
Datos de entrada	Cookie de sesión.
Resultado esperado	Listado de artículos (200).
Resultado obtenido	Correcto

Tabla 71 - PCN-10 - Listar artículos por autor

PCN-11 Subir un artículo	
Objetivo	Publicar un artículo.
Precondiciones	El usuario debe disponer de permisos para poder publicar y estar autenticado.
Datos de entrada	Título, descripción, fuente, temática y archivo pdf.
Resultado esperado	Mensaje “Artículo creado con éxito” (201).
Resultado obtenido	Correcto

Tabla 72 - PCN-11 - Subir un artículo

PCN-12 Eliminar un artículo	
Objetivo	Eliminar un artículo.
Precondiciones	El usuario debe ser el autor del artículo a eliminar y debe estar autenticado.
Datos de entrada	Cookie de sesión.
Resultado esperado	Mensaje “Artículo eliminado con éxito” (200).
Resultado obtenido	Correcto

Tabla 73 - PCN-12 - Eliminar un artículo

PCN-13 Eliminar un artículo	
Objetivo	Eliminar un artículo.
Precondiciones	El usuario debe ser el autor del artículo a eliminar y debe estar autenticado.
Datos de entrada	Cookie de sesión.

Resultado esperado	Mensaje “Artículo eliminado con éxito” (200).
Resultado obtenido	Correcto

Tabla 74 - PCN-013 - Eliminar un artículo

PCN-14 Votar un artículo	
Objetivo	Votar un artículo.
Precondiciones	El usuario debe tener permisos de votación, no haber votado ya ese artículo y debe estar autenticado.
Datos de entrada	Cookie de sesión.
Resultado esperado	El usuario visualizará en tiempo real la repercusión de su voto en la puntuación del artículo y el mensaje “Voto registrado correctamente” (200).
Resultado obtenido	Correcto

Tabla 75 - PCN-14 - Votar un artículo

PCN-15 Explorar artículos	
Objetivo	Buscar artículos en el explorador.
Precondiciones	El usuario debe estar autenticado.
Datos de entrada	Temática y/o fecha o temática y/o puntuación.
Resultado esperado	El usuario pulsará el botón “Buscar” y verá los resultados de su búsqueda (200).
Resultado obtenido	Correcto

Tabla 76 - PCN-15 - Explorador artículos

PCN-16 Editar perfil	
Objetivo	El usuario puede modificar la información de su perfil.
Precondiciones	El usuario debe estar autenticado.
Datos de entrada	Nombre, apellidos, fecha de nacimiento e imagen de perfil.
Resultado esperado	El usuario guardará los cambios y visualizará el mensaje “Perfil actualizado correctamente” (200).
Resultado obtenido	Correcto

Tabla 77 - PCN-16 - Editar perfil

PCN-17 Añadir acreditación	
Objetivo	El usuario puede añadir una o varias acreditaciones.
Precondiciones	El usuario debe estar autenticado.
Datos de entrada	Título, institución y año.
Resultado esperado	El usuario guardará los cambios y visualizará el mensaje “Perfil actualizado correctamente” (200).
Resultado obtenido	Correcto

Tabla 78 - PCN-01 - Añadir acreditación

PCN-18 Eliminar acreditación	
Objetivo	El usuario puede eliminar una o varias acreditaciones.
Precondiciones	El usuario debe estar autenticado.
Datos de entrada	Cookie de sesión.
Resultado esperado	El usuario guardará los cambios y visualizará el mensaje “Perfil actualizado correctamente” (200).
Resultado obtenido	Correcto

Tabla 79 - PCN-18 - Eliminar acreditación

PCN-19 Consultar privilegios	
Objetivo	El usuario puede visualizar sus privilegios.
Precondiciones	El usuario debe estar autenticado.
Datos de entrada	Cookie de sesión.
Resultado esperado	Votos realizados esta semana/ límite votos por semana y publicaciones realizadas ese mes/ límite publicaciones por mes (200).
Resultado obtenido	Correcto

Tabla 80 - PCN-19 - Consultar privilegios

PCN-20 Cerrar sesión	
Objetivo	El usuario termina la sesión abierta.
Precondiciones	El usuario debe estar autenticado.
Datos de entrada	Cookie de sesión.

Resultado esperado	Redirección a la página de inicio de sesión y mensaje “Sesión terminada con éxito” (200).
Resultado obtenido	Correcto

Tabla 81 - PCN-20 - Cerrar sesión

PCN-21 Eliminar cuenta con artículos	
Objetivo	El usuario elimina su cuenta y toda la información asociada.
Precondiciones	El usuario debe estar autenticado.
Datos de entrada	Cookie de sesión.
Resultado esperado	Redirección a la página de inicio de sesión y mensaje “Cuenta eliminada con éxito” (200).
Resultado obtenido	Correcto

Tabla 82 - PCN-21 - Eliminar cuenta con artículos

PCN-22 Eliminar cuenta sin artículos	
Objetivo	El usuario elimina su cuenta manteniendo sus artículos en la aplicación.
Precondiciones	El usuario debe estar autenticado y tener artículos publicados.
Datos de entrada	Cookie de sesión.
Resultado esperado	Redirección a la página de inicio de sesión y mensaje “Cuenta eliminada con éxito” (200). Si se inicia sesión en la aplicación con otro usuario se podrán visualizar los artículos del usuario eliminado y un mensaje “El autor ha eliminado su cuenta”.
Resultado obtenido	Correcto

Tabla 83 - PCN-22 – Eliminar cuenta sin artículos

Capítulo 7: Conclusiones y propuestas de mejora

En la siguiente sección se presentan las conclusiones obtenidas tras el desarrollo del proyecto FACTOR, así como las futuras líneas de mejora y ampliaciones.

