



Trabajo Fin de Grado
Una wiki para regiones delimitadas sobre una malla global discreta
Autor
Diego Raúl Roldán Urueña
Director
Rubén Béjar Hernández
Grado en Ingeniería Informática
Escuela de Ingeniería y Arquitectura
Julio 2025

1 Resumen

Obligatorio según la normativa (revisar). Puede ser algo como Introducción + Conclusiones condensadas en, pongamos, 300 palabras.

2 Tabla de contenidos

Tabla de contenido

1	Resumen.....	2
2	Tabla de contenidos o Índice.....	2
3	Introducción	4
3.1	Contexto.....	4
3.2	Fuentes de datos integrados / mostrados / procesados	5
4	Análisis del problema	6
4.1	Requisitos / Historias de Usuario / Casos de Uso	6
4.2	Casos de uso.....	11
4.2.1	Ver un artículo.....	11
4.2.2	Ver versiones anteriores de un artículo	11
4.2.3	Iniciar Sesión	12
4.2.4	Crear un artículo nuevo	12
4.2.5	Crear un artículo a partir de otro	13
4.2.6	Editar un articulo	13
4.3	Interfaces de usuario.....	14
4.4	Interfaces para la integración en aplicaciones	17
5	Diseño de la solución	19
5.1	Arquitectura.....	19
5.1.1	Contexto.....	20
5.1.2	Modelo de datos	20
5.1.3	Paquetes y clases.....	22
5.1.4	Componentes y conectores	23
5.1.5	Distribución	23
5.2	Implementación.....	23
5.3	Procesamiento de datos.....	26
5.4	Pruebas	26
6	Gestión del proyecto	27
7	Plan de producto / plan de negocio / análisis de mercado	28

8	Conclusiones.....	29
9	Bibliografía	30
10	Anexos.....	31

2.1 Tabla de figuras

Figura 1. Página principal de Locatopedia.....	6
Figura 2. Locatopedia en ordenador.....	14
Figura 3. Locatopedia en móvil.	15
Figura 4. Locatopedia en tablet.....	15
Figura 5. Primera versión del mapa de navegación de la aplicación.....	16
Figura 6. Diagrama Entidad-Relación.	21
Figura 7. Diagrama de paquetes.....	22
Figura 8. Diagrama de clases.	22
Figura 9. Mapa de navegación.....	25

3 Introducción

Uno o dos párrafos de introducción al trabajo: brevemente qué se quería hacer, y cuáles han sido los principales resultados alcanzados. El resto del documento se estructura como sigue: explicar la estructura del documento. Puede ser buena idea señalar que se ha cumplido explícitamente con la recomendación de 20 páginas/10.000 palabras, o explicar explícitamente por qué no se ha hecho.

Este documento se basa en el desarrollo de una aplicación web. Se ha creado un sitio al estilo 'wiki' donde cada artículo se identifica por medio de un lugar o un conjunto de lugares. Cada lugar, es una celda definida sobre una malla global discreta (DGG). De esta forma, cada conjunto de celdas recibe un identificador único que será el que identifique cada artículo y se usa para asignarle un PURL (URL permanente).

El objetivo principal es que cualquier usuario pueda establecer y compartir fácilmente áreas de su interés con descripciones, que pueden incluir imágenes, tablas, enlaces a otras fuentes de datos de interés relacionado, etc. También es interesante que distintos usuarios colaboren para mejorar, completar o extender la información publicada.

Estas regiones pueden no tener vocación ni necesidad para ser permanentes, por ello se permite borrar artículos una vez ya estén creados. Como se puede ver en la Figura 4.

3.1 Contexto

Poner en contexto este TFG: ¿cómo encaja en el Mundo? ¿Es algo innovador o hay cosas similares en el mercado? Si hay cosas similares, ¿cómo se compara esta TFG con ellas? ¿Se hace en una empresa, en un grupo de investigación o es un proyecto independiente? ¿Se plantea como un primer paso hacia algo mayor o como un trabajo por sí mismo? ¿O es una mejora de algo existente? ¿Es una parte de algo que se ha hecho trabajando con más gente o es el resultado de un trabajo individual? Si es parte de un producto/sistema más grande, o una mejora o nueva versión, habrá que esforzarse en la memoria, y luego en la presentación, por trazar límites claros entre lo que había y lo que se ha hecho.

Locatopedia es una idea innovadora. Encontramos en el mercado multitud de proyectos relacionados, también, con un sitio web estilo wiki y geografía. El primero de ellos y más conocido es Wikipedia. Esta aplicación web es una enciclopedia editada de manera colaborativa. Entre otras cosas, se permite crear y editar artículos, al igual que en Locatopedia, pero cada artículo se identifica mediante el título del propio artículo. La diferencia la encontramos en la forma de identificar cada artículo, en Wikipedia se hace mediante el título, y en Locatopedia se hace mediante un identificador único de un conjunto de celdas de una malla global discreta superpuesta sobre el mapa. Otro sitio similar es OpenStreetMap, donde se pueden añadir lugares como carreteras, accidentes geográficos, negocios, tendidos eléctricos, etc. directamente sobre el mapa. En cierta forma es similar a Locatopedia, pero la gran diferencia, frente a OpenStreetMap, es la capacidad de poder escribir el

contenido que el usuario considere relevante, añadir imágenes, enlaces a otras fuentes de información directamente en cada región.

Wikimapia es el sitio más parecido al desarrollado en este proyecto. Los dos softwares permiten al usuario explorar lugares creados por otros, editarlos o completarlos. Wikimapia basa la localización de regiones con localizaciones exactas (usando coordenadas), y esa es la mayor diferencia con respecto a Locatopedia, que usa un conjunto de celdas de una malla global discreta superpuesta sobre el mapa.

Aunque haya sitios similares a Locatopedia, este ofrece algo distinto. Este proyecto introduce un enfoque basado en mallas globales discretas (DGGs), creando identificadores únicos para cada conjunto de celdas posible sobre el planeta. Locatopedia también permite a los usuarios crear nuevos artículos o modificar otros creados por otros usuarios.

Este proyecto se ha desarrollado de manera independiente, sin formar parte de un grupo de investigación o empresa. Sin embargo, su diseño modular y su arquitectura escalable permiten que puedan integrarse otros sistemas dentro de Locatopedia y evolucionar hacia un proyecto más grande en el futuro.

Ya sabemos que los límites de las aplicaciones están en nuestra imaginación.

La aplicación hasta ahora solo es la base de lo que podría ser un gran proyecto. Se puede seguir añadiendo multitud de nuevas características, por ejemplo, categorizar los artículos en, por ejemplo, Edificios, Tiendas, Hoteles, Hospitales, Ríos, Montañas, etc. Otro posible ejemplo podría ser añadir un mapa donde se pudiesen ver y explorar todas las regiones en la aplicación directamente sobre el plano. También podría añadirse una sección de discusión en cada artículo, de una forma similar a como lo hace Wikipedia.

