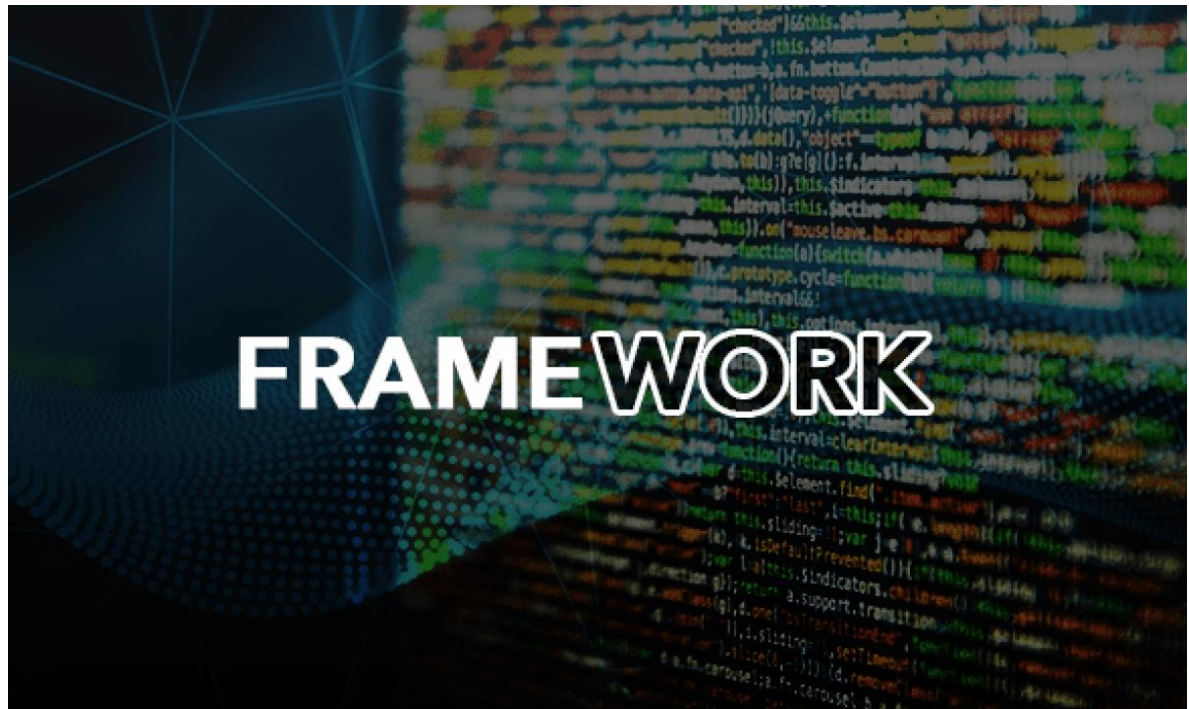




FRAMEWORK PARA DESARROLLO WEB



Estudiantes: Amadeo Santos Vargas

Docente: Ambrosio Cardoso Jiménez

Trabajo: Bitácora

Carrera: Informática

Semestre: Octavo

Grupo: A



18 de febrero de 2022



Índice.

Introducción.....	4
Objetivos	4
Problemas.....	4
Instalación y configuración del FrameWork.....	5
Objetivo.....	5
Requisitos:	5
Paso 1: Carpeta lara	5
Paso 2: Archivo nginx.conf.....	5
Paso 3: Carpeta www	6
Paso 4: Archivo docker-compose.yml	6
Paso 5: Archivo Dockerfile	7
Paso 6: Creación de contenedores.....	8
Paso 7: Iniciar los contenedores.....	8
Paso 8: Enlistar contenedores creados.....	9
Paso 9: Acceso al contenedor php8.....	9
Paso 10: composer.....	9
Paso 11: Directorio html.....	10
Paso 12: Proyecto_laravel	10
Paso 13: Archivo .env	10
Paso 14: Probar resultado	12
Configuración de SGBD	13
Objetivo.....	13
Requisitos.....	13
Paso 1: Contenedor de mysql8.....	13



