

JDBC y Swing

Antes de nada debemos crear un contenedor docker con **MariaDB**. Para ello podemos usar el siguiente docker-compose.yml

version: '3'

services:

mariadb:

image: mariadb

container_name: mariadb

volumes:

- mariadb_data_container:/var/lib/mysql

restart: always

environment:

MYSQL_ROOT_PASSWORD: root

MYSQL_USER: root

ports:

- "3310:3306"

networks:

- mariadb-network

networks:

mariadb-network:

volumes:

mariadb_data_container:

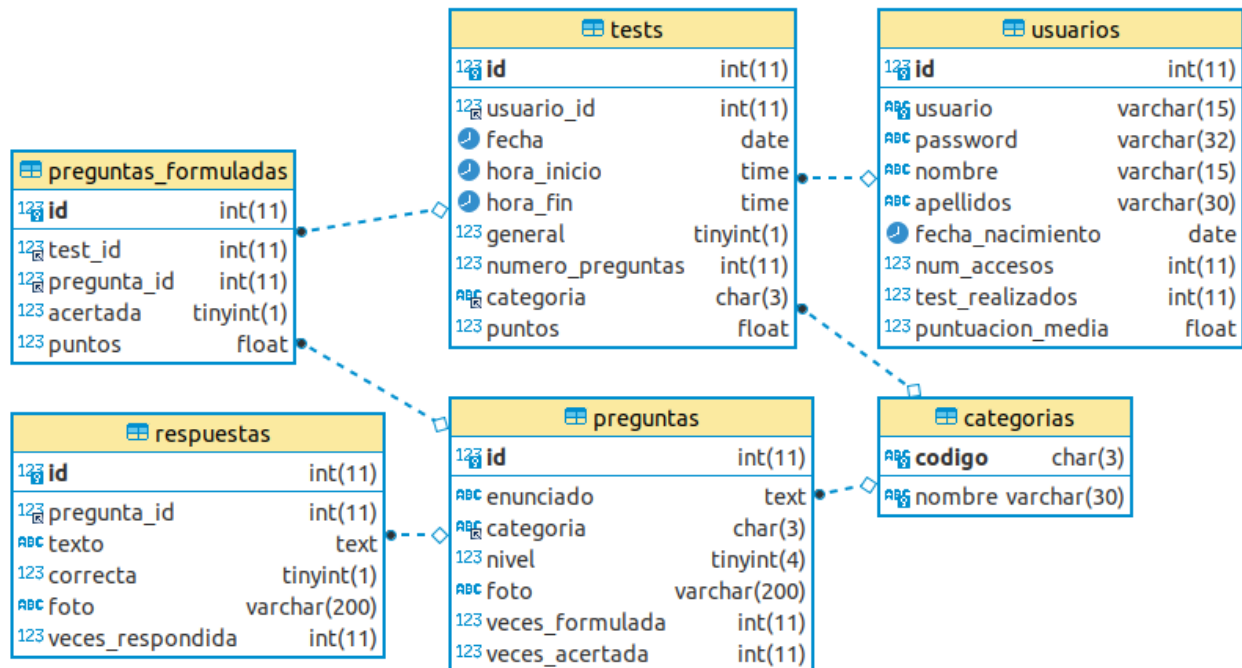
En los ejercicios de esta actividad tienes que usar la base de datos MariaDB **preguntastest**. Para ello crearemos la base de datos con ese nombre desde Dbeaver, por ejemplo.

Luego descargaremos el script preguntastest.sql y ejecutaremos este comando desde la carpeta donde lo tengamos:

docker exec -i mariadb mysql -uroot -proot preguntastest < preguntastest.sql

Hoja04_BDRelacionales_06

Una vez importada la base de datos podemos observar su esquema relacional:



La tabla **categorías** tiene información sobre los temas posibles de las preguntas: ART-Arte y cultura, CIE-Ciencias, DEP-Deportes, ECO-Economía, FIL-Filosofía, etc.

La tabla **preguntas** contiene datos sobre una batería de preguntas de test.