7.1 Conclusiones

El desarrollo de este proyecto ha supuesto un reto, así como la materialización de todo lo aprendido a lo largo del grado. Desde el primer momento, busqué una idea que de verdad despertara mi interés y a la que pudiese aportar mi propia visión, para posteriormente darle forma a través de todas las habilidades adquiridas estos últimos años: desde la planificación del proyecto hasta la implementación, pasando por la especificación de requisitos, diseño de los algoritmos o las pruebas realizadas.

Con respecto al resultado, me encuentro muy satisfecho. Mi objetivo era crear una plataforma dotada de unos mecanismos para la divulgación y verificación de contenido, gestionada por la propia comunidad, y sin ninguna otra intervención. Considero que la aplicación final cumple con esta visión.

Además, uno de los aspectos más enriquecedores del proyecto ha sido la necesidad de reflexionar no solo sobre el desarrollo técnico del sistema, sino también sobre cómo diseñar mecanismos que permitan otorgar veracidad a los usuarios y a sus publicaciones, así como establecer métodos para aumentar y distribuir dicha veracidad. El proceso de conceptualización de estos mecanismos antes de abordar la implementación, junto con la variedad de tecnologías aplicadas, han sido, sin duda, los elementos más satisfactorios de todo el proyecto.

7.2 Propuestas de mejora

A continuación, se describen una serie de propuestas de mejora identificadas para una posible evolución futura del sistema. Estas mejoras surgen en parte debido a las limitaciones propias de un Trabajo de Fin de Grado, especialmente en lo referente al tiempo. Algunas de las posibles ampliaciones o mejoras detectadas son las siguientes:

- **Parámetros dinámicos:** Debido a la cantidad y complejidad de los parámetros existentes, la versión actual de FACTOR utiliza valores estáticos. En una futura evolución, sería conveniente que estos parámetros pudieran ajustarse dinámicamente en función del número de usuarios registrados en la plataforma, adaptando así el comportamiento del sistema y mejorando la escalabilidad.
- **Suscripción a temáticas:** Otra ampliación importante sería una mejora en el sistema de temáticas, permitiendo la opción a los usuarios de suscribirse a aquellas que sean de su interés y recibiendo en el *feed* mayoritariamente artículos de dichas temáticas.
- **Suscripción a autores:** Siguiendo la línea de personalización del contenido, sería interesante incorporar la posibilidad de que los usuarios puedan seguir a sus autores favoritos, visualizando en su *feed* las últimas publicaciones de estos.
- **Amigos:** Con el objetivo de reforzar el enfoque de red social de la plataforma, podría implementarse un sistema de amistad, mediante el cual los usuarios puedan establecer contactos, visualizar de manera más directa las interacciones de sus amigos fomentando una mayor interconexión dentro de la comunidad.
- **Verificación de perfiles:** Una mejora en la línea de la fiabilidad y confianza de la aplicación, se propone la implementación de un sistema de verificación de perfiles, que permita a los usuarios acreditar sus titulaciones o experiencia, fortaleciendo así la credibilidad de sus publicaciones.
- **Integración con redes sociales externas:** Facilitar la opción de compartir artículos directamente en otras redes sociales como *Twitter* o *LinkedIn* podría aumentar la visibilidad del contenido y potenciar la adopción de FACTOR.

