

INF 324 - Informe de desarrollo primer parcial

Jesus Rodolfo Izurieta Veliz

16 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.

2.1. 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.

3. Desarrollo

El proyecto consta de dos secciones para resolver los dos problemas propuestos en el enunciado, la página de perfil de usuario, que mostrará la foto del usuario y la parte de notas, donde se mostrará el contenido de la tabla notas.

4. Modelos

Codeigniter nos permite abstraer los datos de una base de datos y las acciones que podemos realizar sobre ellas mediante modelos que definimos por cada tabla de la base de datos.

4.0.1. Usuario

Definimos el modelo usuario

5. Vistas

5.1. Inicio de sesión

Para el inicio de sesión se ingresa en la dirección raíz (/) y si no se tiene una sesión iniciada, se mostrará el formulario de inicio de sesión. En caso de tener una sesión, el usuario podrá acceder automáticamente a su página de perfil, donde se mostrará su foto y la opción de cambiar el color de la página.

Para esto, empezaremos por definir los modelos que representarán a las tablas de la base de datos. Se puede acceder a esta vista mediante la url `/login` para iniciar sesión, el sistema nos llevará a la url `/profile` si el inicio de sesión es exitoso

5.2. Lista de notas

La vista del listado de notas muestra la cantidad de aprobados en la base de datos `notas`, se contará en la consulta a cualquier entrada de la base de datos con una nota igual o mayor a 51. Se puede acceder a esta vista mediante la url `/scores`, donde se mostrará la cantidad de aprobados.

6. Pruebas

6.1. Perfil de usuario

Las consultas a la base de datos se realizaron usando los modelos que proporciona codeigniter, ya que es una opción más segura ante ataques de inyección sql por ejemplo, la consulta para el inicio de sesión, usando modelos de codeigniter es:

```
$query = $this->db
```

```
->get_where('usuario', array('dni' => $username, 'password' => $password));
```

Que sería equivalente a la consulta:

```
select * from usuario where dni = $username and password = $password;
```

Ingresamos a la dirección /login e ingresamos con el CI 9860616 y la contraseña 12345678x.

Inicio de sesión

Nombre de usuario

9860616

Contraseña


.....

Enviar


Inicia sesión con tu usuario y contraseña.

Luego de esto, iniciaremos sesión y podremos acceder a la página de perfil.

Perfil de usuario




izjr.jpg

Seleccione un color: Rojo 

INF 324

En la página de perfil veremos detalles del usuario y también un select que nos permitirá cambiar el color de la página, si seleccionamos otro color, veremos el cambio.

Perfil de usuario



izjr.jpg

Seleccione un color: Azul

INF 324

Podremos verificar los datos de usuarios en la base de datos:

```
select * from identificador;
```

pk	user_pk	dni	first_name	last_name	birth_date	residence
1	1	9860616	Jesus	Izurieta	NULL	LP

6.2. Notas

Para el conteo de notas, usando los modelos de codeigniter se usó la consulta:

```
$this->db->select('i.residence, count(i.residence) as aprobados');
$this->db->from('notas n');
$this->db->join('identificador i', 'i.user_pk = n.user_pk');
$this->db->where('n.score >=', 51);
$this->db->group_by('i.residence');
```

La consulta equivalente que resuelve el segundo enunciado sería:

```
select i.residence, count(i.residence) as aprobados
from notas n join identificador i on i.user_pk = n.user_pk
where n.score >= 51
group by i.residence;
```

Ingresamos a la dirección `/scores` podremos ver una lista de las materias aprobadas.

Notas
Cantidad de aprobados <ul style="list-style-type: none">• CBBA 1• LP 1
Inicia sesión con tu usuario y contraseña.

Podremos comprobar que el conteo es correcto comparando con la consulta directa a la base de datos:

```
MariaDB [academico]> select i.residence, count(i.residence) as aprobados from notas n join identificador i on i.user_pk = n.user_pk group by i.residence;
```

residence	aprobados
CBBA	1
LP	1