
2º CFGS Desarrollo de Aplicaciones Web

Módulo Profesional Bases de Datos noSQL y Frameworks JavaScript

Base de datos Blog

1.- Enunciado

Se desea diseñar una base de datos para un blog en Internet según los siguientes supuestos:

1. Hay un conjunto de usuarios registrados, los cuales pueden publicar artículos y hacer comentarios a los mismos. De un usuario registramos: el email, la clave, nombre, apellidos y la fecha de nacimiento.
2. Cada usuario tiene un perfil, formado por su identificador en las principales redes sociales, la fecha en que se registró, la fecha del último acceso.
3. Además, en el perfil también se guarda el rol del usuario (autor, lector, admin, ...) y las preferencias (el color del tema y el idioma).
4. Los usuarios escriben artículos en el blog. Cada artículo tiene un título, fecha, texto, cuantas veces se ha visitado, un conjunto de etiquetas, un conjunto de categorías y un conjunto de comentarios. Además, como los artículos pueden ser editados por sus autores, también guardará las fechas de cada versión.
5. De cada etiqueta queremos saber su nombre y una url que permite consultar todos los artículos que tengan asignada dicha etiqueta.
6. Lo mismo ocurre con una categoría. Disponemos de su nombre y una url para consultar los artículos de esa categoría.
7. Cada comentario almacena el texto y la fecha en que se produjo.
8. Los usuarios podrán reaccionar a los artículos con *like* o *dislike*.

Según todo lo anterior:

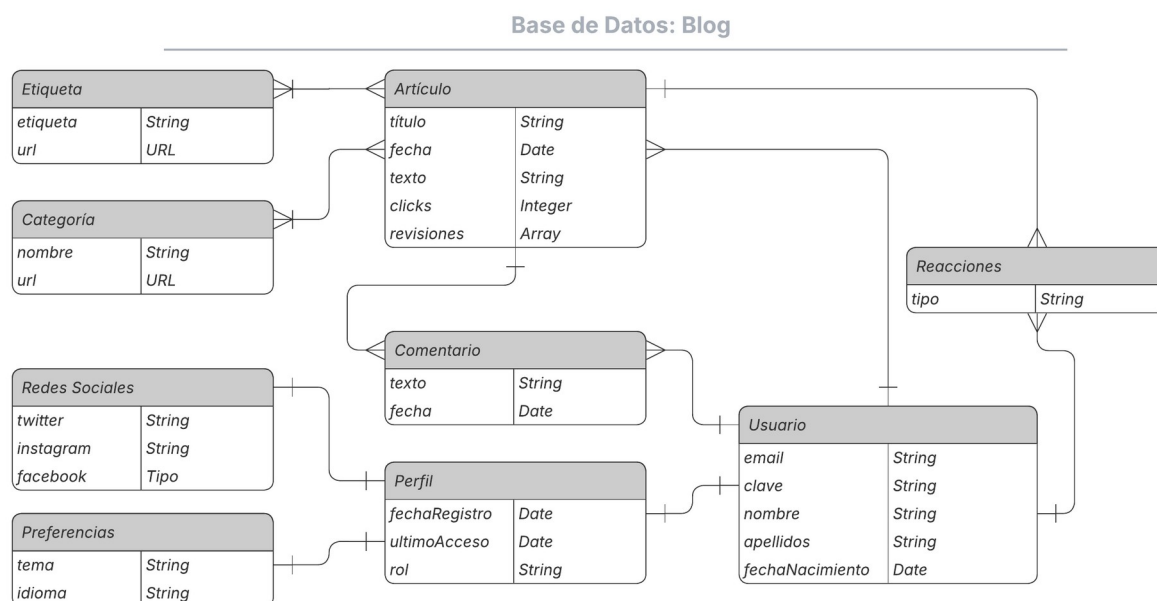
1. Identificar una posible carga de trabajo.
2. Realizar un mapa de relaciones entre las entidades del dominio del problema.
3. Establecer que datos se incrustan y que datos se referencian.

2.- Carga de trabajo

Se realiza el siguiente esbozo de las posibles acciones que realiza la aplicación.

Acción	Tipo	Información	Frecuencia	Prioridad
Crear un artículo	Write	Usuarios, artículos, etiquetas, categorías	10 al día	Alta
Comentar un artículo	Write	Usuarios, comentarios	100 al día	Media
Leer un artículo	Read	Artículos, etiquetas, usuarios categorías, comentarios	1000 al día	Alta
Leer artículos de una etiqueta	Read	Artículos, usuarios	100 al día	Media
Leer artículos de una categoría	Read	Artículos, usuarios	100 al día	Media
Reaccionar a un artículo	Write	Artículos	100 al día	Media

3.- Mapa de relaciones



Una vez se dispone del mapa de relaciones y la carga de trabajo se define la siguiente forma de relacionar los datos:

1. Una colección para los artículos que almacena:
 - a) Las claves `título`, `fecha`, `texto`, `clicks` y `revisiones`
 - b) Una clave `etiquetas` que consiste en un array de documentos incrustados con las etiquetas. Cada documento contendrá las claves: `etiqueta`, `url`.

- c) Una clave `categorias` que consiste en un array de documentos incrustados con las categorías. Cada documento contendrá las claves: `nombre`, `url`.
 - d) Una clave `comentarios` que consiste en un array de documentos incrustados con los comentarios del artículo. Cada documento contendrá las claves `texto`, `fecha` y `usuario` con una referencia a la colección de usuarios
 - e) Una clave `usuario` con una referencia al documento de usuarios.
2. Una colección para los `usuarios` que almacena:
- a) Las claves `email`, `nombre`, `apellidos`, `clave` y `fechaNacimiento`. Además tendrá una clave `perfil` que consiste en un documento incrustado que contiene las claves `fechaRegistro`, `ultimoAcceso`, y `rol`.
 - b) El documento con el perfil incluye también un documento incrustado `redesSociales`, con una clave por cada red social del usuario, y otro con `preferencias`, con dos claves: `tema` (claro u oscuro) e `idioma`.
3. Una colección para las reacciones de los usuarios a los artículos que almacena:
- a) Una clave `articulo_id` con una referencia a la colección artículos.
 - b) Una clave `usuario_id` con una referencia a la colección usuarios.
 - c) Una clave `tipo` (*like* o *dislike*).

4.- Implementación física

Una vez instalada la base de datos MongoDB podemos comenzar a crear nuestra BD definiendo las colecciones de la misma. Además, incluiremos algunos documentos para realizar pruebas. El archivo `blog.js` proporcionado contiene la creación de las colecciones y la inserción de algunos datos de prueba. Ejecutaremos este archivo con MongoShell.

Como uno de ellos incluye una clave cifrada tendremos que instalar previamente la biblioteca `bcrypt` que nos permitirá crear claves cifradas. En una ventana de terminal ejecutamos lo siguiente.

```
$ sudo npm install bcrypt
```

Damos por hecho que ya hemos creado el usuario `blog` en la base de datos `blog`. Con este usuario abrimos una conexión con MongoSH. Será con este usuario con el que abramos sesión y ejecutemos el script `blog.js`.

```
$ mongosh mongodb://blog:usuario@localhost/blog < blog.js
```