

INF 324 - Informe de desarrollo primer parcial

Jesus Rodolfo Izurieta Veliz

14 de julio de 2020

Para este proyecto usaremos la biblioteca codeigniter para desarrollar en php, para la base de datos usaremos mariadb, un motor de base de datos open source basado en mysql. El proyecto estará organizado en contenedores docker, para facilitar su distribución y replicar su funcionamiento sin tener que lidiar con problemas de versiones ni configuraciones extra para el funcionamiento del proyecto.

1. Configuración del proyecto

La estructura del proyecto estará definida por el archivo `docker-compose.yml`, en el que indicaremos las imágenes de docker que se usarán para el proyecto, en este caso, usaremos contenedores de codeigniter y mariadb.

```
version: '2'
services:
  myapp:
    image: 'docker.io/bitnami/codeigniter:3-debian-10'
    ports:
      - '8000:8000'
    volumes:
      - './app'
    depends_on:
      - mariadb
  mariadb:
    image: 'docker.io/bitnami/mariadb:10.3-debian-10'
    ports:
      - '3306:3306'
```

```
volumes:
  - './mysql:/bitnami/mariadb'
environment:
  - ALLOW_EMPTY_PASSWORD=yes
```

2. Base de datos

Después de inicializar el proyecto con docker, necesitaremos crear la base de datos, nos conectamos al contenedor de mariadb con el comando:

```
docker exec -it 65bdc8d72f1 bash
```

Donde 65bdc8d72f1 es el código del contenedor, que puede obtenerse con el comando `docker ps` cuando el proyecto está ejecutándose.

3. Creación de tablas

En la línea de comandos, iniciamos una sesión en mysql con el comando:

```
mysql -u root -p
```

Se nos pedirá una contraseña, que será la que tenemos configurada en el archivo `docker-compose.yml` o en su defecto, nos permitirá ingresar sin una contraseña la primera vez.

Creamos un usuario con el comando

```
GRANT ALL PRIVILEGES ON *.* TO 'username'@'localhost' IDENTIFIED BY 'password';
```

Poniendo el nombre del usuario en lugar de «username» y la contraseña en lugar de «password».

En este caso creamos el usuario admin.

Ahora saldremos de mysql con `\q`, y volveremos a ingresar con el usuario y contraseña recientemente creados.

```
mysql -u admin -p
```

Ahora crearemos la base de datos y la seleccionaremos para crear las tablas en ella.

```
CREATE DATABASE academico; USE academico;
```

ahora crearemos las siguientes tablas:

Tabla USUARIO

Campo	Tipo de dato	Longitud	PK	Descripción
pk	int		si	Llave primaria
dni	int			CI del usuario
password	varchar	40		Clave para ingresar
photo	varchar	200		Url de la fotografía

Con la consulta:

```
CREATE TABLE usuario
(
    pk int unsigned not null auto_increment,
    dni varchar(20) not null,
    password varchar(40) not null,
    photo varchar(200),
    primary key (pk)
);
```

Tabla IDENTIFICADOR:

Campo	Tipo de dato	Longitud	PK	Descripción
pk	int		si	Llave primaria
user _{pk}	int			Llave foránea de usuario
dni	int	20		CI del usuario
first _{name}	varchar	100		Nombres
last _{name}	varchar	100		Apellidos
birth _{date}	date			Fecha de nacimiento
residence	varchar	10		Código del lugar de residencia

Con la consulta:

```
CREATE TABLE identificador
(
    pk int unsigned not null auto_increment primary key,
    user_pk int unsigned,
    dni varchar(20) not null,
    first_name varchar(100),
    last_name varchar(100) not null,
```

```

    birth_date date,
    residence varchar(10),
    foreign key (user_pk) REFERENCES usuario(pk)
);

```

Tabla NOTAS

Campo	Tipo de dato	Longitud	PK	Descripción
pk	int		si	Llave primaria
user_pk	int			Llave foránea de usuario
matter	varchar	50		Materia
score	int			Calificación

Con la consulta:

```

CREATE TABLE notas
(
    pk int unsigned not null auto_increment,
    user_pk int unsigned,
    dni varchar(20) not null,
    matter varchar(50),
    score int,
    primary key (pk),
    foreign key (user_pk) REFERENCES usuario(pk)
);

```

Podremos comprobar que las tablas se crearon con el comando `show tables;`, que nos mostrará una lista de las tablas de nuestra base de datos.