# Práctica Bases de Datos I

## *Diseño e implementación de una BD*


Alumnos: Carlos Marcos Guillem
Eugenio Pablo Murillo Solanas
Enrique Robles Uriel


Madrid, 10 de enero de 2021 Curso: 2020/21

Asignatura: **BASES DE DATOS I** Grupo: 01

Profesores: Guillermo de la Calle Velasco Grado en Ingeniería de Sistemas de la Información

# Índice

# Introducción

Este primer (y único) trabajo de la asignatura, tiene como finalidad afianzar y poner en práctica todos los conocimientos adquiridos a lo largo del curso. Además busca hacer esto introduciendo al alumno de forma práctica al funcionamiento real de una base de datos, con todos los procesos de diseño, aplicación y uso que toman parte durante su periodo de vida.

Para alcanzar estos objetivos, se plantea la creación y la posterior consulta de una base de datos relacional sencilla y de tema libre a través del programa gestor *SQLite*. De esta manera, los alumnos en primer lugar diseñamos de manera conceptual y lógica la base de datos, empleando los diagramas entidad-relación y el paso a tablas (intercambiando proyectos con los compañeros), para posteriormente aplicar físicamente el diseño mediante SQLite y familiarizarnos con el uso del mismo a base de realizar consultas, actualizaciones, borrados, vistas, etc.

# Especificación de requisitos

Siguiendo con las recomendaciones del profesor, el tema que escogimos para nuestra base de datos está completamente ligado con la Práctica 3 de la otra asignatura que este nos imparte, Metodología y Tecnología de la Programación. Esto se debe a que necesitamos de una base de datos a la que conectarnos desde una aplicación de escritorio en dicha práctica y, como está claro, usando la que hemos hecho para este trabajo nos ahorra tiempo y esfuerzo en el otro.

De esta manera, puesto que nuestra aplicación de escritorio consistirá en un periódico (tal y como antiguamente existían en vez de sus versiones web), nuestra base de datos contendrá la información relevante para el mismo: secciones, temas, noticias, imágenes… Además, contará con un sistema de inicio de sesión por usuarios que podrán modificar su perfil, comentar en las noticias, suscribirse a secciones, etc.
Todo ello queda explicado en el siguiente enunciado que, tal y como el propio planteamiento de la práctica nos exigía, redactamos para especificar los requisitos de nuestra base de datos y con el fin de pasárselo a otro de los grupos, quienes nos harían una primera versión del diagrama entidad-relación (así como nosotros hicimos con otro grupo):

*Se desea crear una base de datos para almacenar y gestionar las noticias y usuarios de un periódico digital. Para ello, se requiere crear un diagrama Entidad-Relación teniendo en cuenta las siguientes especificaciones:*

- *El periódico se divide en varias secciones para clasificar las noticias, cada una de ellas diferenciándose de las demás por su nombre. Además, cada sección posee una edad mínima recomendada para la lectura de los artículos contenidos en la misma.*

- *Cada sección contiene como mínimo un tema que, de nuevo, se diferencia del resto de temas por su nombre. Cada tema solo puede pertenecer a una única sección.*

*Todos los temas contienen al menos una noticia; sin embargo, una misma noticia puede encontrarse contenida en más de un tema incluso de diferentes secciones.*

- *Cada noticia se distingue de las demás por su título y fecha de publicación, pero también se requiere guardar el subtítulo de la noticia, su cuerpo y un lugar relacionado con la misma (localización del suceso acontecido, localidad sede de la redacción...).*

  *Con el objetivo de aumentar el número de visitas, se desea que toda noticia tenga otras noticias relacionadas, pero a su vez una noticia en particular no tiene por qué estar relacionada con otras noticias.*