Webgrafía

- 1) IMB. (n.d.). *¿Qué es la pila MEAN?* IBM. <https://www.ibm.com/es-es/think/topics/mean-stack>
- 2) Express.js. (n.d.). *Express routing*. Express Documentation. <https://expressjs.com/en/guide/routing.html>
- 3) Intelequia. (2021). *Cómo subir un proyecto a un repositorio de GitHub*. Intelequia. <https://intelequia.com/es/blog/post/c%C3%B3mo-subir-un-proyecto-a-un-repositorio-de-github>
- 4) Esther Vaati. (2022). *Subir archivos con Multer en Node.js y Express*. Envatotuts. <https://code.tutsplus.com/es/file-upload-with-multer-in-node--cms-32088t>
- 5) CodingPotions. (2018). *Angular - Seguridad, protegiendo vistas con guards*. CodingPotions. <https://codingpotions.com/angular-seguridad/>
- 6) Desarrolloweb. (2020). *Servicios en Angular*. Desarrolloweb. <https://desarrolloweb.com/articulos/servicios-angular.html>
- 7) Esther Vaati. (2018). *Autenticación Angular Con JWT*. Envatotuts. <https://code.tutsplus.com/es/jwt-authentication-in-angular--cms-32006t>
- 8) OMG. (2014). *Automated Function Points (AFP)*. Object Management Group. <https://www.omg.org/spec/AFP/1.0/PDF>
- 9) Carlos Eduardo Vázquez. (n.d.). *Estimación de Software con COSMIC*. SG. <https://sg.com.mx/revista/49/estimacion-software-cosmic>
- 10) Talent. (2025). *Salario medio para Desarrollador en España, 2025*. Talent. <https://es.talent.com/salary?job=desarrollador>
- 11) Codezup (2025). *Angular and MongoDB: A Practical Guide to Integrating NoSQL Data with Your App*. Codezup. <https://codezup.com/angular-mongodb-integration-guide/>
- 12) Xataka. (2025). *Las notas de la comunidad llegan a Twitter. Podrás ponerle la "puntilla" a un tweet, con ciertas normas*. Xataka. <https://www.xataka.com/aplicaciones/notas-comunidad-llegan-a-twitter-podras-ponerle-puntilla-a-tweet-ciertas-normas>

- 13) Reddit Help. (2025). *¿Qué es el karma?* RedditHelp. <https://support.reddithelp.com/hc/es-es/articles/204511829--Qu%C3%A9-es-el-karma>
- 14) Abiztar (n.d.). *Casos a incluir y casos a extender*. Abiztar. <https://www.abiztar.com.mx/articulos/casos-a-incluir-casos-a-extender.html>
- 15) PMOInformatica (2017). *Pruebas de caja negra: Ejemplos*. PMOInformatica. <https://www.pmoinformatica.com/2017/02/pruebas-de-caja-negra-ejemplos.html>

Anexo 1 Manual de Instalación

En esta sección se detallan todos los pasos a seguir para realizar la correcta instalación de la aplicación FACTOR. Para ello se proporciona el código fuente alojado en GITHUB.

- **Instalación del software necesario para la ejecución de la aplicación:** Se recomienda la instalación del software siempre desde páginas oficiales, para la ejecución de FACTOR sería requerido la instalación de *Node.js*, *Git* y *AngularCLI*.
- **Instalación del software de FACTOR alojado en el repositorio:** El usuario debe crear un directorio donde quiera que se realice la clonación del código fuente y se sitúa en el terminal en dicho directorio:

```
$ cd C:\Directorio_Creado
```

A continuación, se introduce el comando para realizar la clonación del repositorio:

```
$ git clone https://github.com/Mario-Olmos/Factor.git
```

- **Instalación de módulos y dependencias:** Una vez realizada la clonación, el usuario debe situarse en el directorio raíz del proyecto:

```
$ cd C:\Directorio_Creado\Factor
```

Una vez allí deberá instalar los diferentes módulos y dependencias tanto del *backend* como del *frontend*:

```
$ cd Front
```

```
$ npm install
```

```
$ cd Back
```

```
$ npm install
```

- **Despliegue de la aplicación:** Una vez concluidos los pasos anteriores, el usuario podrá desplegar la aplicación con los siguientes pasos:

1. En el directorio del *frontend* (C:\Directorio_Creado\Factor\Front) ejecutar en la terminal el siguiente comando:

```
$ ng serve
```

2. En el directorio del *backend* (C:\Directorio_Creado\Factor\Back) ejecutar en la terminal el siguiente comando:

```
$ node app.js
```

3. Por último, el usuario deberá introducir la siguiente URL en el navegador accediendo a la aplicación: <http://localhost:4200/>

Anexo 2 Manual de usuario

En este anexo se detalla el manual de uso para el uso de la aplicación, abarcando todas las funcionalidades que puede realizar un usuario.

Página registro:

FACTOR

Iniciar Sesión

Registrarse

Registro de Usuario

Nombre

Apellidos

Email

Contraseña

dd/mm/aaaa

Registrar

[¿Ya tienes una cuenta? Inicia sesión aquí](#)

Ilustración 46 - MU - Página de Registro

El primer paso que deber realizar un nuevo usuario en la aplicación es crearse una cuenta, para ello debe introducir todos los campos del formulario, ser mayor de edad, introducir un email no registrado en la aplicación e introducir una contraseña con más de 8 caracteres, una mayúscula, una minúscula y un carácter especial.

Después de registrarse en la aplicación el sistema lo redirigirá a la página de inicio de sesión, donde deberá introducir sus credenciales:

Página inicio de sesión:

FACTOR

Iniciar Sesión
Registrarse

Inicio de Sesión

Email

Contraseña

Iniciar Sesión

[¿No tienes una cuenta? Regístrate aquí](#)

Ilustración 47 - MU - Página de Inicio de Sesión

Una vez ingresamos las credenciales el sistema verificará si son correctos, ingresando en la aplicación, de otra manera nos mostrará un mensaje indicando que las credenciales no son válidas.

