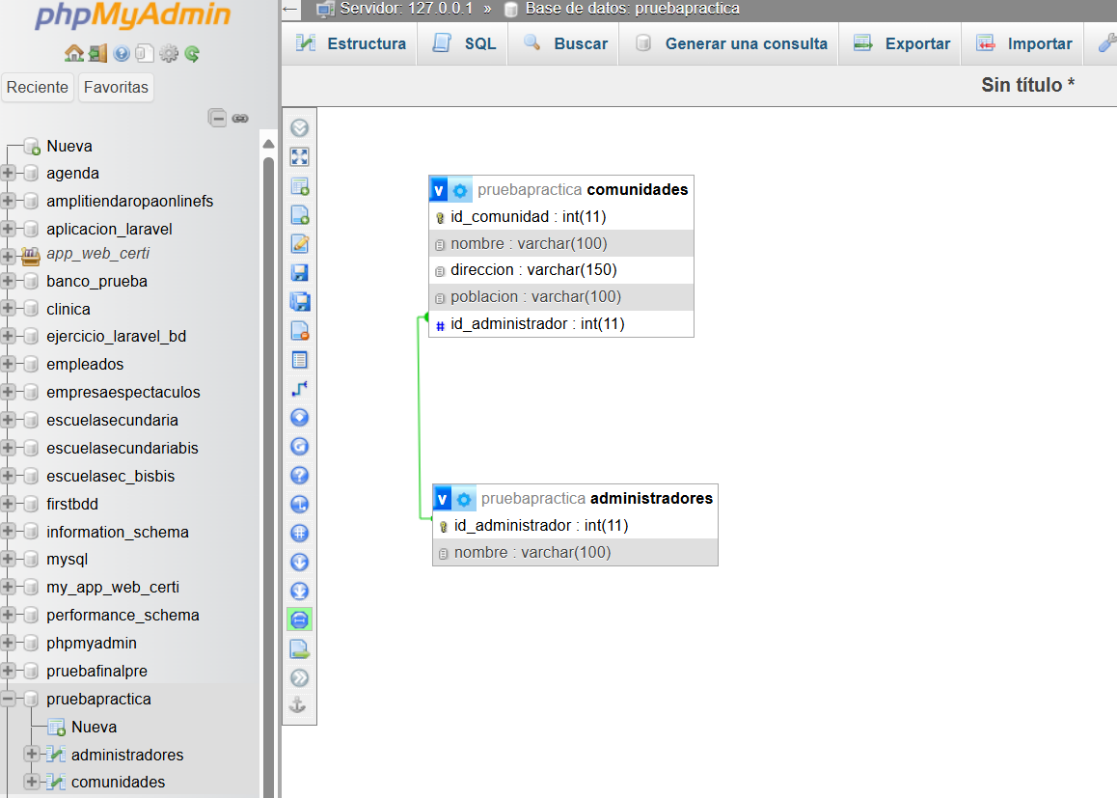
**Autor: Ernesto Calvo Moreno**

**Prog. Lenguajes Estructurados**

**Fecha: 30/07/24**

1. **Crea la BBDD PruebaPractica con las tablas y campos especificados, utilizando los tipos y tamaños más adecuados y sus correspondientes relaciones**

****

1. **Realiza el pseudocódigo para dar de alta un vecino**

INICIO

1. Conectar a la Base de Datos

a. conexion = ConectarABaseDeDatos('localhost', 'usuario', 'contraseña', 'PruebaPractica')

b. SI conexion ES NULO

i. Mostrar 'Error al conectar a la base de datos'

ii. TERMINAR

c. FIN SI

2. Solicitar Datos del Nuevo Vecino

a. nombre = LeerEntrada('Ingrese el nombre del vecino:')

b. escalera = LeerEntrada('Ingrese la escalera:')

c. piso = LeerEntrada('Ingrese el piso:')

d. puerta = LeerEntrada('Ingrese la puerta:')

e. id\_comunidad = LeerEntrada('Ingrese el ID de la comunidad:')