- *Todas las noticias son escritas por un único autor registrado como tal, del cual se almacenan su nombre, apellidos y DNI. Ya que existen autores freelancer o de opinión no contratados por el periódico, un autor tiene que haber escrito una noticia como mínimo para que se almacene en la base de datos.*
- *Las noticias pueden contener archivos multimedia. Estos se distinguen unos de otros por su ruta de enlace, pero se necesita también conocer su extensión. Además, cada vez que un archivo multimedia sea referenciado en una noticia, se necesita saber si este es portada de la misma o no. Para que se guarde un archivo es necesario que tenga como mínimo una inserción en alguna noticia.*

- *Por otro lado, dentro de la base de datos de nuestro periódico se quieren guardar distintos perfiles que se puedan crear los usuarios. Todos los usuarios se distinguen por su correo electrónico y su nickname, no pudiendo existir más de uno bajo el mismo email o alias. De dicho usuario se quiere conocer también una contraseña para su inicio de sesión, su fecha de nacimiento, una foto de perfil (que se almacenará como archivo multimedia y que podrían compartir varios usuarios) y un atributo que nos indique si desea recibir notificaciones de las secciones a las que sigue.*

  *De este último requisito se extrae que un usuario puede seguir varias secciones (aunque no se le obliga a suscribirse a ninguna al*

*registrarse en el periódico). A su vez, una sección puede no tener ningún seguidor.*

- *Un usuario puede comentar en tantas noticias como quiera. De los comentarios escritos se quiere saber la fecha en la que se escribieron, así como el texto que contienen. Todos los comentarios se diferenciarán con un identificador único para facilitar su búsqueda y gestión por parte de los administradores.*

  *Además, se quiere conocer si este comentario fue escrito en respuesta de otro. No obstante, un comentario puede tanto no responder a ningún otro como no contar de ninguna respuesta.*
  *Por último, una noticia puede tener numerosos comentarios aunque no es necesario que contenga ninguno.*

# Diseño conceptual

El diagrama entidad-relación que nos hizo el grupo correspondiente para nuestra base de datos fue expuesto ante el resto de alumnos un día de clase, pero nunca lo llegamos a recibir, por lo que no lo podemos incluir en este apartado. En su lugar, incluimos el que nosotros hicimos para el grupo 7:

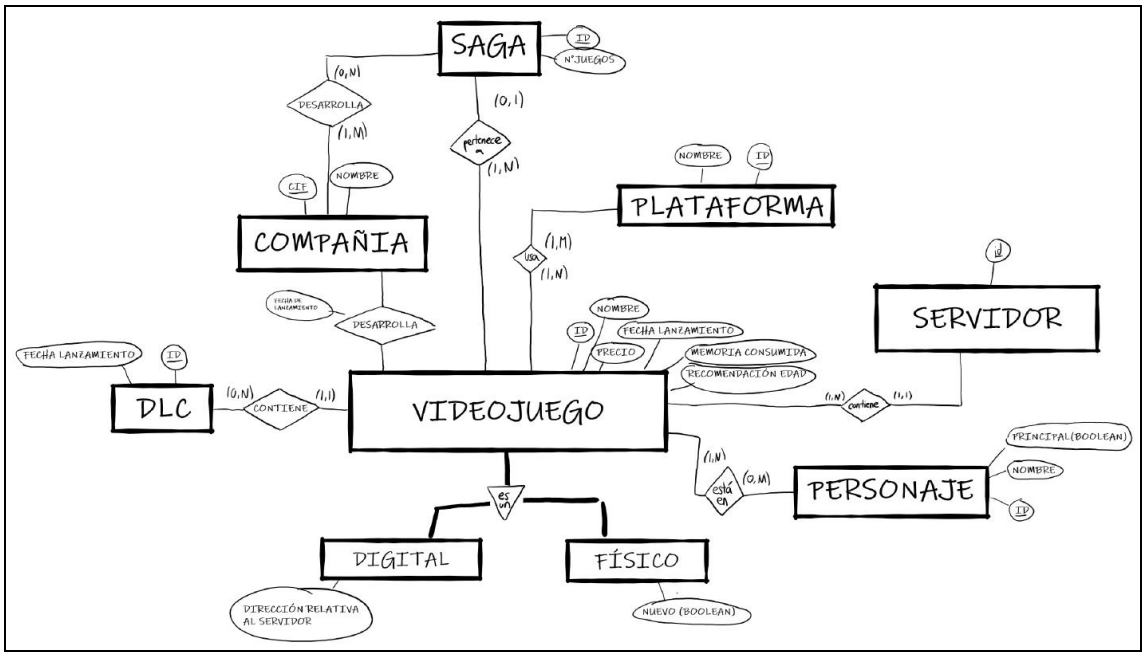


*Diagrama entidad-relación que hicimos para el grupo 7. Fuente: realización propia a través de Whiteboard.*

No obstante, el diagrama que nos plantearon era bastante deficiente en comparación a los requisitos especificados en nuestro enunciado: faltaban relaciones claramente explícitas en el enunciado como la de las noticias relacionadas o la de las respuestas a comentarios, existía una entidad que no debería para el periódico en su totalidad…

Al final decidimos casi empezar desde cero el diagrama, conservando entidades, atributos y relaciones principales, que el otro grupo sí había conseguido distinguir a partir de nuestro enunciado. Suprimimos la entidad sobrante, añadimos relaciones reflexivas para las noticias y los comentarios, etc. Conseguimos acabar con un diagrama que satisfacía todas nuestras necesidades y expectativas:

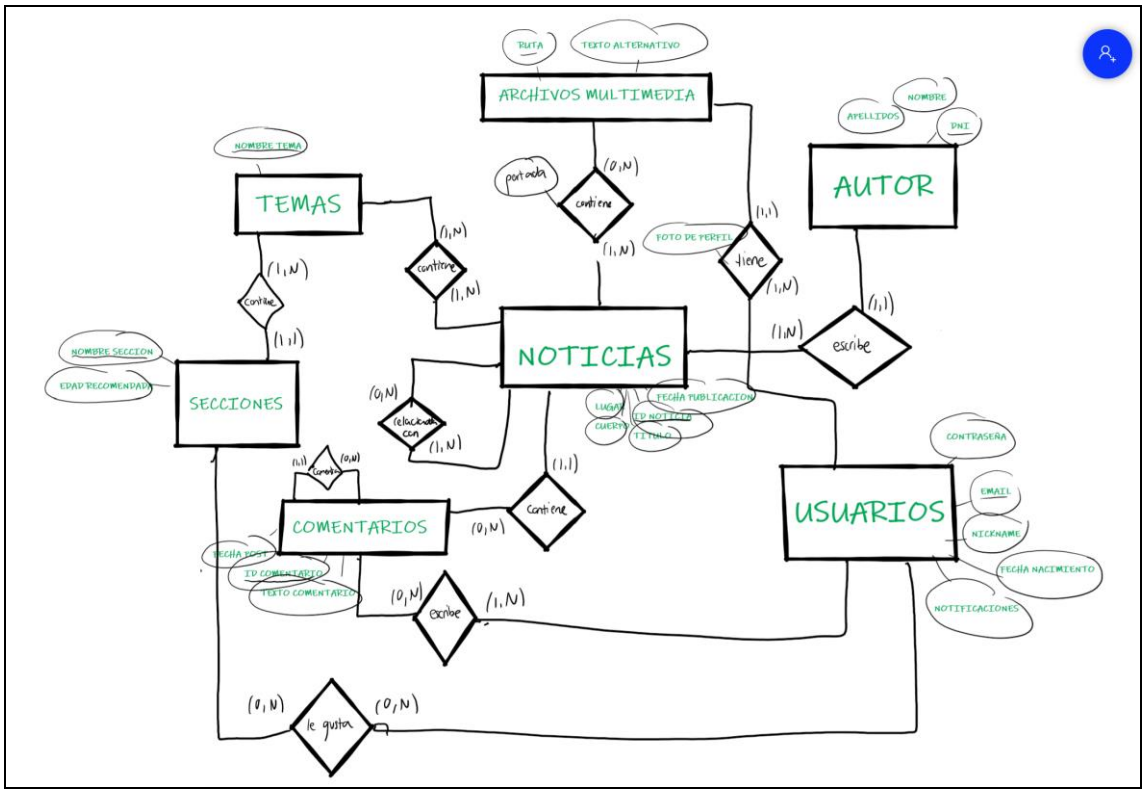


*Diagrama entidad-relación de nuestra base de datos. Fuente: realización propia a través de Whiteboard.*