Una vez el usuario ha sido autenticado el sistema lo envía a la página principal, llamada *home*. En esta página se muestran las temáticas y artículos en tendencia:

Página home – temáticas:*Ilustración 48 - Página Home - Temáticas***Página home – artículos:***Ilustración 49 - Página Home - Artículos*

Volviendo a la parte de las temáticas, el usuario puede clicar en la tarjeta que le interese, el sistema le llevará entonces a la página del buscador donde el usuario podrá ver los últimos artículos de esa temática, además de poder filtrar por otras temáticas y ordenar los resultados por puntuación o por fecha de publicación:

Página explorador – Buscar temática desde el home:

Cerrar Sesión

Temática: • Aprendizaje Supervisado y No Supervisado

Ordenar por Puntuación: Ninguno

Ordenar por Fecha: Ninguno

Buscar

Explorador de Artículos

5.0

SIN EVALUAR

IA aplicada a modelos de rotación de jugadores en equipos de fútbol

Este trabajo se centra en evaluar diversas técnicas de inteligencia artificial para predecir el rendimiento de equipos de fútbol mediante el análisis de la rotación de jugadores. El objetivo del proyecto es detectar cómo las diferentes estrategias de rotación aplicadas por los equipos afectan al desempeño del equipo y cómo la inteligencia artificial puede realizar predicciones en base a ellas para ayudar a optimizar estas estrategias y detectar cuáles son las mejores. Este trabajo abarca desde la recopilación y el análisis de datos asociados a las ligas seleccionadas, la creación de modelos de inteligencia artificial y la evaluación de su eficacia. Se pretende que con este proyecto se puedan obtener resultados significativos para optimizar la gestión de equipos de fútbol y así poder facilitar el trabajo a sus dirigentes.

Inteligencia Artificial | Aprendizaje Automático (Machine Learning) | Aprendizaje Supervisado y No Supervisado

Javier Navarro Díaz

INICIADO

Fecha de publicación: May 1, 2025

Ver perfil

Ilustración 50 - Página Explorador - Buscar Temática desde el Home

Página explorador – Temáticas y parámetros de ordenación

FACTOR

Home

Explorador

Subir Artículo

Perfil

Cerrar Sesión

Temática: Ingeniería de Hardware

Ordenar por Puntuación: Puntuación 1

Ordenar por Fecha: Ninguno

Buscar

Inteligencia Artificial

- Aprendizaje Automático (Machine Learning)
 - Aprendizaje Supervisado y No Supervisado
 - Redes Neuronales Artificiales
 - Aprendizaje Profundo (Deep Learning)
 - Modelos Generativos (GANs)
- Procesamiento del Lenguaje Natural (NLP)
 - Análisis de Sentimientos
 - Traducción Automática
 - Chatbots y Asistentes Virtuales
- Visión por Computador
 - Reconocimiento de Imágenes
 - Detección de Objetos
 - Visión Artificial para Automóviles Autónomos
- Sistemas Expertos
 - Sistemas Basados en Reglas

Explorador de Artículos

mediante RL para neuroestimuladores inteligentes

Las sanitarias ha supuesto la utilización de nuevos métodos de tratamiento para las enfermedades como pueden ser el Parkinson o la epilepsia. Estos trastornos afectan gravemente a la calidad de vida de los pacientes que producen de forma incremental temblores, rigidez muscular y dificultad para realizar actividades cotidianas. Para estos síntomas, se ha desarrollado una técnica neuroquirúrgica llamada Estimulación

Ilustración 51 - Página Explorador - Temáticas y Parámetros de Ordenación

Estas dos páginas conforman los mecanismos de *feed* y de búsqueda de la aplicación, en cualquiera de ellas se desplegarán los artículos registrados en la aplicación.

En cuanto a las interacciones con los artículos, los usuarios podrán votarlos si no lo han hecho ya y si disponen de los privilegios para ello, además de poder visualizar el detalle del artículo y el perfil de su autor:

Artículo:

5.0

SIN EVALUAR

Agentes inteligentes en videojuegos: Bloons TD 6

La Inteligencia Artificial es una herramienta vital para los videojuegos desde los inicios de este medio de entretenimiento en la década de los 70. Pero tanto la Inteligencia Artificial como los videojuegos han sufrido una gran evolución desde entonces. Por ello, en este proyecto se desarrolla un sistema de Inteligencia Artificial por Aprendizaje por Refuerzo Profundo capaz de jugar al videojuego Bloons Tower Defense 6, de Ninja Kiwi. Este sistema, denominado agente inteligente, es capaz de leer el estado de la partida, y a partir de él procesar cuál es la acción más adecuada para realizar a continuación. Al utilizar Aprendizaje por Refuerzo Profundo (algoritmo PPO), el agente recibe premios y castigos según las acciones que realiza, en forma de puntuación,

Inteligencia Artificial Aprendizaje Automático (Machine Learning) Aprendizaje Profundo (Deep Learning)