3.2 Fuentes de datos integrados / mostrados / procesados

Si es un proyecto que integra, muestra, procesa... datos, hablar de las fuentes de los mismos. De dónde salen, qué licencias tienen, cuánto ha habido que procesarlos... Si el proceso de los datos ha sido una parte importante del trabajo, habrá que contarlos en detalle en la sección 3. Si el proyecto usa datos pero el procesado no se ha hecho como parte del trabajo, entonces basta con contarlos aquí.

4 Análisis del problema

Explicar brevemente, de manera general, lo que se va a hacer.

Se ha desarrollado una aplicación wiki en la que cada página se refiere a un lugar distinto.



Figura 1. Página principal de Locatopedia.

Cada región geográfica se identifica por medio de unas celdas de una malla global discreta superpuesta sobre el mapa. Cada conjunto de celdas tiene asociado un identificador único que sirve para identificar el artículo que trate este conjunto de celdas. Aparte de esto, se ha desarrollado un sistema de sesión, para que los usuarios puedan guardar sus artículos favoritos o acceder rápidamente a los artículos creados o editados por ellos. Para poder crear o editar artículos es necesario iniciar sesión aportando información personal, de esta forma se intenta controlar las versiones y evitar que usuarios hagan un uso incorrecto de la aplicación.


Los usuarios también tienen la posibilidad de realizar búsquedas de artículos tanto por texto como geográficamente. Al introducir un texto en el buscador, el servidor selecciona coincidencias exactas entre el texto buscado y todos los títulos y subtítulos de artículos en la aplicación. Además, se ha habilitado un mapa que permite búsqueda geográfica. En este caso, se añaden puntos en el mapa que van a crear un polígono que indica el área de búsqueda. Todos los artículos que, de manera parcial, o completa, estén incluidos en esta región, se mostrarán como resultados de la búsqueda. Los dos tipos de búsqueda se pueden combinar, es decir, se puede buscar artículos con un título o subtítulo determinado en una región determinada por el usuario.

4.1 Requisitos / Historias de Usuario / Casos de Uso

Se pueden combinar requisitos y casos de uso. No se suelen hacer otras combinaciones (p.ej. no tiene sentido tener requisitos e historias de usuario,

dado que ambas cosas sirven para lo mismo). Los requisitos pueden dividirse en funcionales y no funcionales, pero esta distinción es a veces un poco arbitraria. Si hay requisitos estrictos de prestaciones o de seguridad, generalmente serán no funcionales, describirlos lo más objetiva y cuantitativamente posible.

Requisito	Descripción
RF 1	El sistema debe permitir a los usuarios ver el contenido de artículos. Se debe permitir que el usuario navegue por el mapa y vea seleccionada la región sobre la que trata el artículo, y también debe poder ver el contenido del artículo.
RF 2	El sistema debe permitir a los usuarios ver el contenido de todas versiones anteriores de artículos.

RF3 <i>RF 3</i>	El sistema debe mostrar una página principal con los artículos más populares del momento, los artículos recientemente creados y editados y un artículo destacado.
RF4 <i>RF 4</i>	El sistema debe permitir a los usuarios realizar búsquedas de texto para encontrar artículos que contengan ese texto en su título o subtítulo.
RF5 <i>RF 5</i>	El sistema debe permitir a los usuarios realizar búsquedas geográficas para encontrar artículos que estén contenidos, de manera parcial o completa, en la región seleccionada por el usuario.
RF6 <i>RF 6</i>	El sistema debe permitir a los usuarios registrarse e iniciar sesión con su cuenta de Google.
RF7 <i>RF 7</i>	El sistema debe permitir a los usuarios registrados acceder a la información de su cuenta donde se encontrarán los artículos marcados como favoritos y artículos creados y modificados por el usuario.
RF8 <i>RF 8</i>	El sistema debe permitir a los usuarios registrados marcar y desmarcar como favoritos artículos.
RF9 <i>RF 9</i>	El sistema debe permitir a los usuarios registrados crear artículos añadiendo una región sobre la que trata el artículo, título, subtítulo y contenido. 
RF10 <i>RF 10</i>	El sistema debe permitir a los usuarios registrados crear artículos a partir de uno ya creado. Se usará como plantilla el artículo ya creado, y el usuario deberá modificar la región sobre la que trata el artículo, y podrá modificar título, subtítulo y contenido del artículo ya existente.
RF11 <i>RF 11</i>	El sistema debe permitir a los usuarios registrados editar artículos ya creados por cualquier usuario, modificando el título, subtítulo y/o contenido del artículo.
RF12 <i>RF 12</i>	El sistema debe permitir a los usuarios que estén creando o editando el contenido de un artículo, añadir formato al texto (negrita, cursiva, etc.), imágenes, tablas, enlaces, encabezados.

	Markdown es una opción válida para esto.
RF13 <i>RF 13</i>	El sistema debe permitir a los usuarios que estén creando o editando el contenido de un artículo, ver una previsualización del contenido del artículo.
RF14 <i>RF 14</i>	El sistema debe permitir a los usuarios registrados que hayan creado un artículo, eliminar dicho artículo.
RF15 <i>RF 15</i>	El sistema debe permitir a los usuarios administradores eliminar cualquier artículo del sistema.

RF 16

Tabla 1. Requisitos funcionales de la aplicación.

Requisito	Descripción
RNF1	El sistema tendrá una interfaz gráfica estéticamente agradable para cualquier tamaño de pantalla.
RNF2	El sistema usará los mapas de OpenStreetMap.
RNF3	El sistema será compatible con Google Chrome, Firefox y Safari.
RNF4	La aplicación web debe usar HTTPS y el dominio <i>locatopedia.com</i>
RNF5	El sistema debe de respetar la privacidad de los usuarios y no publicar sus datos privados, salvo dirección de correo.
RNF6	El sistema debe almacenar las imágenes en el servidor en formato webp, comprimidas.
RNF7	El sistema debe permitir a los usuarios expandir cualquier mapa en la aplicación para que ocupe las dimensiones completas de la pantalla.
RNF8	La división geográfica en celdas se hará utilizando una DGGS (Discrete Global Grid System). Se utilizará H3, desarrollada por Uber. Esta malla global discreta divide la superficie terrestre en hexágonos (y algunos pentágonos). Cada celda tiene un identificador único y puede subdividirse en celdas más pequeñas en niveles de mayor resolución.

Tabla 2. Requisitos no funcionales.