# Diseño lógico

Puesto que ya hemos explicado nuestro diagrama entidad-relación en el apartado anterior, dejamos a continuación nuestro paso a tablas, hecho de acorde a la transformación de relaciones vista en clase, por lo que incluye claves foráneas (marcadas en rojo) para las relaciones de cardinalidades máximas 1:1 y 1:N y nuevas tablas para las N:N y debería estar normalizada al tercer orden:

**Archivos multimedia(**ruta, texto_alternativo**)**
**Autores(**nombre, apellidos, dni**)**
**Usuarios(**email, nickname, contraseña, fecha_nacimiento, notificaciones, foto_perfil**)**
**Secciones(**nombre_seccion, edad_recomendada**)**
**Temas(**nombre_tema, nombre_seccion**)**
**Noticias(**id_noticia, titulo, fecha_publicacion, lugar, cuerpo, dni_autor**)**
**Comentarios(**id_comentario, comentario, fecha_post, email_usuario, id_noticia, id_respuesta**)**
**Secciones favoritas(**id_relacion, email, nombre_seccion**)**
**Temas_noticias(**id_relacion, nombre_tema, id_noticia**)**
**Noticias relacionadas(**id_relacion, id_noticia1, id_noticia2**)**
**Multimedia_noticias(**id_relacion, ruta, id_noticia, portada**)**

# Diseño físico

Durante esta última etapa de la creación de la base de datos es cuando empezamos a utilizar SQLite por primera vez para implementar el paso a tablas de la etapa anterior.

Gracias a que hicimos un buen trabajo a la hora de transformar las relaciones y a que lo realizamos siguiendo el método presente en las transparencias de la asignatura que vimos en clase, no tuvimos que llevar a cabo ningún tipo de cambio ni normalización. Es decir, las tablas de nuestra base de datos son calcadas a las de la lista anterior.
Esto se puede comprobar en el script adjunto a esta memoria llamado *creación.sql*.
Sobre la fase de creación cabe destacar que:

- Utilizamos las cláusulas UNIQUE y NOT NULL de forma correcta y acorde a los requisitos de nuestro enunciado.
- Empleamos IF NOT EXISTS como medida de seguridad a la hora de crear las tablas.
- Comprobamos que la longitud del texto para el DNI de los autores fuera menor o igual que 9 caracteres, como ocurre en la realidad.
- Creamos dos vistas que agruparan los valores de varias entidades relacionadas a través de tablas de relación para facilitar las consultas de la última etapa del trabajo.

La carga de datos se encuentra en el script *carga.sql*.

# Consultas

En este último apartado de la práctica, tal y como su propio nombre indica, nos hemos limitado a incluir aquellas consultas que cumplen con los requisitos de la práctica, pues un periódico online es una aplicación ambiciosa, que puede ejecutar centenares de consultas y actualizaciones diferentes, sobre todo si cuenta con un motor de búsqueda con filtros por ejemplo.

