



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

I. Resumen

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

II. Tabla de contenidos o Índice

III. 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.

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.

2. Fuentes de datos integrados / mostrados / procesados [elegir lo que sea]

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 contarlo 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 contarlo aquí.

IV. Análisis del problema

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

Se ha desarrollado una aplicación web con estilo wiki en la que cada artículo es identificado geográficamente. Esta identificación se hace 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.

1. Requisitos / Historias de Usuario / Casos de Uso [elegir lo que sea]

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
RF1	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.
RF2	El sistema debe permitir a los usuarios ver el contenido de versiones anteriores de artículos.
RF3	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	El sistema debe permitir a los usuarios realizar búsquedas de texto para encontrar artículos que contengan ese texto en su título o subtítulo.
RF5	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	El sistema debe permitir a los usuarios expandir cualquier mapa en la aplicación para que ocupe las dimensiones completas de la pantalla.
RF7	El sistema debe permitir a los usuarios registrarse e iniciar sesión con su cuenta de Google.
RF8	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.
RF9	El sistema debe permitir a los usuarios registrados marcar como favoritos artículos.
RF10	El sistema debe permitir a los usuarios registrados crear artículos añadiendo arbitrariamente región sobre la que trata el artículo, título, subtítulo y contenido.
RF11	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.
RF12	El sistema debe permitir a los usuarios registrados editar artículos ya creados por cualquier usuario, modificando el título, subtítulo y/o contenido del artículo.
RF13	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.
RF14	El sistema debe permitir a los usuarios que estén creando o editando el contenido de un artículo, ver una previsualización del contenido del artículo.
RF15	El sistema debe permitir a los usuarios registrados que hayan creado un artículo, eliminar dicho artículo.
RF16	El sistema debe permitir a los usuarios administradores eliminar cualquier artículo del sistema.

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.

Tabla 2. Requisitos no funcionales.

Titulo	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 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.	M
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	S

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

Tabla 3. Primeras ideas.

2. Casos de uso

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 deslizar 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.

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.

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. 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.

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.

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.

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.

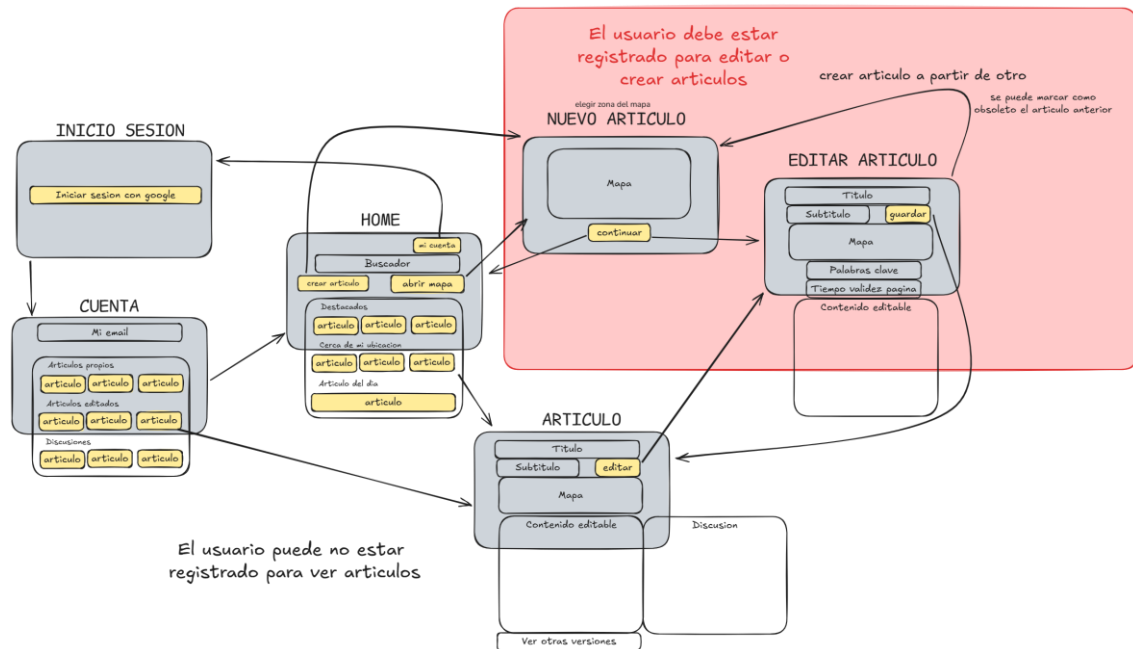


Figura 1. Primera versión del mapa de navegación de la aplicación.