Ver

0 0

Daniel Martínez Rubio

INICIADO

Fecha de publicación: Apr 27, 2025

Ver perfil

Ilustración 52 - Vista Artículo

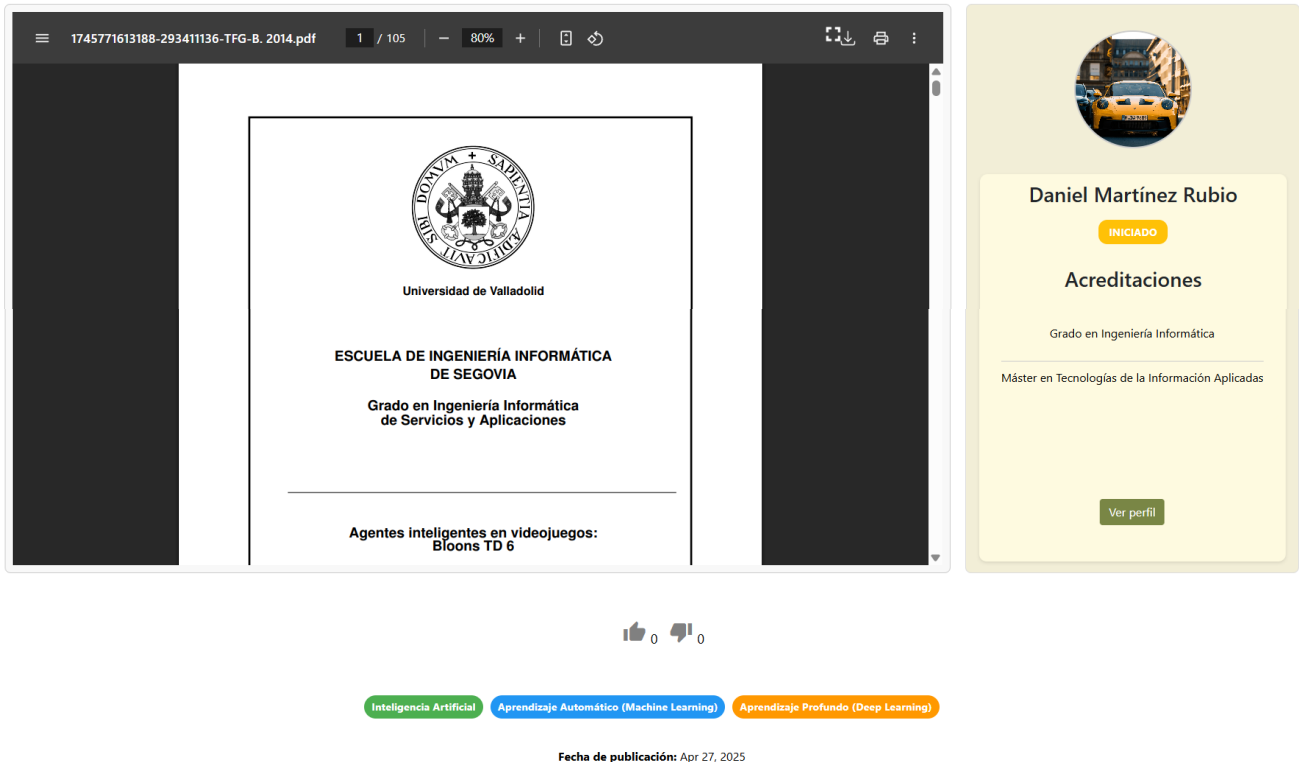
Página detalle de artículo:

SIN EVALUAR

5.0

Agentes inteligentes en videojuegos: Bloons TD 6

La Inteligencia Artificial es una herramienta vital para los videojuegos desde los inicios de este medio de entretenimiento en la década de los 70. Pero tanto la Inteligencia Artificial como los videojuegos han sufrido una gran evolución desde entonces. Por ello, en este proyecto se desarrolla un sistema de Inteligencia Artificial por Aprendizaje por Refuerzo Profundo capaz de jugar al videojuego Bloons Tower Defense 6, de Ninja Kiwi. Este sistema, denominado agente inteligente, es capaz de leer el estado de la partida, y a partir de él procesar cuál es la acción más adecuada para realizar a continuación. Al utilizar Aprendizaje por Refuerzo Profundo (algoritmo PPO), el agente recibe premios y castigos según las acciones que realiza, en forma de puntuación,

*Ilustración 53 - Página Detalle Artículo*

En la página detalle del artículo encontramos la información detallada, donde también podemos votar positiva o negativamente, ver el perfil del autor y consultar la fuente original del artículo.

En esta página encontramos el visor del artículo completo, con funcionalidad para ampliarlo a pantalla completa y descargarnos el artículo.