Paso 2: root.....	13
Paso 3: Crear la BD, las tablas y sus relaciones.	14
SHOW DATABASES;	14
USE bdlara;	14
Crear tablas.	14
Tabla casilla.....	14
Tabla candidato	14
Tabla elección.....	15
Tabla funcionario	15
Tabal rol.....	15
Tabla eleccionconmite	15
Tabla voto	16
Tabal votocandidato.....	16
Tabla funcionariocasilla.	16
Tabla imeiautorizado.....	17
Show tables;	17
Insertar información	17
Paso 4: Usuario	18
Paso 5: php8.....	18
Paso 6: Archivo app.php	18
Paso 7: Generar las clases	19
Resultado.....	20
Conclusión.....	21
Referencia.....	21



Introducción.

La siguiente practica esta enfocado en dos puntos el primero es la instalación y configuración de un framework que como sabes es es un esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos, una especie de plantilla que sirve como punto de partida para la organización y desarrollo de software, y al igual sabes que hay muchos framework, pero en este caso utilizaremos el lavavel, que es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5, PHP 7 y PHP 8, la cual para esta asignatura nos ayudara mucho para comprender y aprender sobre el mundo de framework,.

La segunda parte de es la configuración de SBBD, en la cual esta mas enfocado en la realización de una base, como a la igual la creación de un usuario, y todo esto se desarrollara una vez terminado la configuración del framework ya que a través de mysql y php, se realizara esta práctica.

Objetivos

Los objetivos de esta practica es para conocer la funcionalidad que tienen los framework, y más específicamente como manipular el Laravel, ya con base a este estaremos trabajaremos en el semestre, también uno de objetivos es como profesionales conocer ha cerca el mundo de framework ya que como ingenieros en informática es fundamental conocer esta rama de informática y por eso es fundamental prepararnos como profesionales.

Problemas.

En la siguiente práctica no se presento casi problemas, ya que el manual proporcionado por el docente estaba completo y detallado a la perfecto, solo tuve un inconveniente que fue al comprar en la navegador, ya que al momento de hacerlo no me aparecía y esto lo solucione con docente ya que era por falta de permiso.



Instalación y configuración del FrameWork

Objetivo

Configurar docker para nginx (latest), php (latest), mysql8 y laravel 8 (latest)

Requisitos:

- Docker 20.10.12
- Tener instalado docker-compose
- Tener instalado composer (gestor de paquetes para programar en PHP el cual provee los formatos estándar necesarios para manejar dependencias y librerías).

Paso 1: Carpeta lara

Se realizó la creación de una carpeta llamada lara para el almacenamiento de los contenedores.

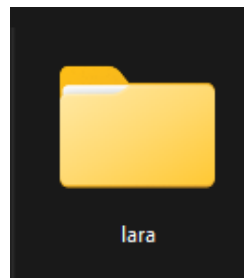


Ilustración 1 Carpeta lara

Paso 2: Archivo nginx.conf

Se realizó la creación de una carpeta llamada nginx y en la cual dentro de ella se creó un archivo llamado nginx.conf la cual se utilizará para descargar las imágenes de nginx.

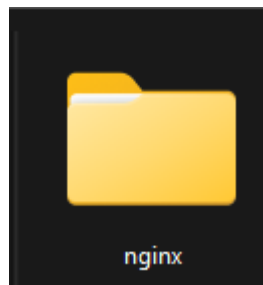


Ilustración 2. Carpeta nginx

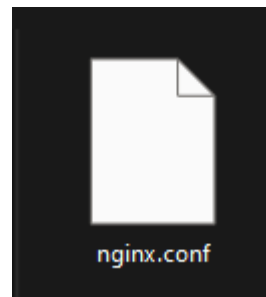


Ilustración 3. Archivo nginx.conf

```
server {
    listen 80;
    root /var/www/html/proyecto_laravel/public;
    #root /var/www/html;
    index index.php index.html;
    server_name localhost;

    error_log /var/log/nginx/error.log;
    access_log /var/log/nginx/access.log;

    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }

    location ~ \.php$ {
        try_files $uri =404;
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_pass php8:9000;
        fastcgi_index index.php;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param PATH_INFO $fastcgi_path_info;
    }
}
```

Ilustración 4. Contenido del archivo nginx

Paso 3: Carpeta www

Se creo una carpeta llamada www, donde se almacenará el proyecto que se creara.



