



Instituto Tecnológico del valle de Oaxaca



Manual de prácticas de la Materia de Framework para el desarrollo de aplicaciones Web

Presenta

Ambrosio Cardoso Jiménez

Práctica 4. Conexión del framework a una base de datos

Objetivo: Configurar la conexión del framework a una base de datos de mysql8

Requisitos:

- Tener instalado el framework y el sistema gestor de base de datos

Temas relacionadas.

2.4 Conexión a SGBD

Condiciones.

- Tener habilitado los servicios y permitir conexiones entrantes y salientes en el firewall

Procedimiento y guía ilustrativa

Paso 1: Entrar al contenedor de mysql8

```
docker exec -it mysql8 sh
```

Paso 2: Entrar a mysql con la cuenta de root

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

password: ***** Nota: La contraseña de root se ubica en el archivo

docker-compose.yml

Paso 3: Crear la BD, las tablas y sus relaciones con las siguientes instrucciones:

```
CREATE DATABASE bdlara CHARACTER SET utf8mb4
```

```
COLLATE utf8mb4_unicode_ci;
```

```
use bdlara;
```

```
show tables;
```

```
CREATE TABLE casilla
```

```
( id BIGINT AUTO_INCREMENT
```



```
, ubicacion VARCHAR (100) NOT NULL
, CONSTRAINT pkcasilla PRIMARY KEY(id)
);
CREATE TABLE candidato
( id BIGINT AUTO_INCREMENT
, nombrecompleto VARCHAR (200) NOT NULL
, foto VARCHAR (200)
, sexo CHAR
, perfil VARCHAR (200)
, CONSTRAINT pkcandidato PRIMARY KEY (id)
);
CREATE TABLE eleccion
( id BIGINT AUTO_INCREMENT
, periodo VARCHAR (100) NOT NULL
, fecha DATE
, fechaapertura DATE
, horaapertura TIME
, fechacierre DATE
, horacierre TIME
, observaciones TEXT
, CONSTRAINT pkeleccion PRIMARY KEY (id)
);CREATE TABLE funcionario
```



```
( id BIGINT AUTO_INCREMENT
, nombrecompleto VARCHAR (200) NOT NULL
, sexo CHAR
, CONSTRAINT pkfuncionario PRIMARY KEY (id)
);

CREATE TABLE rol
( id BIGINT AUTO_INCREMENT
, descripcion VARCHAR (100) NOT NULL
, CONSTRAINT pkrol PRIMARY KEY(id)
);

CREATE TABLE eleccioncomite
( id BIGINT AUTO_INCREMENT
, eleccion_id BIGINT
, funcionario_id BIGINT
, rol_id BIGINT
, CONSTRAINT pkeleccioncomite PRIMARY KEY (id)
, CONSTRAINT ukeleccioncomite UNIQUE KEY (eleccion_id,funcionario_id)
, CONSTRAINT fkfuncionario_eleccioncomite
FOREIGN KEY (funcionario_id)
REFERENCES funcionario (id)
, CONSTRAINT fkrol_eleccioncomite
FOREIGN KEY (rol_id)
```



```
REFERENCES rol (id)
, CONSTRAINT fkeleccion_elecomite
FOREIGN KEY (eleccion_id)
REFERENCES eleccion (id)
);
CREATE TABLE voto
( id BIGINT AUTO_INCREMENT
, eleccion_id BIGINT
, casilla_id BIGINT
, evidencia VARCHAR (200)
, CONSTRAINT pkvoto PRIMARY KEY (id)
, CONSTRAINT ukvoto UNIQUE
(eleccion_id, casilla_id)
, CONSTRAINT fkeleccion_voto
FOREIGN KEY (eleccion_id)
REFERENCES eleccion (id)
, CONSTRAINT fkcasilla_id
FOREIGN KEY (casilla_id)
REFERENCES casilla (id)
);CREATE TABLE votocandidato
( voto_id BIGINT
, candidato_id BIGINT
```



```
, votos INT
, CONSTRAINT pkvotocandidato
PRIMARY KEY(voto_id, candidato_id)
, CONSTRAINT fkvoto_vc FOREIGN KEY (voto_id)
REFERENCES voto (id)
, CONSTRAINT fkcandidato_vc
FOREIGN KEY (candidato_id)
REFERENCES candidato (id)
);
CREATE TABLE funcionariocasilla
( id BIGINT AUTO_INCREMENT
, funcionario_id BIGINT
, casilla_id BIGINT
, rol_id BIGINT
, eleccion_id BIGINT
, CONSTRAINT pkfc PRIMARY KEY (id)
, CONSTRAINT ukfc
UNIQUE KEY (funcionario_id, eleccion_id)
, CONSTRAINT fkfunc_fc FOREIGN KEY (funcionario_id)
REFERENCES funcionario(id)
, CONSTRAINT casilla_fc FOREIGN KEY (casilla_id)
REFERENCES casilla (id)
```



```
, CONSTRAINT fkrol_fc FOREIGN KEY (rol_id)
REFERENCES rol (id)
, CONSTRAINT fkeleccion_fc FOREIGN KEY (eleccion_id)
REFERENCES eleccion (id)
);
CREATE TABLE imeiautorizado
( id BIGINT AUTO_INCREMENT
, funcionario_id BIGINT
, eleccion_id BIGINT
, casilla_id BIGINT
, imei VARCHAR(20) NOT NULL
, CONSTRAINT pkimei PRIMARY KEY (id)
, CONSTRAINT ukimei UNIQUE KEY (funcionario_id, eleccion_id)
, CONSTRAINT fkfun_imei FOREIGN KEY (funcionario_id)
REFERENCES funcionario (id)
, CONSTRAINT fkeleccion_imei FOREIGN KEY (eleccion_id)
REFERENCES eleccion (id)
, CONSTRAINT fkcasilla_imei FOREIGN KEY (casilla_id)
REFERENCES casilla (id)
);
INSERT INTO casilla (ubicacion )
VALUES ('Rectoria'),('Facultad de medicina'),('Facultad de arquitectura');
```