3. Validar y Sanitizar los Datos Ingresados

a. SI nombre ESTÁ VACÍO O escalera ESTÁ VACÍO O piso ESTÁ VACÍO O puerta ESTÁ VACÍO O id\_comunidad ESTÁ VACÍO

i. Mostrar 'Datos inválidos. Por favor, intente de nuevo.'

ii. TERMINAR

b. FIN SI

c. nombre = htmlspecialchars(nombre, ENT\_QUOTES, 'UTF-8')

d. escalera = htmlspecialchars(escalera, ENT\_QUOTES, 'UTF-8')

e. piso = htmlspecialchars(piso, ENT\_QUOTES, 'UTF-8')

f. puerta = htmlspecialchars(puerta, ENT\_QUOTES, 'UTF-8')

g. id\_comunidad = htmlspecialchars(id\_comunidad, ENT\_QUOTES, 'UTF-8')

4. Validar que el ID de la Comunidad Exista en la Tabla `Comunidades`

a. resultadoComunidad = EjecutarConsulta(conexion, 'SELECT COUNT(\*) FROM Comunidades WHERE id\_comunidad = ?', [id\_comunidad])

b. SI resultadoComunidad[0] = 0

i. Mostrar 'ID de la comunidad no existe. Por favor, verifique y vuelva a intentar.'

ii. TERMINAR

c. FIN SI

5. Insertar los Datos del Nuevo Vecino en la Tabla `Vecinos`

a. sentenciaSQL = 'INSERT INTO Vecinos (Nombre, Escalera, Piso, Puerta, id\_comunidad) VALUES (?, ?, ?, ?, ?)'

b. parametros = (nombre, escalera, piso, puerta, id\_comunidad)

c. resultado = EjecutarSentencia(conexion, sentenciaSQL, parametros)

i. SI resultado ES FALLO

Mostrar 'Error al insertar datos del vecino'

ii. SINO

Mostrar 'Vecino registrado exitosamente'

d. FIN SI

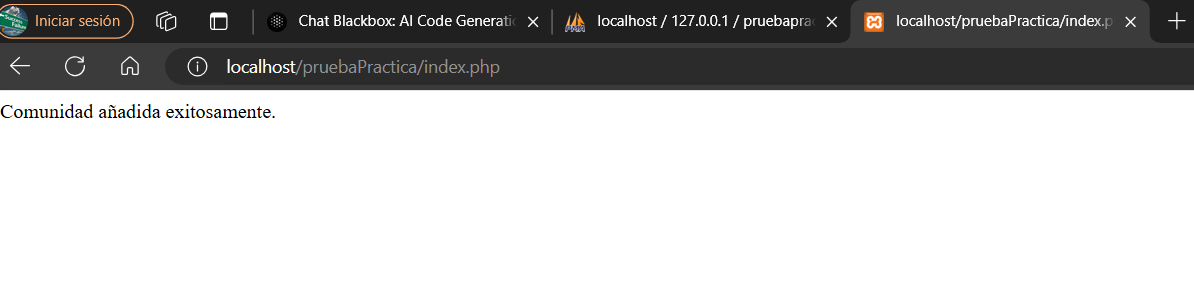
6. Cerrar la Conexión a la Base de Datos

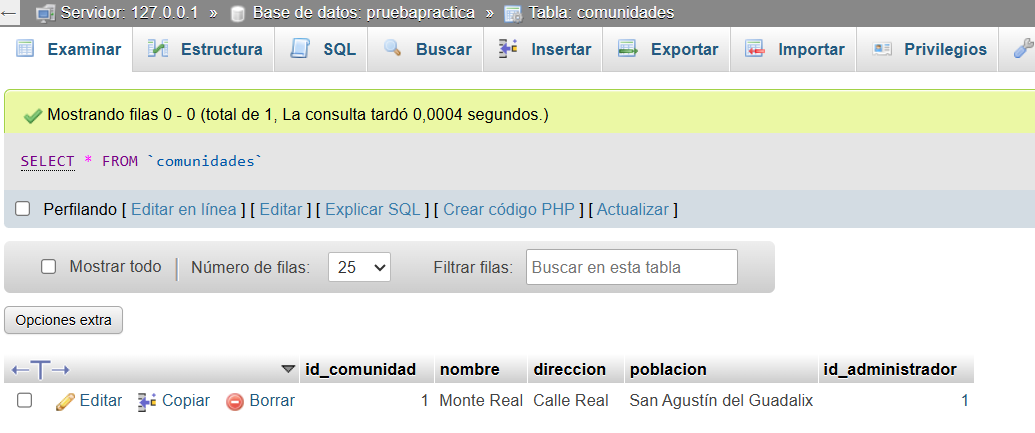
CerrarConexion(conexion)