Ilustración 5. Carpeta www

Paso 4: Archivo docker-compose.yml

Crea un archivo para creación de los contenedores.



Ilustración 6. Archivo docker-compose



```
version: '3.9'

networks:
  lara:
services:
  nginx:
    image: nginx:stable-alpine
    container_name: lara_nginx
    ports:
      - "8888:80"
    volumes:
      - "/var/www/html"
      - "/nginx/nginx.conf:/etc/nginx/conf.d/default.conf"
    depends_on:
      - php8
      - mysql8
    networks:
      - lara

  mysql8:
    image: mysql:latest
    container_name: mysql8
    volumes:
      - mysql8_data:/var/lib/mysql
    command: ['--character-set-server=utf8mb4', '--collation-server=utf8mb4_unicode_ci', '--default-authentication-plugin=mysql_native_password']
    tty: true
    ports:
      - "3333:3306"
    environment:
      MYSQL_DATABASE: bdlara
      MYSQL_USER: webdev
      MYSQL_PASSWORD: pwdWebDev
      MYSQL_ROOT_PASSWORD: t0pS3cr3t
      SERVICE_NAME: mysql8
    networks:
      - lara
```

Ilustración 7. Contenido de Archivo docker-compose.yml

Paso 5: Archivo Dockerfile

Dentro de carpeta llamada lara, se guarda el archivo llamado Dockerfile.

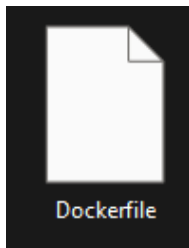


Ilustración 8 Archivo Dockerfile

```
FROM php:8.0.5-fpm-alpine
RUN apk add --no-cache bash
RUN docker-php-ext-install pdo pdo_mysql mysqli

#--- Usar apk en lugar de apt-get para cuando se usa alpine
#--- nota de Cardoso 26-Ene-2022
RUN apk update && apk add \
    curl \
    wget \
    zip \
    unzip \
    composer \
    npm
```

Ilustración 9. Contenido del archivo Dockerfile



Paso 6: Creación de contenedores.

Para la creación de los contenedores se ejecute la siguiente instrucción dentro de la carpeta lara.

#docker-compose build

```
PS C:\Users\52951\Desktop\lara> docker-compose build
mysql8 uses an image, skipping
nginx uses an image, skipping
Building php8
[+] Building 727.8s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 32B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/php:8.0.5-fpm-alpine
=> [1/4] FROM docker.io/library/php:8.0.5-fpm-alpine@sha256:52f7e2b5bd14fa3d6572c6a2439c8bb640f
=> => resolve docker.io/library/php:8.0.5-fpm-alpine@sha256:52f7e2b5bd14fa3d6572c6a2439c8bb640f
=> => sha256:52f7e2b5bd14fa3d6572c6a2439c8bb640f0376872e7ae987c3907d0dfd6f038 1.65kB / 1.65kB
=> => sha256:dea422cf1adb9c1968b14cd94d9631d1fe01269ca5a278e8482ec9c6c478d88e 2.41kB / 2.41kB
=> => sha256:bf970e21a22f1a2ac68f77c7144f3f72b09691330906105cf5e4174eb34b4cab 10.86kB / 10.86kB
=> => sha256:540db60ca9383eac9e418f78490994d0af424aab7bf6d0e47ac8ed4e2e9bcbba 2.81MB / 2.81MB
=> => sha256:933cf2f4a68f7b603d67468c6e390ce893a1410ea927dc00e8faabfd01032afa 1.70MB / 1.70MB
=> => sha256:93c5cc202a60c205410f5462131556b8ecfba3092bceab1bf75723d1a356c7fb 1.26kB / 1.26kB
=> => sha256:74403c16157d84037726eebe566275f9e5fdb3f301ce6c101eeb3fb37b8914ef 269B / 269B
=> => sha256:878a150115619341fad3f5d7ab3363c111289b1de879fafa40151881930b6e40 10.69MB / 10.69MB
=> => extracting sha256:540db60ca9383eac9e418f78490994d0af424aab7bf6d0e47ac8ed4e2e9bcbba
=> => sha256:de4d5d10496329b3362538702bedd5706dee0de45ad5162a9cdeb712e9a9bd25 495B / 495B
=> => sha256:e0eb8654ca396b4bd91d8e9be19358d25ad593dd15f98007dc49e9d5f3f2f060 15.02MB / 15.02MB
=> => sha256:8638523c0348721dd87ef9feaf895254303fe3c944b28a5c9fcd66f7810ebe98 2.27kB / 2.27kB
=> => sha256:9180defaaac9672dcd2ecd1cd5fc2741163619d5b3af0270b645f347e22627f5 17.54kB / 17.54kB
```