Página perfil no personal- ventana información:

En caso de querer consultar el perfil de un autor, se puede hacer mediante el botón “Ver Perfil” en un artículo suyo. En este caso, el sistema nos mostrará el perfil del autor con la información esencial y dos pestañas, en la de información encontraremos las acreditaciones del autor, además de los artículos con los que ha interactuado:

Alejandro Torres Sánchez
INICIADO

Información Artículos

Acreditaciones

Título: Doctorado en Inteligencia Artificial
Institución: Universidad de Stanford
Año: 2020

Título: Profesor Titular en Ciencias de la Computación
Institución: Universidad Autónoma de Madrid
Año: 2023

Artículos votados por Alejandro Torres Sánchez

6.7
SIN EVALUAR

Algoritmo de control mediante RL para neuroestimuladores inteligentes

El creciente avance de las tecnologías sanitarias ha supuesto la utilización de nuevos métodos de tratamiento para las enfermedades del sistema nervioso, como pueden ser el Parkinson o la epilepsia. Estos trastornos afectan gravemente a la calidad de vida de los pacientes, ya que producen de forma incremental temblores, rigidez muscular y dificultad para moverse. Con la finalidad de reducir estos síntomas, se ha desarrollado una técnica neuroquirúrgica llamada Estimulación cerebral profunda. Este método consiste en enviar, mediante unos electrodos implantados en el cerebro, pulsos eléctricos a las zonas causantes de los temblores y convulsiones, reduciendo así su frecuencia e intensidad a largo plazo. Sin embargo, en la actualidad, estos tratamientos no se adaptan de forma dinámica al estado del paciente, ya que los parámetros del dispositivo de control deben ser ajustados manualmente por los sanitarios.

Inteligencia Artificial Sistemas Expertos Razonamiento Automático

1 0 Ver

Ilustración 54 - Página Perfil no Personal - Información

Página perfil no personal - ventana artículos:

Por otro lado, si clicamos en la ventana de artículos, nos mostrará todos los artículos publicados por ese autor:



Ilustración 55 - Página Perfil no Personal - Artículos

Otra funcionalidad que puede realizar el usuario consiste en dar de alta un artículo, tanto el acceso a la página de subida como el del perfil personal se encuentran en la barra de navegación, la cual muestra estas opciones al momento de iniciar sesión:



Bienvenido a FACTOR

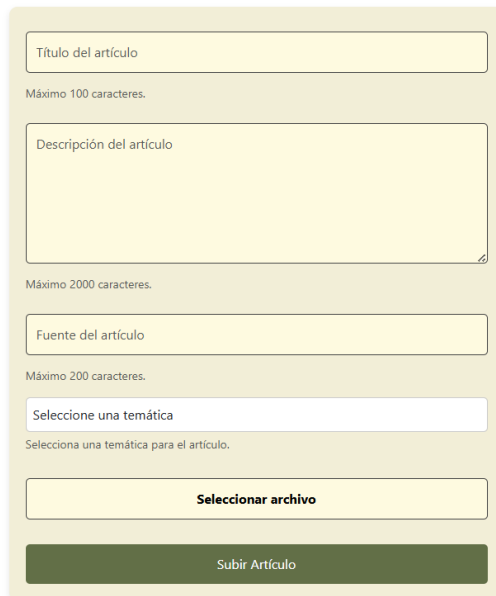
Temas Populares

Ilustración 56 - Navbar

Página subir artículo:

En esta página el usuario introduce los datos del artículo a subir, debe rellenar todo el formulario e introducir los datos permitidos. Por último, el sistema evaluará si el usuario dispone de permisos para publicar un nuevo artículo

Subir un Nuevo Artículo

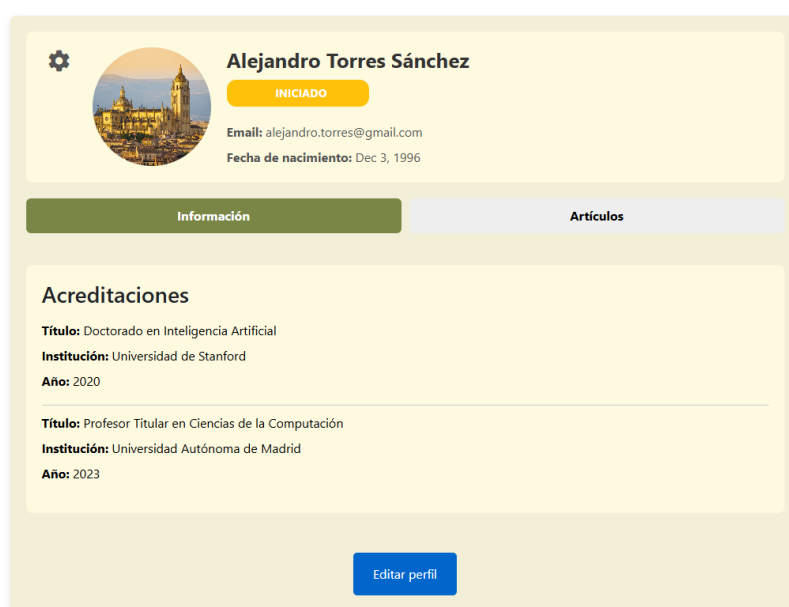


Formulario para subir un nuevo artículo. El formulario está dividido en secciones con campos de texto y botones. La primera sección es 'Título del artículo' con un campo de texto y una etiqueta 'Máximo 100 caracteres.'. La segunda sección es 'Descripción del artículo' con un campo de texto grande y una etiqueta 'Máximo 2000 caracteres.'. La tercera sección es 'Fuente del artículo' con un campo de texto y una etiqueta 'Máximo 200 caracteres.'. La cuarta sección es 'Seleccione una temática' con un menú desplegable y una etiqueta 'Selecciona una temática para el artículo.'. La quinta sección es 'Seleccionar archivo' con un botón. La sexta sección es 'Subir Artículo' con un botón.