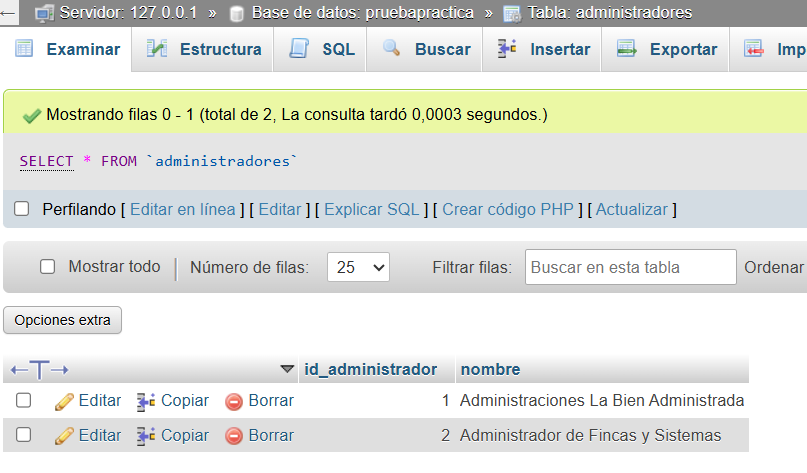
7. Fin del Proceso

TERMINAR

1. **Alta de nueva Comunidad**:

****

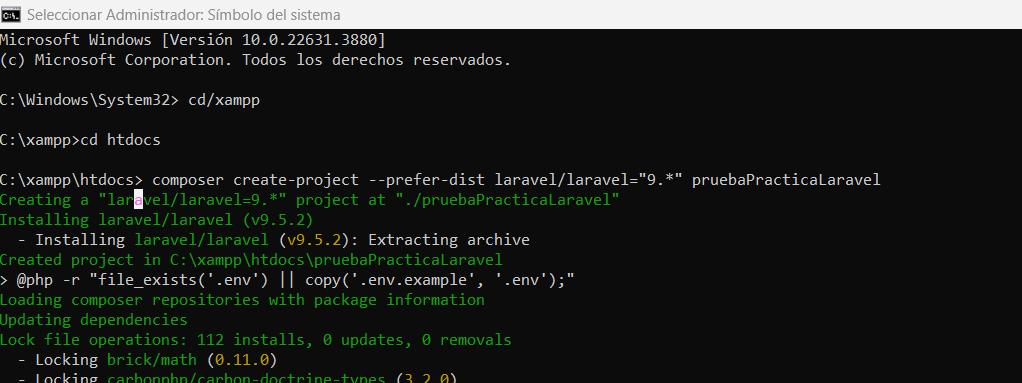


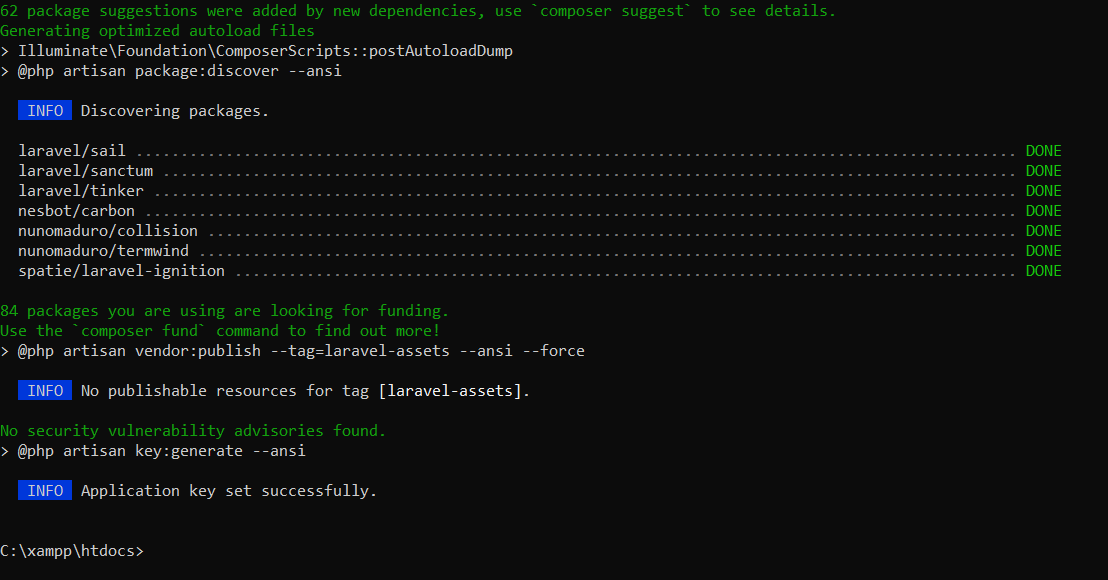


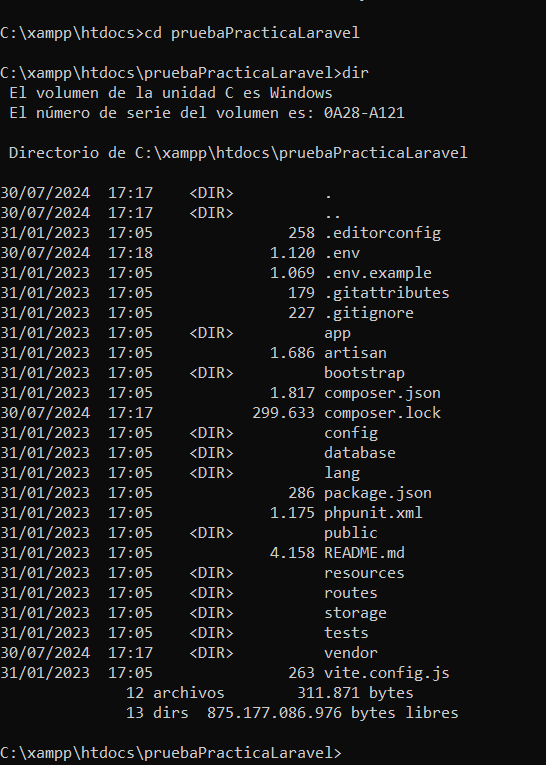
1. **Enlace GitHub**

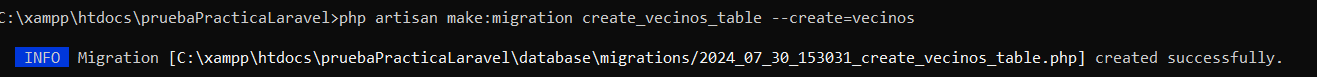
https://github.com/ECM1966/pruebaPractica