Título	Descripción	Prioridad
Crear página	Crear una nueva página de la wiki requiere seleccionar una o más celdas de un DGG sobre un mapa para establecer el lugar al que corresponde esa página. Una vez determinado el lugar, se le asigna automáticamente un id único (Area Unique)	M

	Identifier, AUID) y un hash con el que se construye su PURL (Permanent URL). Una vez creada una página, el conjunto de celdas que la definen no se puede cambiar.	
Contenido de una página	Título. Descripción. Palabras clave (pueden ser categorías o tags). Enlaces (a otras páginas de la locatopedia, o a otras páginas web en general). Fecha de creación. Periodo de tiempo en el que la entrada es “válida” (se pueden definir cosas que solo tienen interés temporalmente). Media (imágenes, vídeos, ficheros adjuntos...). Etc.	M
Borrar páginas	En principio borrar una página no debería ser una acción común. Si alguien la enlazaba, ese enlace se rompe, lo que nunca es bueno en la Web. Podemos marcar páginas como “obsoletas”, quizás apuntando a una “versión mejorada” de la página o algo así, si nos hemos equivocado al crearla. El único que debería poder borrar páginas sería un usuario administrador.	S
Editar página	Las páginas deberían guardar el historial de cambios de las mismas, no solo la última versión. También se debería poder tener en cada página algún tipo de chat que permita tener discusiones sobre la misma (y que estas discusiones queden también guardadas). Esto es como funciona p.ej. la Wikipedia.	M
Buscador de texto	Permite buscar palabras o frases en función de los títulos, subtítulos, descripciones de páginas (palabras clave etc.)	S
Buscador espacial	Permite seleccionar en el mapa una región (¿?rectangular (AKA “bounding box”) y muestra los artículos que contengan celdas dentro de esa región. El buscador puede usarse como filtro adicional sobre la consulta basada en texto (p.ej., “páginas que contengan `Zaragoza` pero que estén localizadas en el bounding box que corresponde a Sudamérica”).	S
Gestión de usuarios	Un usuario debe haber iniciado sesión para editar una página. ¿Iniciar sesión con Google? Debemos distinguir usuarios administradores y otros.	M
Página Home	Muestra las páginas más destacadas, páginas que hablen sobre tu ubicación actual, últimas actualizaciones, recurso del día...	M
Crear página en función de una	Permite crear una página nueva a partir de una página ya existente. La idea es facilitar la creación de localizaciones “alternativas” o parecidas modificando el conjunto de celdas original. Se	S

existente	debe facilitar que las páginas derivadas de otras automáticamente estén enlazadas con estas, posiblemente asignando una semántica a esta relación (p.ej. “modificación”, “ampliación”, ...).	
Links a otras páginas	Permite añadir links para hacer referencia a otras páginas de Wikiplace.	M

Tabla 3. Primeras ideas.

	Usuario No Registrado	Usuario Registrado	Usuario Administrador

Tabla 4. Comparativa de funcionalidades de distintos tipos de usuarios.

4.2 Casos de uso

4.2.1 Ver un artículo.

- Descripción: Este caso de uso permite a los usuarios de Locatopedia ver el contenido de un artículo.
- Flujo de Eventos Principal:
 1. El usuario encuentra un artículo en la página Home o de su cuenta, busca un artículo en la barra de búsqueda o con el mapa o entra en un enlace a un artículo de Locatopedia.
 2. El usuario ve el título, subtítulo, mapa y contenido. Puede deslizar hacia abajo la página y ver el contenido completo del artículo.
- Flujo de Eventos Alternativo:
 - El usuario puede deslizarse por el mapa viendo partes que no forman parte del conjunto de celdas al que el artículo se refiere.
 - El usuario puede hacer clic en el corazón para marcar este artículo como favorito. O eliminarlo de favoritos si ya estaba marcado como favorito.

4.2.2 Ver versiones anteriores de un artículo

- Descripción: Este caso de uso permite a los usuarios de Locatopedia ver el contenido de una versión pasada de un artículo.
- Flujo de Eventos Principal:

3. El usuario encuentra un artículo en la página Home o de su cuenta, busca un artículo en la barra de búsqueda o con el mapa o entra en un enlace a un artículo de Locatopedia.
4. El usuario desliza abajo y selecciona la opción de 'See other versions of this page'.
5. El usuario selecciona una versión.
6. El usuario ve el título, subtítulo y contenido de la versión seleccionada.

4.2.3 Iniciar Sesión

- Descripción: Este caso de uso permite a los usuarios de Locatopedia iniciar sesión con su cuenta de Google.
- Flujo de Eventos Principal:
 7. El usuario pulsa el botón de 'Sign Up/Sign In'.
 8. El usuario selecciona la opción de 'Continue with Google'.
 9. El usuario inicia sesión en Google y otorga los permisos requeridos por Locatopedia.
- Flujo de Eventos Alternativo
 - El usuario puede seleccionar la opción de cancelar. En este caso volvería a la página Home sin iniciar sesión.

4.2.4 Crear un artículo nuevo

- Descripción: Este caso de uso permite a los usuarios con sesión iniciada en Locatopedia crear un artículo nuevo en el sistema.
- Flujo de Eventos Principal:
 10. El usuario entra en la página personal.
 11. El usuario selecciona la opción de 'Create new article'.
 12. El usuario decide un título, subtítulo y contenido para el artículo a crear.
 13. El usuario selecciona un conjunto de celdas sobre las que tratará su artículo.
 14. El sistema comprueba que los campos de título, subtítulo y contenido no estén vacíos. También comprueba que la selección de celdas no sea ni nula ni esté ya registrada en el sistema. En caso correcto, se habilita el botón de 'publish', que publica el artículo.
 15. El usuario selecciona la opción de 'publish' para publicar el artículo.
 16. El usuario confirma su elección y publica el artículo.
- Flujo de Eventos Alternativo

- Si el usuario no tiene una sesión válida, deberá iniciar sesión previamente. Ya explicado en punto 2.
- Si el usuario selecciona la opción de cancelar, volverá a la página personal.
- En el paso 7, si el usuario decide no confirmar, se volverá al paso 5.

4.2.5 Crear un artículo a partir de otro

- Descripción: Este caso de uso permite a los usuarios con sesión iniciada en Locatopedia crear un artículo nuevo en el sistema a partir de otro.
- Flujo de Eventos Principal:
 - 17. El usuario entra en el artículo que quiere usar como plantilla para el nuevo artículo.
 - 18. El usuario selecciona la opción de 'create similar'.
 - 19. El usuario decide un título, subtítulo y contenido para el artículo a crear.
 - 20. El usuario selecciona un conjunto de celdas sobre las que tratará su artículo.
 - 21. El sistema comprueba que los campos de título, subtítulo y contenido no estén vacíos. También comprueba que la selección de celdas no sea ni nula ni esté ya registrada en el sistema. En caso correcto, se habilita el botón de 'publish', que publica el artículo.
 - 22. El usuario selecciona la opción de 'publish' para publicar el artículo.
 - 23. El usuario confirma su elección y publica el artículo.
- Flujo de Eventos Alternativo
 - Si el usuario no tiene una sesión válida, deberá iniciar sesión previamente. Ya explicado en punto 2.
 - Si el usuario selecciona la opción de cancelar, volverá a la página personal.
 - En el paso 7, si el usuario decide no confirmar, se volverá al paso 5.