Ilustración 57 - Página Subir Artículo

Página perfil personal -información:

En la página de perfil personal el usuario encontrará más información que en un perfil ajeno, tendrá accesible además una ruleta con diferentes acciones y en la pestaña de información encontrará un botón para editar su información personal:



Perfil personal de Alejandro Torres Sánchez. El perfil muestra un icono de configuración, un botón 'INICIADO', un correo electrónico 'alejandro.torres@gmail.com' y una fecha de nacimiento 'Dec 3, 1996'. Hay dos pestañas: 'Información' (seleccionada) y 'Artículos'. La pestaña 'Información' muestra 'Acreditaciones' con dos entradas: 'Título: Doctorado en Inteligencia Artificial', 'Institución: Universidad de Stanford', 'Año: 2020' y 'Título: Profesor Titular en Ciencias de la Computación', 'Institución: Universidad Autónoma de Madrid', 'Año: 2023'. Hay un botón 'Editar perfil'.

Ilustración 58 - Página Perfil Personal - Información

Página perfil personal – menú de edición:

Nombre:
Alejandro

Apellidos:
Torres Sánchez

Fecha de nacimiento:
03/12/1996

Seleccionar imagen de perfil:

Acreditaciones

Título:
Doctorado en Inteligencia Artificial

Institución:
Universidad de Stanford

Año:
2020

Eliminar

Título:
Profesor Titular en Ciencias de la Computación

Institución:
Universidad Autónoma de Madrid

Año:
2023

Eliminar

Añadir acreditación

Guardar Cancelar

Artículos votados por Alejandro Torres Sánchez

Ilustración 59 - Página Perfil Personal - Menú Edición

Página perfil personal – menú de artículos:

En el perfil personal, en la pestaña de artículos también encontramos la opción de eliminar un artículo, a diferencia del perfil ajeno:



Ilustración 60 - Página Perfil Personal - Artículos

Perfil personal – acciones:

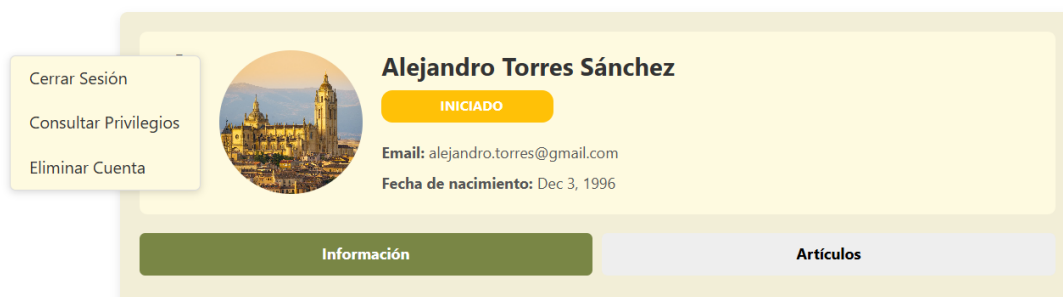


Ilustración 61- Perfil Personal - Acciones

Perfil personal – consultar privilegios:

El usuario puede consultar en su perfil personal el número de votos que puede realizar a la semana y la cantidad de artículos que puede publicar mensualmente. Además de los votos y publicaciones realizadas en la semana/mes actual:

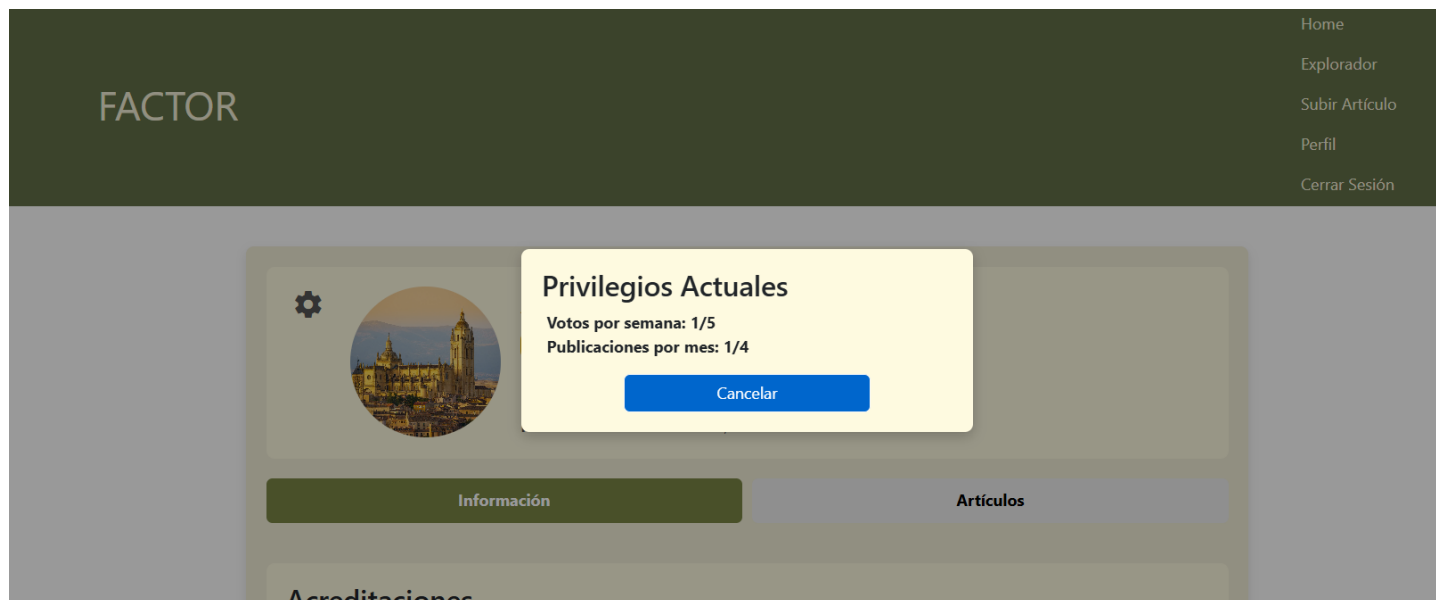


Ilustración 62 - Perfil Personal - Consultar Privilegios

Perfil personal – eliminar cuenta:

Por último, el usuario puede eliminar su cuenta. En este caso el sistema le preguntará si desea mantener sus artículos en la aplicación o eliminarlos a la vez que su cuenta:

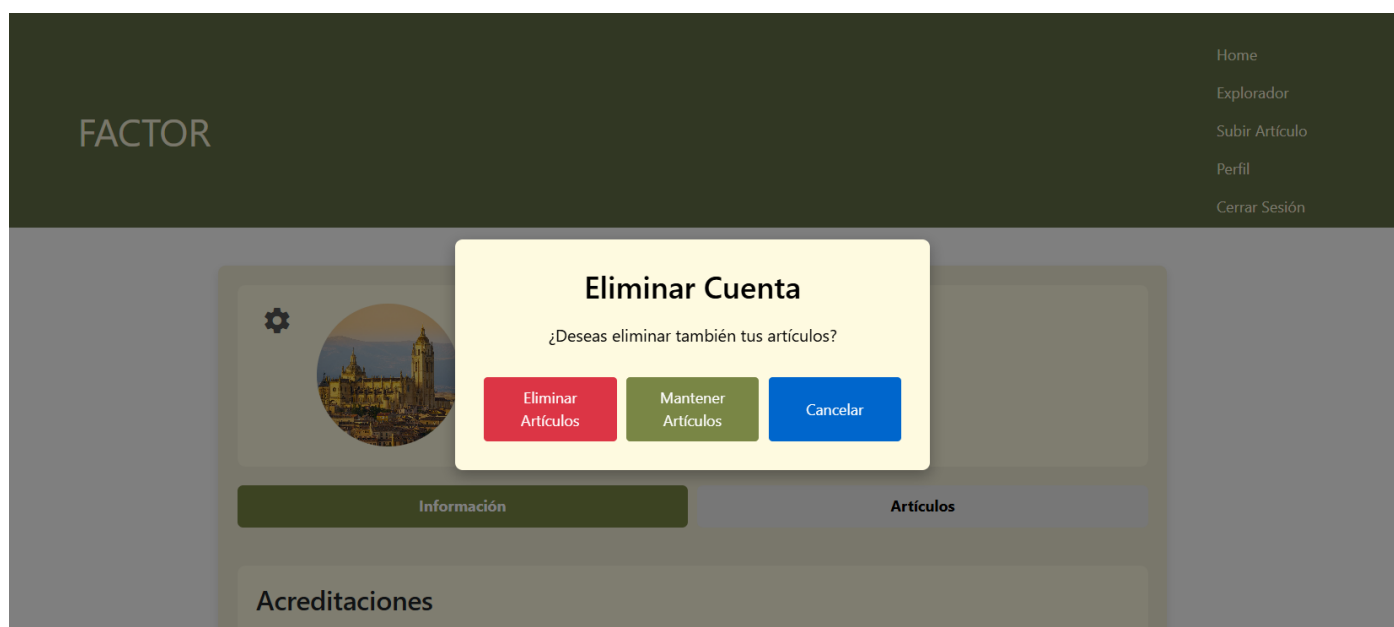


Ilustración 63 - Perfil Personal - Eliminar Cuenta