Se utiliza para ver las secciones favoritas asociadas a un usuario:
*SELECT nombre_seccion FROM "secciones favoritas" WHERE email = "correo@correo.com";*

Se utiliza para hacer estadisticas anonimas de cuantos usuarios tienen cierta sección como favorita:
*SELECT distinct count(email) FROM "secciones favoritas" WHERE nombre_seccion = "Economía";*

Se utiliza para cargar los datos del usuario al iniciar sesión:
*SELECT * from "usuarios" WHERE email = "correo@correo.com";*

Se utiliza para modificar la foto de perfil o eliminar si es null:
*UPDATE OR FAIL usuarios SET foto_perfil = "prueba" WHERE email = "correo@correo.com";*

Se utiliza para modificar la contraseña:
*UPDATE OR FAIL usuarios SET contraseña = "contraseña" WHERE email = "correo@correo.com";*

Se utiliza para modificar el valor de notificaciones:
*UPDATE OR FAIL usuarios SET notificaciones = "prueba" WHERE email = "correo@correo.com";*

Se utiliza para modificar el nickname:
*UPDATE OR FAIL usuarios SET nickname = "nickname" WHERE email = "correo@correo.com";*

Se utiliza para obtener todas las secciones:
*SELECT nombre_seccion FROM secciones;*

Se utiliza para obtener todos los temas de una sección:
*SELECT nombre_tema FROM temas WHERE nombre_seccion = "Deporte";*

Se utiliza para borrar un comentario:
*DELETE FROM comentarios WHERE id_comentario = 3;*

Se utiliza para banear/borrar un usuario que ha hecho un comentario negativo:
*DELETE FROM usuarios WHERE email_usuario = SELECT email_usuario FROM comentarios WHERE id_comentario=1);*

Se utiliza para borrar todas las noticias de la base de datos (ejecutar solo en caso de emergencia) :
*DROP TABLE noticias;*

Se utiliza para obtener algunos campos de todos los comentarios de un usuario:
*SELECT comentario,fecha_post,id_noticia,id_respuesta FROM comentarios WHERE email_usuario = "correo@correo" ORDER BY comentario;*

Se utiliza para obtener la ruta de la portada de una noticia, el título de la noticia y el id de la noticia:
*SELECT ruta,titulo,multimedia_noticias.id_noticia FROM multimedia_noticias inner join noticias on noticias.id_noticia = multimedia_noticias.id_noticia WHERE noticias.id_noticia = 1 and portada = 1;*

Se utiliza para obtener el título y la ruta de la portada de las noticias relacionadas con otras:
*SELECT ruta, titulo FROM imagenes_noticia INNER JOIN "noticias relacionadas" ON portada = 1 and id_noticia = "noticias*

*relacionadas".id_noticia2 WHERE "noticias relacionadas".id_noticia1 = 3;*

Se utiliza para obtener el nombre y apellidos del autor de una noticia:
*SELECT group_concat(nombre, apellidos) AS Autor FROM autores INNER JOIN noticias WHERE autores.dni = noticias.dni_autor and id_noticia = 1 ORDER BY apellidos;*

Se utiliza para obtener todas las noticias de un autor limitándolo a un número X de obras(por si fuera muy extensa la obra del autor) :
*SELECT id_noticia, titulo FROM noticias INNER JOIN autores ON nombre = "Eugenio Pablo" and apellidos = "Murillo Solanas" WHERE autores.dni = noticias.dni_autor limit 50;*

Se utiliza para saber el número de comentarios de una noticia:
*SELECT count(comentario) AS "Comentarios" FROM comentarios where id_noticia = 1;*

Se utiliza para obtener los comentarios de una noticia:
*SELECT comentario,usuarios.nickname,fecha_post,foto_perfil FROM comentarios INNER JOIN usuarios ON id_noticia = 1 WHERE usuarios.email = comentarios.email_usuario GROUP BY usuarios.nickname;*

Se utiliza para obtener la noticia que ha sido introducida en la base de datos más recientemente:
*SELECT titulo, cuerpo FROM noticias WHERE id_noticia=(select max(id_noticia) FROM noticias);*

Se utiliza para buscar los usuarios que tengan un nickname que empiece por una letra determinada:
*SELECT nickname FROM usuarios WHERE nickname LIKE "q%";*

Se utiliza para saber cuáles son las secciones y temas con una edad recomendada menor a X:
*SELECT * FROM secciones_temas INNER JOIN nombre_seccion HAVING edad_recomendada<=8;*