- El id es autoincrementado por lo que nunca se le debe asignar un valor.
- El nivel es 1, 2 o 3 (de momento todas las preguntas tendrán nivel 1, así que ese campo no lo usaremos)
- La columna **ruta_foto** contiene la ruta de un posible archivo de imagen que se usaría en la pregunta.

La tabla **usuarios** contiene información de los usuarios que se registran para realizar test. La columna **id** es autoincrementada.

La tabla **respuestas** contiene las respuestas que se mostrarán en el test para cada pregunta. En la columna **correcta** se almacena un uno cuando es una respuesta correcta y un cero cuando no lo es.

EJERCICIO 1

Realizar un programa para registrarte como usuario en **preguntastest**. Sólo debe pedir el nombre, los apellidos, la fecha de nacimiento, el usuario de acceso y la contraseña.

Debes usar una clase **Conexion** singleton.

Hoja04_BDRelacionales_06

Para gestionar los datos de un usuario, debes desarrollar una clase POJO **Usuario** con los datos miembro correspondientes a las columnas de la tabla **usuarios**.

La contraseña hay que guardarla con la función **MD5** de MySQL.

EJERCICIO 2 (en el mismo proyecto que el 1)

Antes de desarrollar el programa crear un fichero CSV con datos de preguntas y respuestas. Los datos de cada pregunta están separados por el carácter punto y coma (;). Las preguntas se identificarán con el carácter P y las respuestas con la R.

De cada pregunta se indicará el enunciado, la categoría y el nivel.

De cada respuesta se indicará el texto y si es correcta o no.

Un ejemplo de una pregunta es el siguiente:

```
P;¿Cuál era el nombre original de la ciudad de Nueva York?;HIS;1
R;Nueva Ámsterdam;1
R;La gran manzana;0
R;Empire State;0
R;Gotham;0
P;¿En qué país se celebraron los primeros Juegos Olímpicos?;DEP;1
R;Italia;0
R;Grecia;1
R;Japón;0
R;Francia;0
```

Se debe crear una clase Pregunta con los atributos necesarios. Además, cada pregunta tendrá una colección de objetos de tipo Respuesta.

Se puede crear una paquete util que contenga una clase GestorFicheros que contenga un método leerPreguntasCSV que devuelva una colección de preguntas.

Después, esas preguntas y respuestas serán cargadas en la base de datos.

En la clase Principal haz un menú con las dos opciones que tenemos hasta ahora:

1. Insertar usuario
2. Cargar fichero CSV

En esta opción 2 deberemos solicitar la ruta del fichero CSV a cargar

Hoja04_BDRelacionales_06

EJERCICIO 3 (en el mismo proyecto que los anteriores)

Realizar un programa que muestra el texto de 5 preguntas de la categoría indicada por el usuario (el código) y, a continuación de cada pregunta, los textos de cada respuesta.

Si la categoría indicada tiene menos de 5 preguntas las mostrará todas. Si la categoría indicada tiene más de 5 preguntas elegirá aleatoriamente 5.

Por cada pregunta mostrada habrá que pulsar ENTER para pasar a ver la siguiente.

Añadir al menú la opción 3. Visualizar preguntas de una categoría

EJERCICIO 4 (en el mismo proyecto que los anteriores)

Realiza un programa que pide un nombre usuario y contraseña para acceder a la base de datos **preguntastest**. Hay que tener en cuenta que la contraseña está guardada con MD5.

Si las credenciales son correctas, se muestra un mensaje indicando que se ha iniciado una sesión de usuario, el nombre y apellidos del usuario y:

- Se contabiliza un acceso más del usuario.
- Se formulan al usuario 4 preguntas aleatorias. Por cada pregunta se escriben sus respuestas.
- Por cada pregunta se pide al usuario que responda cuál es la correcta y se le responde si ha acertado o no.
- Por cada pregunta, se registra que ha sido formulada una vez más y, en su caso, que fue acertada la respuesta.
- Para la respuesta respondida se registra que ha sido respondida una vez más.
- Se contabiliza que el usuario ha realizado un test más.

Añadir al menú la opción 4. Jugar