Ilustración 10. Creación de los contenedores

Paso 7: Iniciar los contenedores.

Con la siguiente ejecución los contenedores se ejecuten en segundo plano para poder trabajar.

#docker-compose up -d

```
PS C:\Users\52951\Desktop\lara> docker-compose up -d
Creating network "lara_lara" with the default driver
Creating volume "lara_mysql8_data" with default driver
Pulling mysql8 (mysql:latest)...
latest: Pulling from library/mysql
6552179c3509: Pull complete
d69aa66e4482: Pull complete
3b19465b002b: Pull complete
7b0d0cfe99a1: Pull complete
9ccd5a5c8987: Pull complete
2dab00d7d232: Pull complete
64d3afdccc4a: Pull complete
82148d50b16c: Pull complete
8bb7d73a7d0c: Pull complete
74778cd68a75: Pull complete
d7e5f9309140: Pull complete
f2e376ecd59f: Pull complete
```

Ilustración 11. Contenedores ejecutando



Paso 8: Enlistar contenedores creados.

Comprobar que los tres contenedores (nginx, mysql8 y php) estén corriendo a través de la siguiente instrucción.

#docker container ls

```
PS C:\Users\52951\Desktop\lara> docker container ls
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
286b2af782fe	nginx:stable-alpine	"/docker-entrypoint..."	23 hours ago	Up 39 seconds	0.0.0.0:8888->80/tcp	lara_nginx
4ab5a5937a23	lara_php8	"docker-php-entrypoi..."	23 hours ago	Up 23 hours	0.0.0.0:9999->9000/tcp	php8
afeb5ee1fca2	mysql:latest	"docker-entrypoint.s..."	23 hours ago	Up 23 hours	33060/tcp, 0.0.0.0:3333->3306/tcp	mysql8

Ilustración 12. Contenedores corriendo.

Paso 9: Acceso al contenedor php8

Entrar al contenedor a través de la siguiente instrucción para poder empezar a trabajar.

#docker exec -it php8 /bin/sh

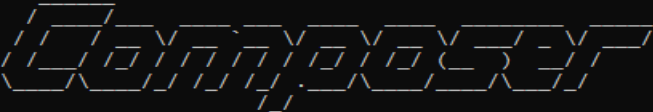
```
PS C:\Users\52951\Desktop\lara> docker exec -it php8 /bin/sh
^H/var/www/html # composer
```

Ilustración 13 Acceso al contenedor php8

Paso 10: composer.

Escribir composer para que nos muestra la documentación de composer.