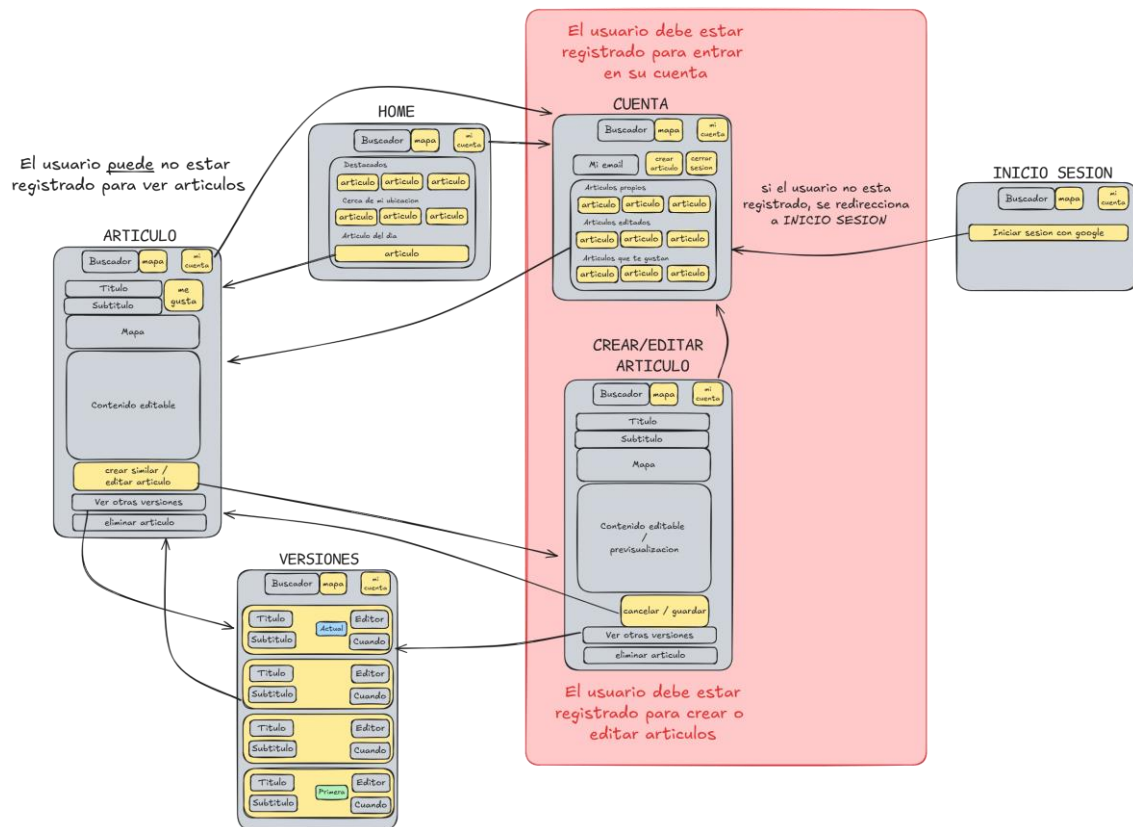


Figura 2. Versión final del mapa de navegación de la aplicación.

4. Interfaces para la integración en aplicaciones [este título de sección seguramente haya que adaptarlo a lo que se haya hecho]

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.

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 consultas API 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 asociada al usuario relacionado con dicho token. El servidor debe proporcionar una ruta API para

V. Diseño de la solución

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.

7. 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.

8. 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.

9. 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.
- 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.

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),

10. 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 del 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.

11. Distribución

- Un diagrama UML del despliegue (artefectos en nodos interconectados) y/o de la instalación (artefectos 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.

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.

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.).

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.

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.

VI. 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.

VII. 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.

VIII. 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.

IX. Bibliografía

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

X. Anexos

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.