MEMORIA - PRÁCTICA 2

Grupo: 3B

Integrantes:

Cristian Ruiz Martín

Pablo Rubia Arias

Mikolaj Zabski

Ignacy Borzestowski

Olivier Gabana Gómez

José Francisco Ruiz Sierras

Tecnologías usadas

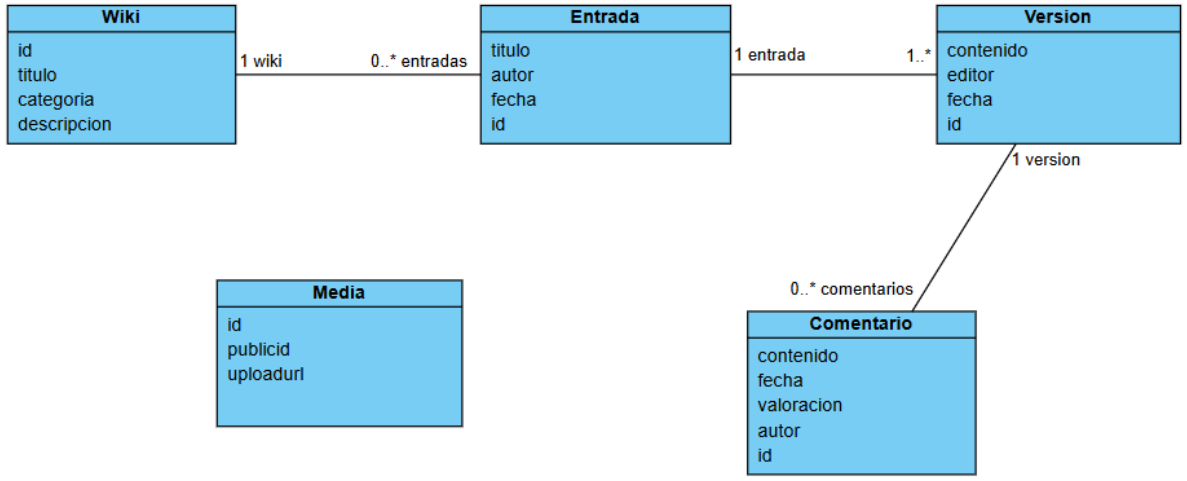
MongoDB: una base de datos no relacional. Es simple y no requiere esquema. Las bases de datos no relacionales tienen un diseño flexible que permite agrupar las entidades de una forma más libre que una relacional. MongoDB es una de las más conocidas dentro de esta categoría y está orientada a documentos (JSON).

Go: es adecuado para aplicaciones web y de redes debido a su sistema de concurrencia eficiente, su velocidad de ejecución y su facilidad para trabajar con APIs y servicios.

Chi: es una biblioteca de enrutamiento HTTP ligera y eficiente diseñada para aplicaciones en Go. Se ha elegido Chi por su bajo costo en términos de rendimiento y porque permite definir rutas y middlewares de forma clara y directa.

Cloudinary: es el servicio de gestión de imágenes en la nube. Se seleccionó por sus capacidades y la facilidad de integración mediante su API, permitiendo almacenar y acceder a los archivos multimedia de forma segura y escalable.

Diseño preliminar de la base de datos



Se ha decidido representar en el diagrama las entidades Wiki, Entrada, Versión y Comentario.

El diagrama es bastante sencillo, una wiki puede estar compuesta por varias entradas y cada entrada solo puede estar relacionada con una wiki (multiplicidad 1 a 0..*). Una entrada está relacionada con al menos una versión ya que al menos tiene una versión original y una versión solo puede estar relacionada con una Entrada (multiplicidad 1 a 1..*). Por último, una versión puede estar relacionada con varios comentarios y un comentario solo puede estar relacionada con una versión. Hemos elegido este diseño debido a su simplicidad y facilidad de entendimiento y de uso.

Por otro lado, hemos decidido implementar el microservicio de media, que también estaría en la base de datos, pero no estaría relacionada con ninguna otra entidad.

La base de datos estará alojada en la web MongoDB Atlas.

Creación de microservicios a partir de entidades de la base de datos

Los microservicios han sido diseñados de manera que cada uno representa una entidad de la base de datos, que son: Wiki, Entrada, Versión y Comentario. Así, cada uno accede a la base de datos únicamente si se necesita información de su propio servicio. Para demostrar que los microservicios están relacionados entre sí, se muestra en el desglose de las funcionalidades cómo se realizan búsquedas que dependen de varios servicios, como por ejemplo la búsqueda versiones por la id de una entrada o la búsqueda de entradas por la id de una wiki.