```
^H/var/www/html # composer
```



```
Composer version 2.0.13 2021-04-27 13:11:08

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display this help message
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
  --ansi                   Force ANSI output
  --no-ansi                Disable ANSI output
  -n, --no-interaction      Do not ask any interactive question
  --profile                Display timing and memory usage information
  --no-plugins              Whether to disable plugins.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache                Prevent use of the cache
  -v|vv|vvv, --verbose     Increase the verbosity of messages: 1 for normal output, 2

Available commands:
  about                Shows the short information about Composer.
  archive              Creates an archive of this composer package.
  browse               Opens the package's repository URL or homepage in your browser.
  cc                  Clears composer's internal package cache.
  check-platform-reqs Check that platform requirements are satisfied.
  clear-cache          Clears composer's internal package cache.
  clearcache           Clears composer's internal package cache.
  config               Sets config options.
  create-project        Creates new project from a package into given directory.
```

Ilustración 14 Documentación de composer



Paso 11: Directorio html

Estando en el contenedor cambiar al directorio con

```
#cd /var/www/html
```

```
/var/www/html # cd /var/www/html
```

Ilustración 15 Cambio de directorio dentro de contenedor.

Paso 12: Proyecto_laravel

Crear el proyecto **proyecto_laravel** estando en la ruta **/var/www/html** ejecutamos la siguiente instrucción.

```
#composer create-project --prefer-dist laravel/laravel proyecto_laravel
```

```
/var/www/html # composer create-project --prefer-dist laravel/laravel proyecto_laravel
Creating a "laravel/laravel" project at "./proyecto_laravel"
Installing laravel/laravel (v9.0.0)
- Downloading laravel/laravel (v9.0.0)
- Downloading laravel/laravel (v9.0.0)
- Downloading laravel/laravel (v9.0.0)
- Downloading laravel/laravel (v9.0.0)
- Installing laravel/laravel (v9.0.0): Extracting archive
Created project in /var/www/html/proyecto_laravel
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 109 installs, 0 updates, 0 removals
- Locking asm89/stack-cors (v2.1.1)
- Locking brick/math (0.9.3)
- Locking dflydev/dot-access-data (v3.0.1)
- Locking doctrine/inflector (2.0.4)
- Locking doctrine/instantiator (1.4.0)
- Locking doctrine/lexer (1.2.2)
- Locking dragonmantank/cron-expression (v3.3.1)
- Locking egulias/email-validator (3.1.2)
- Locking facade/ignition-contracts (1.0.2)
- Locking fakerphp/faker (v1.19.0)
```

Ilustración 16 Creacion del proyecto_laravel

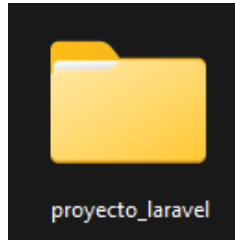
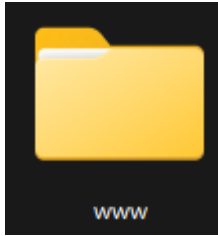
Comprobar resultado.

```
/var/www/html # ls
proyecto_laravel
/var/www/html #
```

Paso 13: Archivo .env

Desl proyecto creado editar el archivo llamado .env

/var/www/html/proyecto_laravel/.env.



Contenido del archivo modificado.

```
APP_NAME=lara
APP_ENV=local
APP_KEY=base64:6q9Is8pxmQ5nMDaRrNVzkFJiIWyi5BAuw4qGwGV+qes=
APP_DEBUG=true
APP_URL=http://localhost:8000

LOG_CHANNEL=stack
LOG_DEPRECATED_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=172.18.0.3
DB_PORT=3306
DB_DATABASE=bdlara
DB_USERNAME=webdev
DB_PASSWORD=pwdWebDev
```

Ilustración 17 Archivo. env modificado

La IP se obtiene con la siguiente instrucción

```
docker inspect -f '{{range .NetworkSettings.Networks}} {{.IPAddress}}{{end}}' mysql8
```



```
PS C:\WINDOWS\system32> docker inspect -f '{{range .NetworkSettings.Networks}} {{.IPAddress}}{{end}}' mysql8
172.18.0.3
PS C:\WINDOWS\system32>
```

Ilustración 18 Obtencion de la ip

Asignar permisos.