Paso 4: Crear una cuenta de usuario que se usará para conectarse desde el framework

```
CREATE USER webdev@'%' IDENTIFIED WITH caching_sha2_password BY  
'pwdWebDev' PASSWORD EXPIRE INTERVAL 160 DAY  
FAILED_LOGIN_ATTEMPTS 3 PASSWORD_LOCK_TIME 2;  
GRANT ALL ON bdlara.* TO webdev@'%' ;
```

Paso 5: Abrir otra consola y generar los modelos que representen las tablas y sus relaciones, ejecutando la siguiente instrucción.

```
docker exec -it php8 /bin/sh  
cd /var/www/html/proyecto_laravel  
composer require reliese/laravel
```

Paso 6: Editar el archivo config/app.php y agregar en la sección de providers

```
// ...  
'providers' => [  
/*  
* Package Service Providers...  
*/  
Reliese\Coders\CodersServiceProvider::class,  
],
```


Paso 7: Generar las clases modelo de todas las tablas de la BD, ejecutando siguientes instrucciones:

```
php artisan vendor:publish --tag=reliese-models
```

```
php artisan code:models
```

```
php artisan config:clear
```

El resultado de esta última instrucción son los modelos como se aprecia en la siguiente imagen

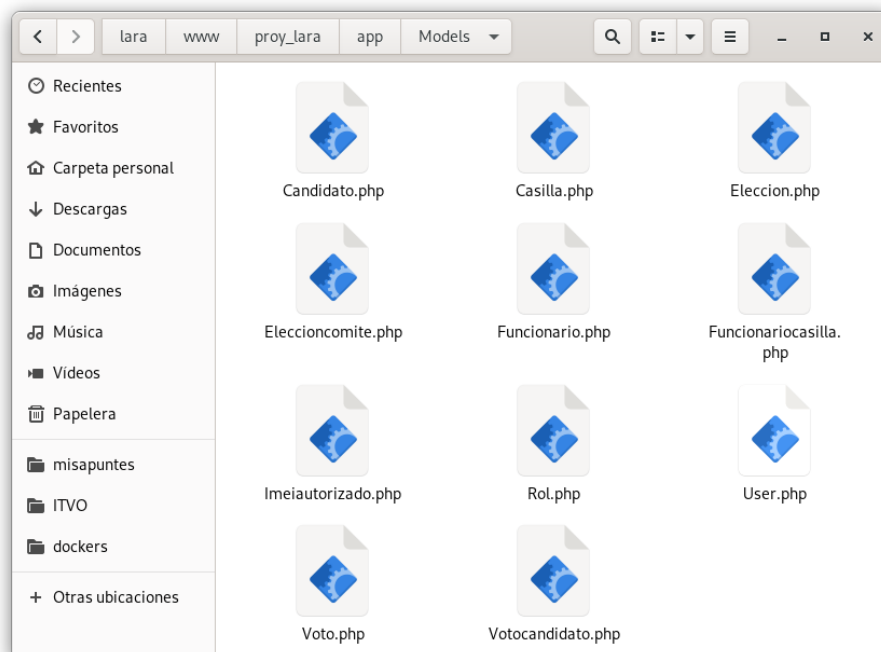


Figura 1. Archivos que representan la clases modelo de las tablas de la BD