De manera adicional, se ha añadido también el microservicio para implementar la media dentro de las entradas.

Las tablas y sus atributos de la base de datos son:

WIKIS	**ENTRADAS**	**VERSIONES**	**COMENTARIOS**	**MEDIA**
ID Title Description Category	ID Title Author CreatedAt WikiID	ID Content Editor CreatedAt EntryID	ID Content Rating CreatedAt Author VersionID	ID PublicID UploadUrl

Las tablas Entradas, Versiones y Comentarios hacen uso de IDs de otras entidades debido a que emplean búsquedas mediante el uso de las relaciones entre las entidades.

Las funcionalidades que se consideran necesarias en esta práctica son las siguientes:

- CRUD (creación, lectura, modificación y borrado) de las siguientes entidades:
    - Wiki
    - Entrada
    - Versión
    - Comentario

- Búsqueda parametrizada
    - Wiki
        - Buscar por id
        - Buscar por título (tanto exacto como parcial)
        - Buscar por categoría
        - Buscar por descripción
    - Entrada
        - Buscar por id
        - Buscar por título (tanto exacto como parcial)
        - Buscar por autor/es
        - Buscar por fecha
    - Versión
        - Buscar por id
        - Buscar por contenido
        - Buscar por editor
        - Buscar por fecha
    - Comentario
        - Buscar por id
        - Buscar por contenido
        - Buscar por puntuación
        - Buscar por fecha
        - Buscar por autor

- Búsqueda con uso de relaciones entre entidades
    - Entrada-Wiki: (*GetEntriesByWikiID*)
        - Buscar todas sus entradas
          En esta operación se buscan todas las entradas cuyo WikiID coincida con el de la wiki para la que estamos buscando las entradas.

    - Entrada-Version: (*GetVersionsByEntryID*)
        - Buscar todas sus versiones
          En esta operación se buscan las versiones cuyo EntryID coincida con el de la entrada para la que estamos buscando sus versiones.

    - Version-Comentario: (*GetCommentsByVersionID*)
        - Buscar todos sus comentarios
          En esta operación se buscan todos los comentarios cuyo VersionID coincida con el de la versión para la que estamos buscando los comentarios.

Adicionalmente, hemos añadido las siguientes funcionalidades:

- Búsqueda parametrizada:
    - Media
        - Buscar por id
        - Buscar por id publica

API REST

El esquema que hemos seguido es que una pasarela (gateway) es la que recibe las peticiones y se encarga de redirigirlas hacia la dirección correspondiente de cada microservicio. Las rutas son:

Rutas:

- /api
    - /wikis
        - GET /health
        - GET /
        - POST /
        - GET /exactTitle
        - GET /title
        - GET /description
        - GET /category
        - /id
            - GET /
            - PUT /
            - DELETE /
    - /entries
        - GET /health
        - GET /
        - POST /
        - GET /exactTitle
        - GET /title
        - GET /author
        - GET /date
        - GET /wiki
        - DELETE /wiki
        - /id
            - GET /
            - PUT /
            - DELETE /

- /comments
  - GET /health
  - GET /
  - POST /
  - GET /content
  - GET /rating
  - GET /date
  - GET /author
  - GET /version
  - DELETE /version
  - /id
    - GET /
    - PUT /
    - DELETE /
- /versions
  - GET /health
  - GET /
  - POST /
  - GET /content
  - GET /editor
  - GET /date
  - GET /entry
  - DELETE /entry
  - /id
    - GET /
    - PUT /
    - DELETE /
- /media
  - GET /health
  - GET /
  - POST /
  - GET /pubid
  - /id
    - GET /
    - PUT /
    - DELETE /

Descripción y prueba de los servicios creados

- Se adjunta un archivo json de un conjunto de pruebas Postman.
- Después de inicializar la aplicación se puede ver la documentación y probar la API en http://localhost:8000/swagger/.

Contenedores

Se ha decidido encapsular cada microservicio en un contenedor docker, desplegando todos de forma simultánea mediante el uso de docker compose y se está usando una base de datos mongo en la nube (MongoDB Atlas).

Por lo general se usaría una base de datos para cada microservicio, pero por simplicidad para la práctica y por recomendación del profesor, hemos usado solo uno. Sin embargo, cada microservicio solo puede acceder a su tabla en la base de datos, y en caso de que se necesite algo de una tabla a la que un microservicio no tiene acceso, este le enviará una petición HTTP al microservicio correspondiente.