4.2.6 Editar un artículo

- Descripción: Este caso de uso permite a los usuarios con sesión iniciada en Locatopedia editar un artículo ya existente en el sistema.
- Flujo de Eventos Principal:
 - 24. El usuario entra en el artículo a editar.
 - 25. El usuario selecciona la opción de 'edit page'.
 - 26. El usuario decide un nuevo título, subtítulo y/o contenido para el artículo a editar.

27. El sistema comprueba que los campos de título, subtítulo y contenido no estén vacíos y al menos se haya modificado uno de ellos. En caso correcto, se habilita el botón de 'save changes, que publica el artículo.
 28. El usuario selecciona la opción de 'save changes para publicar el artículo.
 29. El usuario confirma su elección y modifica el artículo.
- Flujo de Eventos Alternativo
 - Si el usuario no tiene una sesión válida, deberá iniciar sesión previamente. Ya explicado en punto 2.
 - Si el usuario selecciona la opción de cancelar, volverá a la página anterior.
 - En el paso 6, si el usuario decide no confirmar, se volverá al paso 4.

4.3 Interfaces de usuario

Para productos con GUI, orientados principalmente a humanos. Desde el punto de vista de la captura de requisitos: borradores, bocetos, navegación entre pantallas etc. No la GUI tal y como queda finalmente, que se contaría en la sección 3.

Locatopedia es una aplicación web, por lo tanto, debe tener una buena interfaz de usuario. La idea principal del sitio es que este disponible y funcione correctamente en ordenador. De todas maneras, se ha optimizado el diseño para cualquier tipo de dispositivo, incluido teléfonos móviles y tablets.



Figura 2. Locatopedia en ordenador.



Figura 3. Locatopedia en móvil.



Figura 4. Locatopedia en tablet.

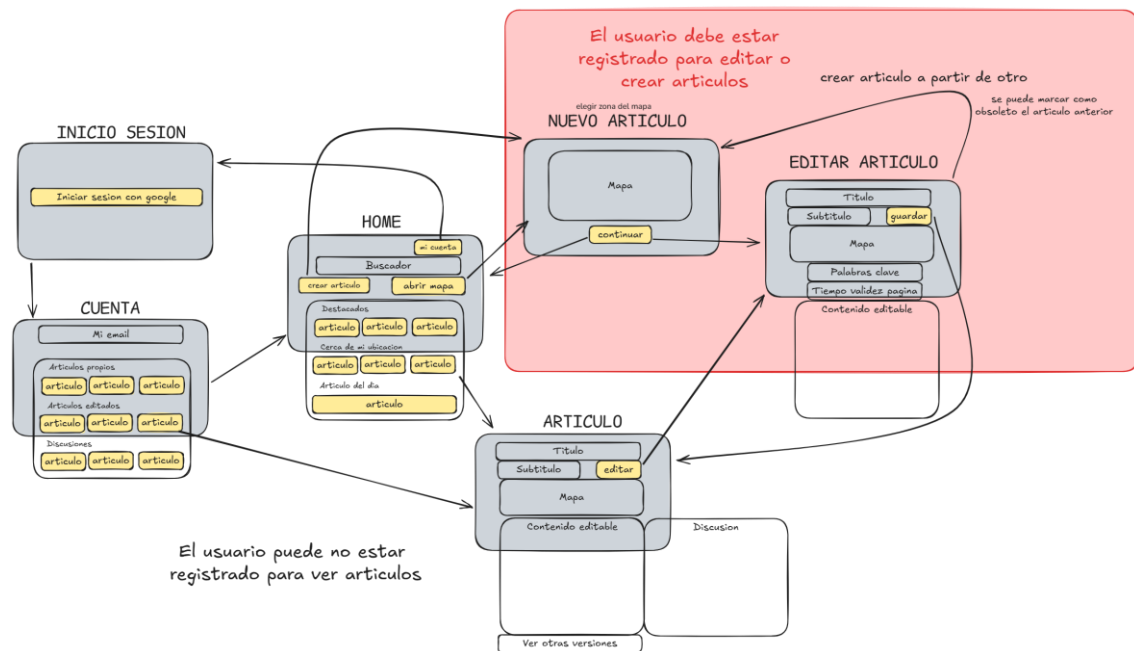


Figura 5. Primera versión del mapa de navegación de la aplicación.

La aplicación tendrá un conjunto de pantallas reducido y cada una será sencilla e intuitiva. Todas las páginas de la aplicación tendrán la misma cabecera que mostrará un enlace a la pagina inicial y a la sección de **MI CUENTA**.

La primera sección que aparecerá al entrar en la aplicación será la página **INICIO**. Tendrá un resumen de los artículos mas populares y los recientemente modificados y creados. En la parte superior se podrán buscar artículos por texto o seleccionado una zona en el mapa. Los criterios de búsqueda podrían combinarse.

La página **ARTICULO** mostrará el contenido del artículo. Apareceran campos como título, subtítulo o contenido. Tambien se verá un mapa interactivo donde se resaltará la región del articulo en cuestión. Se permite ver versiones anteriores del articulo por medio de un botón en la parte inferior de la pagina. Además, se podrá ver la discusión de los usuarios en relación al contenido, e incluso se podrá participar en dicha discusión. Por ultimo encontraremos una opción para editar el articulo según nuestro criterio.

Las opciones de editar articulo y participar en la discusión solo podrán realizarlas usuarios registrados.

En la sección **EDITAR ARTICULO**, aparecerán los mismos campos que en la pagina de **ARTICULO**, pero con campos de contenido editable. Es decir, un usuario registrado será capaz de editar el titulo, subtítulo y/o contenido de cualquier articulo. No será posible modificar la región que ocupa el articulo, ya que esta región es el identificador del articulo.

Para crear un articulo nuevo será necesario usar el mapa de la pagina **NUEVO ARTICULO** para seleccionar la región que tratara el nuevo articulo. Después se establecerá el contenido en una pestaña similar a **EDITAR ARTICULO**.

La página *INICIO SESION* se encargará de redirigir al usuario a servicios de terceros para que inicie sesión usando las cuentas que tenga el usuario en estos servicios. Google será el principal.

Una vez se haya iniciado sesión, se podrá navegar a la sección *MI CUENTA* que mostrará información personal del usuario, como el email y artículos relacionados con el usuario. Aparecerán los artículos creados y modificados y en los que el usuario haya participado en sus discusiones.