```
/var/www/html # cd proyecto_laravel/
/var/www/html/proyecto_laravel # chmod -R 777 storage
/var/www/html/proyecto_laravel #
```

Ilustración 19 Asignación de permisos.

Paso 14: Probar resultado

<http://localhost:8888>

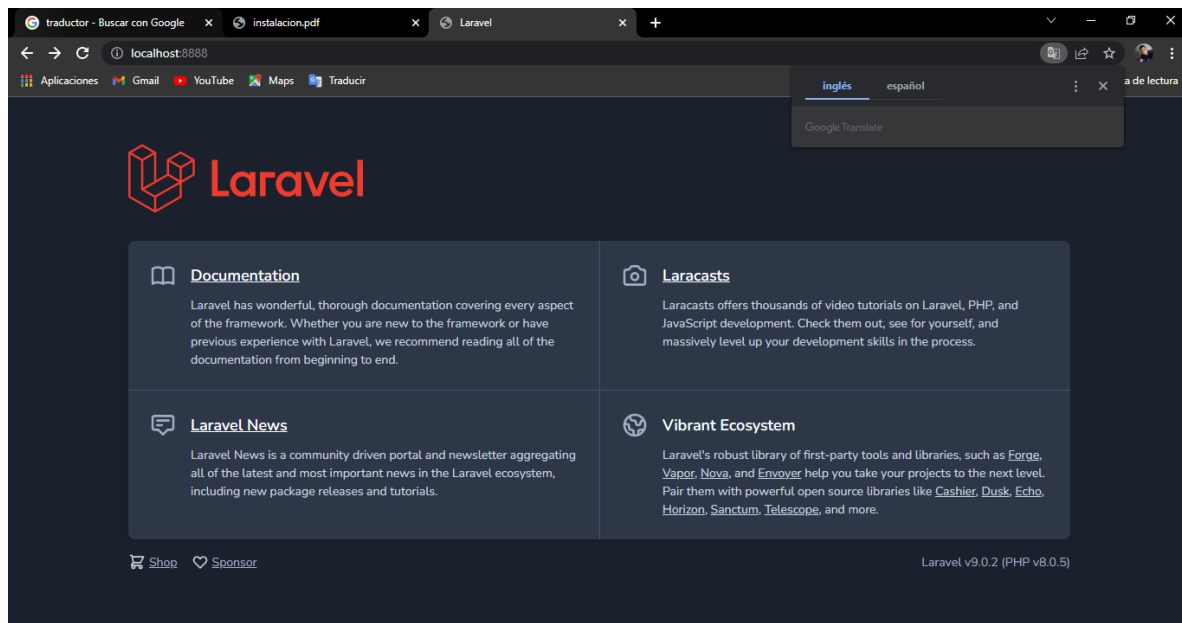


Ilustración 20 Laravel



Configuración de SGBD

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

Paso 1: Contenedor de mysql8

Primero que nada, accederemos al contenedor mysql8 con la siguiente instrucción.

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

```
PS C:\Users\52951\Desktop> docker exec -it mysql8 sh
a#
```

Ilustración 21 acceder al contenedor mysql8

Paso 2: root

Entrar a mysql con la cuenta con la siguiente instrucción.

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

```
# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.28 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```

Ilustración 22 acceder a root



Paso 3: Crear la BD, las tablas y sus relaciones.

Creación de tablas se realizó de siguiente manera con las siguientes instrucciones:

SHOW DATABASES;

```
mysql> show databases;
+-----+
| Database |
+-----+
| bdlara   |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
+-----+
5 rows in set (5.61 sec)

mysql>
```

USE bdlara;

```
mysql> use bdlara;
Database changed
```

Crear tablas.

Tabla casilla

```
mysql> CREATE TABLE casilla
-> ( id BIGINT AUTO_INCREMENT
-> , ubicacion VARCHAR (100) NOT NULL
-> , CONSTRAINT pkcasilla PRIMARY KEY(id)
-> );
Query OK, 0 rows affected (6.54 sec)
```

Tabla candidato

```
mysql> 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)
-> );
Query OK, 0 rows affected (2.97 sec)
```



Tabla elección

```
mysql> 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)
-> );
Query OK, 0 rows affected (1.96 sec)
```

Tabla funcionario

```
mysql> CREATE TABLE funcionario
-> ( id BIGINT AUTO_INCREMENT
-> , nombrecompleto VARCHAR (200) NOT NULL
-> , sexo CHAR
-> , CONSTRAINT pkfuncionario PRIMARY KEY (id)
-> );
Query OK, 0 rows affected (2.88 sec)
```