4.4 Interfaces para la integración en aplicaciones

Para productos sin GUI, o con una GUI de administración o similar pero cuyo principal valor está en proporcionar servicios a otros sistemas software (p.ej. ofreciendo una API). Una API no significa “una API REST”. Una API pueden ser las clases y operaciones públicas principales de una biblioteca de software, un conjunto de scripts de procesamiento de datos... En cualquier caso aquí se describe esa API desde el punto de vista de la captura de requisitos: qué se quiere ofrecer a otros sistemas. El resultado final se describiría en la sección 3.

La aplicación GUI de la página web sigue una estructura organizada para todas las secciones que hay. Cada sección tiene una ruta API distinta (por ejemplo */account* corresponde a la sección personal) con posibles parámetros (por ejemplo el identificador de un artículo). Es decir, si conocemos o leemos la documentación, podemos navegar por la aplicación sin necesidad de hacer clic en botones dentro de la aplicación.

Por otro lado, el servidor del sitio web debe almacenar y proporcionar toda la información necesaria para mostrarla en el sitio web. Aparte de esto, también debe ser posible modificar esa información, acción requerida al crear un nuevo artículo o editar uno ya existente.

Para ello se ha desarrollado un conjunto de APIs que permiten ejecutar todas las acciones requeridas y recuperar toda la información necesaria. Estas consultas se dividen en 2 grandes grupos.

Las primeras, son consultas no protegidas, es decir, para acceder a estas APIs no hace falta que el usuario haya iniciado sesión previamente. En primer lugar, el servidor ofrece una ruta para recuperar toda la información de un artículo (RF 1). Locatopedia ofrece también un historial de versiones, por lo tanto, se ofrece una ruta API para recuperar versiones anteriores de un artículo (RF 2). Además de esto, se ofrecen rutas APIs que devuelven los artículos populares (los que más usuarios han marcado como favoritos), los recientemente creados y editados y un artículo aleatorio que se mostrará como artículo del día (RF 3). Por último, se deben añadir rutas para permitir a los usuarios hacer búsquedas, tanto por texto como geográficas. Para ello, se ha añadido una ruta que devuelve los artículos que coinciden, en texto, con un parámetro de búsqueda (RF4). La coincidencia geográfica se resuelve en el cliente (RF5).

Para iniciar sesión, no se ha añadido ninguna ruta específica. Se utiliza OAuth 2.0, y es Google quién gestiona registro, inicio de sesión en la aplicación y validación de sesiones (RF6).

Por otro lado, encontramos las consultas API que requieren de autenticación en la aplicación. El servicio API recupera el token de identificación, que se debe añadir a la consulta, y el servidor proporciona información privada correspondiente asociada al usuario relacionado con dicho token. Una de las rutas protegidas que el servidor proporciona permite marcar y desmarcar como favorito un artículo (RF8), otra, permite recuperar los artículos favoritos de un usuario y los artículos creados y modificados por un usuario (RF7). Aparte de esto, el servicio de API proporciona rutas para crear nuevos artículos (RF9 y RF10) y editar artículos ya existentes (RF11) incluyendo la información necesaria en la solicitud API. También se ha añadido una ruta API para permitir borrar un artículo al usuario creador (RF14) y a usuarios administradores (RF15).

5 Diseño de la solución

5.1 Arquitectura

Describir la idea general: estilo arquitectural principal del sistema, frameworks utilizados y qué implicaciones arquitecturales tienen. Si hay consideraciones importantes de prestaciones o de seguridad que ha habido que tener en cuenta.

Locatopedia sigue más de una arquitectura para aplicaciones web.

En primer lugar, se ha desarrollado una arquitectura multicapa de tres niveles.

- **Cliente.** Es la aplicación con la que interactúa el usuario final. Le da las funcionalidades necesarias para poder ver, modificar y editar artículos y su cuenta. También se conoce como *frontend* o interfaz gráfica de usuario GUI y se desarrolla con el framework *React*. Proporciona una interfaz dinámica, intuitiva y responsiva ante distintos tamaños de pantalla. Esta capa aplica medidas de seguridad para evitar que un usuario acceda a información sensible, como datos personales de otros usuarios.
 - **Servidor.** Es la aplicación que se ejecuta en la nube que proporciona la lógica al cliente. Se desarrolla utilizando la tecnología moderna llamada *Node.js* que actúa como una capa intermedia que gestiona la lógica de negocio, maneja la autenticación y se comunica con la base de datos. Permite que las modificaciones realizadas por el usuario se vean reflejadas en todos los clientes. Esta capa también aplica medidas de seguridad. Los recursos protegidos requieren el envío de una credencial proporcionada por el servicio de tercero, para autenticar y autorizar el acceso o edición de un cliente a un recurso.
 - **Base de datos.** Almacena de forma estructurada todos los datos de la aplicación para proporcionar un acceso rápido y seguro. Se utiliza un sistema *PostgreSQL* ya que es de código abierto y muy escalable. Por ahora el sistema no almacena mucha información, pero debe ser capaz de soportar un gran volumen de datos. *PostgreSQL* proporciona una escalabilidad excelente, múltiples conexiones simultáneas y replicación que lo convierte en una de las mejores bases de datos actualmente.
-
- El *frontend* o interfaz gráfica de usuario (GUI) se ha desarrollado en *React*. Se encarga de la interacción con el usuario, proporcionando una interfaz dinámica, intuitiva y responsiva ante distintos tamaños de pantalla.
 - El *backend* o lógica de página web se ha desarrollado utilizando la tecnología moderna llamada *Node.js* que actúa como una capa intermedia que gestiona la lógica de negocio, maneja la autenticación y se comunica con la base de datos.
 - La base de datos *PostgreSQL* almacena los datos estructurados de los artículos y sus asociaciones con las celdas de la malla global discreta

(DGG). Además, almacena información adicional, como artículos favoritos por usuario.

Para la parte del *backend* se ha seguido una arquitectura híbrida. En la mayor parte funciona como un *backend* monolítico, donde es propio sistema el que responde a todas las peticiones de la API, pero se ha añadido un servicio en Python. Esto se debe a que la librería de gestión de hashes e identificadores únicos de celdas esta escrita en Python. Se ha desarrollado una API interna con Flask para proporcionar acceso a los métodos y funciones ya desarrollados en Python.

Tanto el cliente como el servidor se ejecutan en la misma máquina.

5.1.1 Contexto

Esta sección seguramente solo tendrá sentido para sistemas que son parte de otros mayores, para dejar claro qué parte es lo del TFG, y qué parte es lo que ya había.

La mayor parte de este proyecto es elaboración propia. Se han utilizado librerías públicas como turf (para gestionar las formas geométricas de las celdas), leaflet (para los mapas), react-oauth (para gestionar la sesión con Google) o h3 (para gestionar las celdas de la malla global discreta). También se ha utilizado la librería en Python mencionada previamente desarrollada por Rubén Béjar (tutor de este trabajo). El resto del desarrollo es parte del TFG.

5.1.2 Modelo de datos

Un modelo de entidades y relaciones (diagrama de clases UML o, si no hay otro remedio, notación prehistórica E-R). Más las explicaciones fundamentales: qué es cada entidad, qué es cada atributo, qué implican las relaciones.

Se puede complementar con un modelo de implementación (modelo relacional, documentos JSON en MongoDB o lo que sea), pero lo más crítico es el modelo de entidades y relaciones.

Este es diagrama del modelo entidad-relación que almacena toda la información necesaria para Locatopedia.

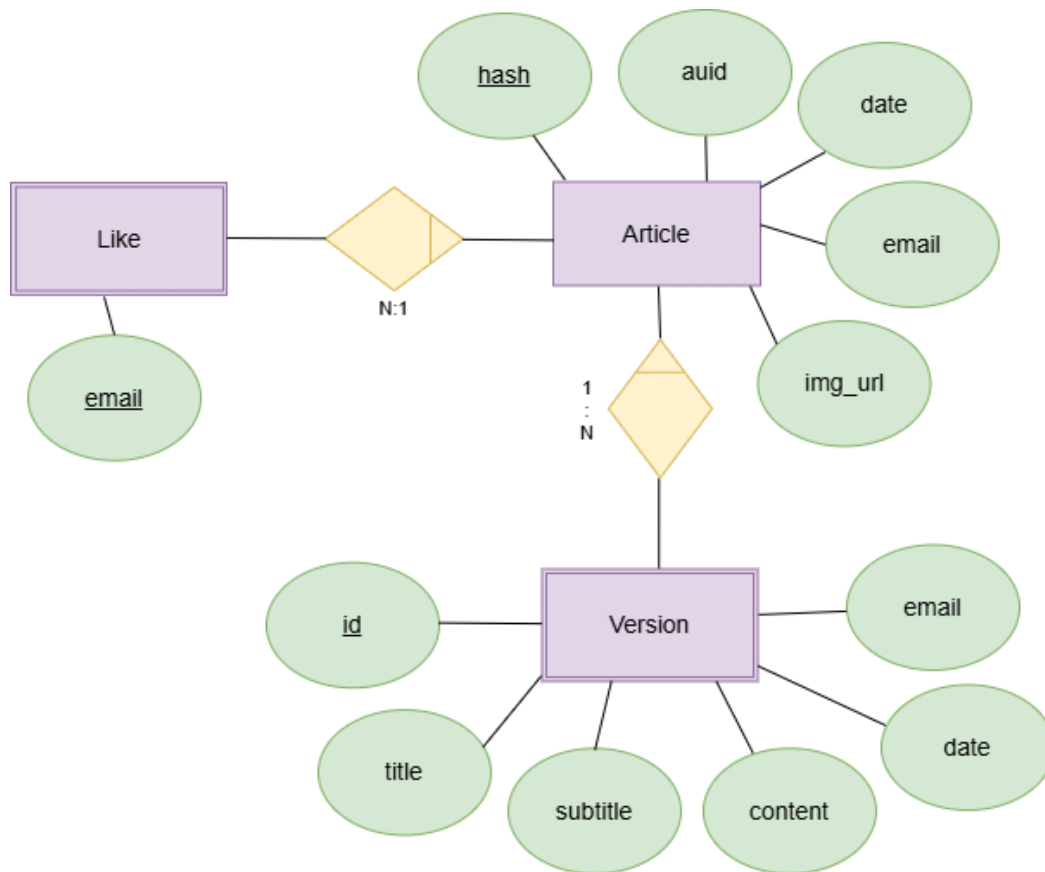


Figura 6. Diagrama Entidad-Relación.

Las entidades se representan con un rectángulo de color morado, y tienen doble marco en caso de ser una entidad débil. Se pueden apreciar 3 entidades, dos de ellas entidades débiles y una entidad fuerte. Recordemos que las entidades débiles dependen de otra entidad, en cambio, las entidades fuertes existen por sí mismas. En este caso, las entidades de 'Like' y de 'Version', deben tener asociadas una entidad 'Article' para que tengan sentido. Por ello, las relaciones 'Like' - 'Article' y 'Version' - 'Article' sufren una dependencia. Si se eliminase un artículo, todo elemento en 'Like' o 'Version' carece de sentido. Además, la cardinalidad de estas relaciones es de uno a muchos (1 : N). Esto quiere decir que un artículo se puede relacionar con muchos *likes*, pero un *like* solo puede relacionarse con un único artículo. De la misma forma, un artículo puede tener muchas versiones, pero una versión siempre está relacionada con un único artículo.

Vemos que todas las entidades tienen atributos que cuelgan de ellas. La entidad 'Article' tiene en total 5. Se puede observar que el atributo *hash* está subrayado, esto indica que este campo forma parte de la clave primaria de la entidad, es decir, solo puede haber un artículo con este *hash* en el sistema. Otros atributos, como 'email' o 'date', entre otros, pueden repetirse en el sistema.

En el caso de la entidad 'Version', también se pueden ver varios atributos, uno de ellos es parte de la clave primaria. Es importante recordar que, al ser una entidad débil, añade a sus atributos la clave primaria de la entidad con la que está relacionada, que funciona como segunda parte de la clave primaria. En este caso, el campo *hash* también se encuentra en la entidad 'Version'.

La entidad 'Like' tiene un comportamiento similar. La primera parte de la clave primaria es el atributo email, y la segunda parte se hereda de la entidad fuerte de la que depende 'Article'.

5.1.3 Paquetes y clases

- Un diagrama UML con los paquetes principales. Tanto del front-end como del back-end, si hay de ambos. Sin mucho detalle, es una idea general.

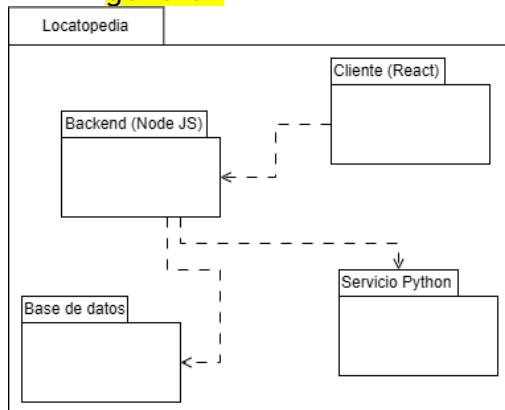


Figura 7. Diagrama de paquetes.

- Un diagrama UML de clases con las clases principales. De nuevo tanto de front-end como de back-end si tiene sentido. Centrarse en las clases principales, en sus relaciones y en sus operaciones principales.