Tabla rol

```
mysql> CREATE TABLE rol
-> ( id BIGINT AUTO_INCREMENT
-> , descripcion VARCHAR (100) NOT NULL
-> , CONSTRAINT pkrol PRIMARY KEY(id)
-> );
Query OK, 0 rows affected (3.02 sec)
```

Tabla eleccioncomite

```
mysql> 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)
-> );
Query OK, 0 rows affected (5.19 sec)
```




Tabla voto

```
mysql> 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)
-> );
Query OK, 0 rows affected (3.24 sec)
```

Tabal votocandidato.

```
mysql> 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)
-> );
Query OK, 0 rows affected (7.42 sec)
```

Tabla funcionariocasilla.

```
mysql> 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)
-> );
Query OK, 0 rows affected (6.15 sec)
```



Tabla imeiautorizado

```
mysql> 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)
-> );
Query OK, 0 rows affected (3.68 sec)
```

Consultar tablas creadas.

Show tables;

```
mysql> show tables;
+-----+
| Tables_in_bdlara |
+-----+
| candidato         |
| casilla           |
| eleccion          |
| eleccioncomite    |
| funcionario        |
| funcionariocasilla |
| imeiautorizado    |
| rol               |
| voto              |
| votocandidato     |
+-----+
10 rows in set (0.25 sec)
```

Insertar información

```
mysql> INSERT INTO casilla (ubicacion )
-> VALUES ('Rectoria'),('Facultad de medicina'),('Facultad de arquitectura');
Query OK, 3 rows affected (2.53 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql> select * from casilla;
+-----+-----+
| id | ubicacion |
+-----+-----+
| 1 | Rectoria |
| 2 | Facultad de medicina |
| 3 | Facultad de arquitectura |
+-----+-----+
3 rows in set (0.34 sec)
```



Paso 4: Usuario

Crear una cuenta de usuario que se usará para conectarse desde el framework con la siguiente instrucción.

```
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@'%' ;
```

```
mysql> 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@'%' ;
```

Ilustración 23 Creación de usuario

Paso 5: php8

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

```
docker exec -it php8 /bin/sh
```

```
PS C:\WINDOWS\system32> docker exec -it php8 /bin/sh
```

Ilustración 24 Acceder a php8

```
cd /var/www/html/proyecto_laravel
```

```
/var/www/html # cd /var/www/html/proyecto_laravel  
/var/www/html/proyecto_laravel #
```

Ilustración 25 Acceder al proyecto_laravel

```
composer require reliese/laravel
```

```
/var/www/html/proyecto_laravel # composer require reliese/laravel  
Using version ^1.1 for reliese/laravel  
./composer.json has been updated  
Running composer update reliese/laravel  
Loading composer repositories with package information  
Updating dependencies  
Nothing to modify in lock file  
Installing dependencies from lock file (including require-dev)  
Nothing to install, update or remove  
Generating optimized autoload files
```

Paso 6: Archivo app.php

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



```
/*
 * Package Service Providers...
 */

/*
 * Application Service Providers...
 */
App\Providers\AppServiceProvider::class,
App\Providers\AuthServiceProvider::class,
// App\Providers\BroadcastServiceProvider::class,
Reliese\Coders\CodersServiceProvider::class,
App\Providers\EventServiceProvider::class,
App\Providers\RouteServiceProvider::class,

/*
 * Package Service Providers...
 */
```

Ilustración 26 Archivo `app.php`

Paso 7: Generar las clases

Para generar las clases modelo de todas las tablas de la BD, ejecutando siguientes instrucciones:

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

```
/var/www/html/proyecto_laravel # php artisan vendor:publish --tag=reliese-models
Publishing complete.
```

```
php artisan code:models
```

```
/var/www/html/proyecto_laravel # php artisan code:models
^[[ACheck out your models for bdlara
```

```
php artisan config:clear
```

```
/var/www/html/proyecto_laravel # php artisan config:clear
Configuration cache cleared!
```



Resultado.