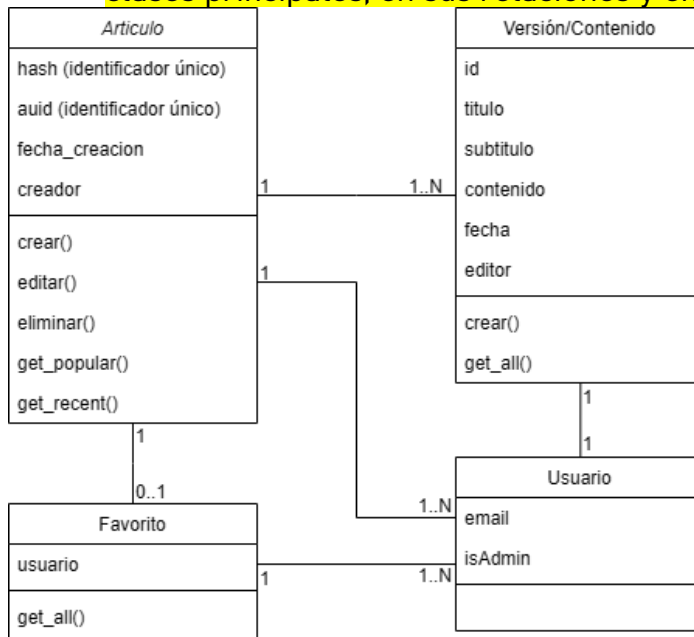


Figura 8. Diagrama de clases.

Y, por supuesto, todas las explicaciones pertinentes que permitan entender esos diagramas: descripción de los paquetes, papel de las clases principales en el sistema, patrones de diseño y estilos arquitecturales empleados,

principales decisiones tomadas (decisiones implica que había varias opciones sensatas y se ha elegido una de ellas basándose en criterios racionales),

5.1.4 Componentes y conectores

- Un diagrama UML con los componentes principales y sus conectores. Dependiendo del tipo de sistema esos componentes serán objetos, o serán procesos que agruparán a muchos objetos, o serán servicios web...
- Algún diagrama UML (secuencia, interacción, máquinas de estados...) ilustrando el comportamiento dinámico del sistema suele ser interesante, especialmente si este comportamiento, o alguna parte de l mismo, tiene cierta complejidad.

Y, por supuesto, todas las explicaciones pertinentes que permitan entender esos diagramas: qué es cada componente, qué permite cada conector, documentación de las interfaces (puertos/API) más relevantes del sistema, aspectos de concurrencia/asincronicidad/paralelismo/operaciones periódicas, estilos y patrones arquitecturales y de diseño que tiene más sentido explicar aquí que en la parte de módulos, decisiones de diseño relevantes en esta parte etc.

Relacionar componentes con módulos es interesante (cada línea de código que hemos puesto en un módulo debería ejecutarse en algún componente), sobre todo si ayuda a clarificar algún aspecto complejo.

5.1.5 Distribución

- Un diagrama UML del despliegue (artefactos en nodos interconectados) y/o de la instalación (artefactos sobre el sistema de ficheros).

Y, por supuesto, todas las explicaciones pertinentes que permitan entender esos diagramas: qué es cada nodo, qué tipo de rutas los interconectan, cómo se relacionan los artefactos de estos diagramas con los componentes de la sección anterior... Todo esto es especialmente interesante si hay un despliegue automatizado, cosas en Cloud, en contenedores (Docker directamente, o se usa Kubernetes o similar...) y habría que explicar todo eso en esta sección.

5.2 Implementación

Si esto es lo bastante grande, puede tenerse un capítulo entero (sería el 4) para la Implementación.

Aspectos interesantes de la implementación que sean de bajo nivel de abstracción, o estén localizados en módulos/componentes particulares y que no se hayan considerado lo bastante interesantes para el punto de arquitectura se pueden poner aquí.

Aquí se puede describir la GUI que ha quedado al final del sistema, e incluso poner un diagrama de navegación definitivo si tiene sentido.

Locatopedia desarrolla una interfaz gráfica de usuario sencilla y funcional, diseñada para que cualquier persona pueda interactuar con ella sin necesidad de una curva de aprendizaje pronunciada. Se ha seguido un estilo minimalista mejorando la experiencia del usuario.

En todas las paginas encontramos una cabecera donde se incluye una barra de búsqueda de texto y un mapa para hacer búsqueda geográfica para que los usuarios puedan leer los artículos que más les interesen en todo momento. Además, encontramos enlaces tanto a la página principal de Locatopedia y otro enlace a la pagina de cuenta personal. En caso de no haber iniciado sesión, este enlace te permitirá hacerlo.

La página principal muestra un resumen general de los artículos existentes, artículos más populares del momento, y que se han editado o modificado recientemente permitiendo a los usuarios explorar algunos de los artículos disponibles de manera intuitiva.

Cada articulo se muestra en una página distinta, identificada en la URL de forma permanente por el conjunto de celdas que contiene dicho artículo. Esto significa que el enlace web de un artículo no cambiará nunca, aunque se modifique el título, subtítulo y/o contenido de un artículo. Aquí se muestra otra informacion relevante sobre el artículo, como el autor, fecha y hora de la última modificación del artículo correspondiente. Además, se puede acceder a la página de versiones del artículo, donde se puede acceder al contenido de todas las versiones anteriores de dicho artículo.

En la pagina de cuenta personal, accesible solo con sesión iniciada, se muestran los artículos marcados como favoritos y los artículos creados y modificados por el usuario. Aquí aparece un botón para cerrar la sesión iniciada y un botón para crear un articulo nuevo. En esta página, se debe elegir un título, un subtítulo, un contenido y el conjunto de celdas sobre las que tratará el nuevo artículo.

Para crear un nuevo artículo o para editar uno ya existente, existe una pagina que permite cambiar los valores de título, subtítulo y contenido y permite modificar la selección de celdas en caso de tratarse de ser creación de un artículo.

Los botones son grandes y bien espaciados, asegurando accesibilidad y facilidad de uso. Cada artículo se puede visualizar con un solo click, mostrando su contenido estructurado con descripciones, imágenes y enlaces relevantes. La edición y colaboración también se simplifican con un sistema intuitivo de modificación de contenido, donde los usuarios pueden agregar o actualizar información sin perder la coherencia del sitio. También pueden ver una previsualización en legible del texto Markdown que escriben en el contenido de un artículo.

El diseño prioriza la claridad y la funcionalidad, evitando elementos visuales innecesarios que puedan dificultar la experiencia. Los colores y tipografías están elegidos para ofrecer una lectura cómoda, con contrastes adecuados y un esquema visual limpio. En general, la interfaz busca ser lo más ligera

posible sin sacrificar usabilidad, asegurando que la aplicación sea rápida y eficiente en cualquier dispositivo.