El resultado de esta última instrucción son los modelos dentro de la carpeta models.

```
/var/www/html/proyecto_laravel # cd app
/var/www/html/proyecto_laravel/app # cd Models
/var/www/html/proyecto_laravel/app/Models # dir
/bin/sh: dir: not found
/var/www/html/proyecto_laravel/app/Models # ls
Candidato.php      Funcionario.php      User.php
Casilla.php        Funcionariocasilla.php Voto.php
Eleccion.php       Imeiautorizado.php   Votocandidato.php
Eleccioncomite.php Rol.php
/var/www/html/proyecto_laravel/app/Models #
```

Ilustración 27 Archivos dentro de la carpeta Modelos por medio de consola

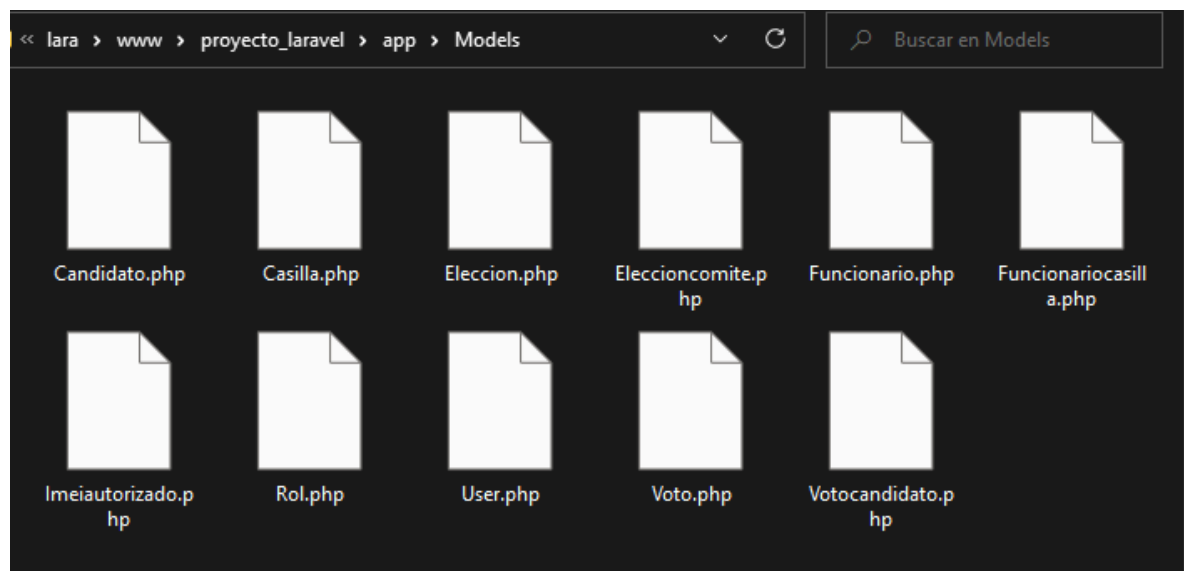


Ilustración 28 Archivos dentro de la carpeta Models de forma Grafica.



Conclusión.

La práctica realizada como se observó fue la instalación y configuración de laravel, y después sobre la base de datos, solamente puedo concluir que fue una práctica de mucha ayuda porque hizo recordar varios conceptos, como también el funcionamiento del Docker, y conocer cosas nuevas como el funcionamiento de los framework aunque solo vimos uno, se que con la esto será más fácil utilizar otro y pues con esto nos vamos preparando profesionalmente, con los objetos que se planteó para la realización de esta práctica y los objetivos que el profe se planteó para los estudiante si se cumplió.

Referencia.

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

- Instalación y configuración de Laravel 8 en Docker en:
https://plataforma.voaxaca.tecnm.mx/pluginfile.php/51481/mod_resource/content/1/instalacion.pdf
- Conexión del framework a una base de datos en
https://plataforma.voaxaca.tecnm.mx/pluginfile.php/51483/mod_resource/content/1/sqbd.pdf