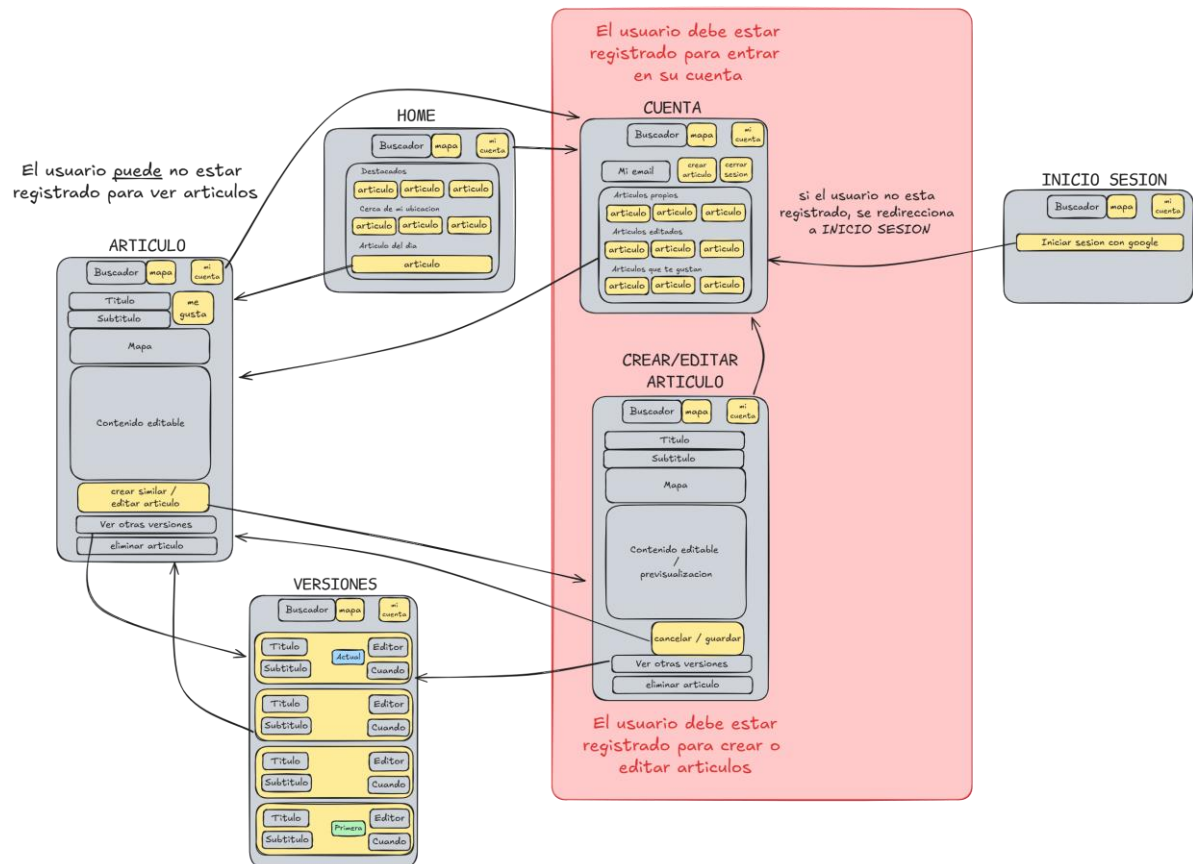


Figura 9. Mapa de navegación.

La aplicación está diseñada para recibir dos tipos de usuarios con niveles de interacción distintos. Los más frecuentes son los usuarios no registrados, quienes simplemente acceden, navegan por la aplicación leyendo y obteniendo información de los artículos ya existentes sin necesidad de realizar ninguna acción adicional. Para ellos, la interfaz es extremadamente sencilla, con una estructura clara que les permite explorar el contenido sin obstáculos ni configuraciones innecesarias.

Por otro lado, los usuarios más avanzados son aquellos que editan y crean nuevos artículos o buscan versiones anteriores de artículos existentes. Para ellos, la interfaz debe ofrecer herramientas adicionales que les permitan gestionar dicha información de manera eficiente. Aunque la aplicación sigue manteniendo un enfoque minimalista, en este caso la GUI se vuelve algo más compleja, ya que incluye opciones para modificar el contenido de los artículos o crear nuevos artículos. Estos usuarios deben aprender a utilizar estas funcionalidades, por lo que la interfaz está diseñada para guiarlos en el proceso sin ser abrumadora.

Este equilibrio entre simplicidad y funcionalidad asegura que la experiencia sea accesible para los visitantes ocasionales, pero suficientemente potente para aquellos que desean contribuir activamente al contenido de la plataforma.

Si la construcción se ha automatizado (TravisCI, Github Actions o similares), describir aquí el pipeline de construcción. Describir el uso de sistemas de control de versiones (cuántos repositorios, qué workflows de Git etc.).

5.3 Procesamiento de datos

Si el proyecto ha tenido una parte importante de procesamiento de datos, explicarlo aquí. Si no, es probable que esta sección sobre.

5.4 Pruebas

Se incluyen tests automáticos y tests manuales para validación y verificación, pero también tests de prestaciones o tests de estrés (sobrecarga) si el tipo de proyecto los requiere. Herramientas usadas, planificación de las pruebas, guiones de las pruebas manuales, integración de las pruebas automáticas en el pipeline de construcción. Si se ha usado Sonar o algún otro analizador estático de código explicarlo aquí.

Cualquier tipo de proyecto va a requerir pruebas: cómo se validan los resultados es crítico se haga lo que se haga.

6 Gestión del proyecto

Señalar la metodología de gestión que se ha seguido. Dependiendo de la misma incluir planes, división del trabajo en iteraciones/sprints, horas de esfuerzo realizadas, análisis de riesgos realizado, etc.

7 Plan de producto / plan de negocio / análisis de mercado

Opcional. En la mayoría de los TFG no vamos a tener esto, pero si el TFG tiene orientación a mercado (por ejemplo, es algo que se ha hecho con la intención de comercializarlo, o de llegar a comercializar una versión futura), cabe perfectamente tener un capítulo sobre esto en la memoria.

8 Conclusiones

Resumir los resultados/aportaciones principales del proyecto. Dar ideas de por dónde podría seguir el trabajo. Se puede incluir una valoración personal del proyecto y una descripción de las incidencias y problemas encontrados y resueltos durante el mismo.

9 Bibliografía

Las citas bibliográficas que se hayan hecho en el texto. Numeradas consecutivamente según se indica en las recomendaciones de la EINA.

10Anexos

Manual de administración si tiene sentido.

Manual de despliegue/instalación si tiene sentido.

Manual de usuario si tiene sentido.

Diagramas detallados que en el cuerpo principal de la memoria eran demasiado.

Resultados detallados de pruebas que en el cuerpo principal de la memoria eran demasiado.

Documentación completa de la/s API más relevantes.

Etc.