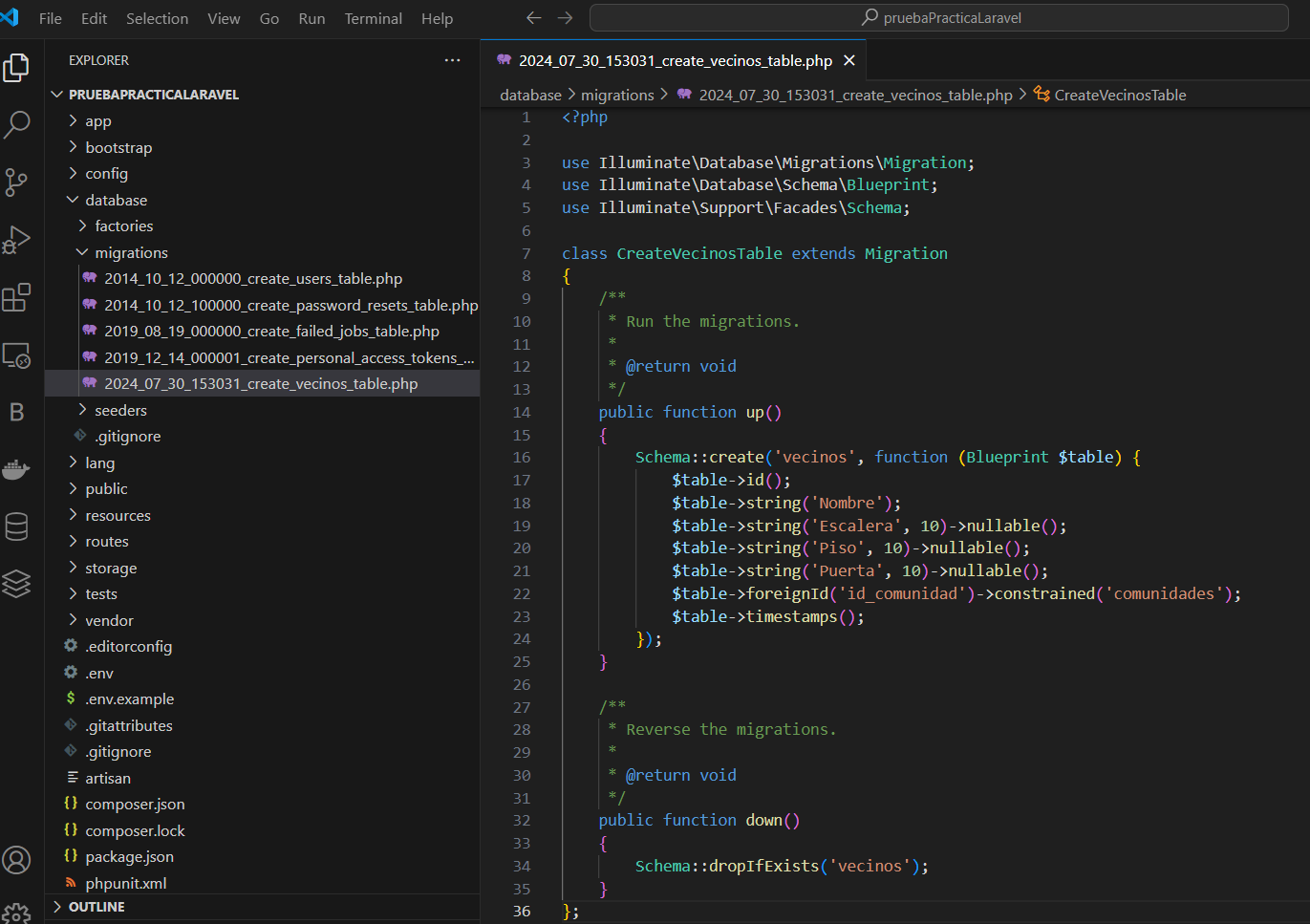
1. **Crea proyecto en Laravel “pruebaPractica”. Crea las migraciones y vistas para la tabla Vecinos de la BD**



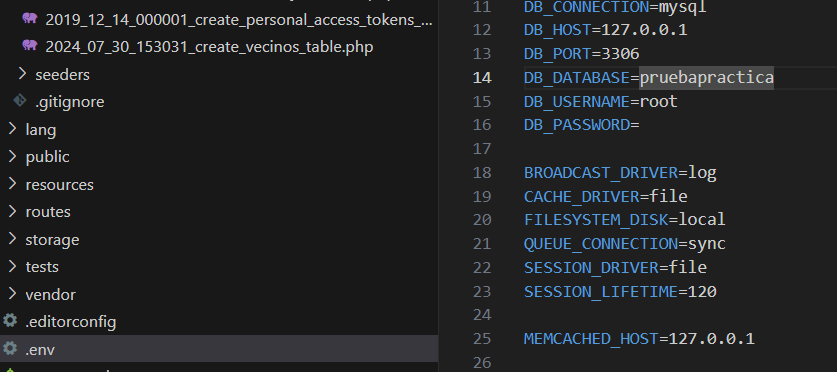




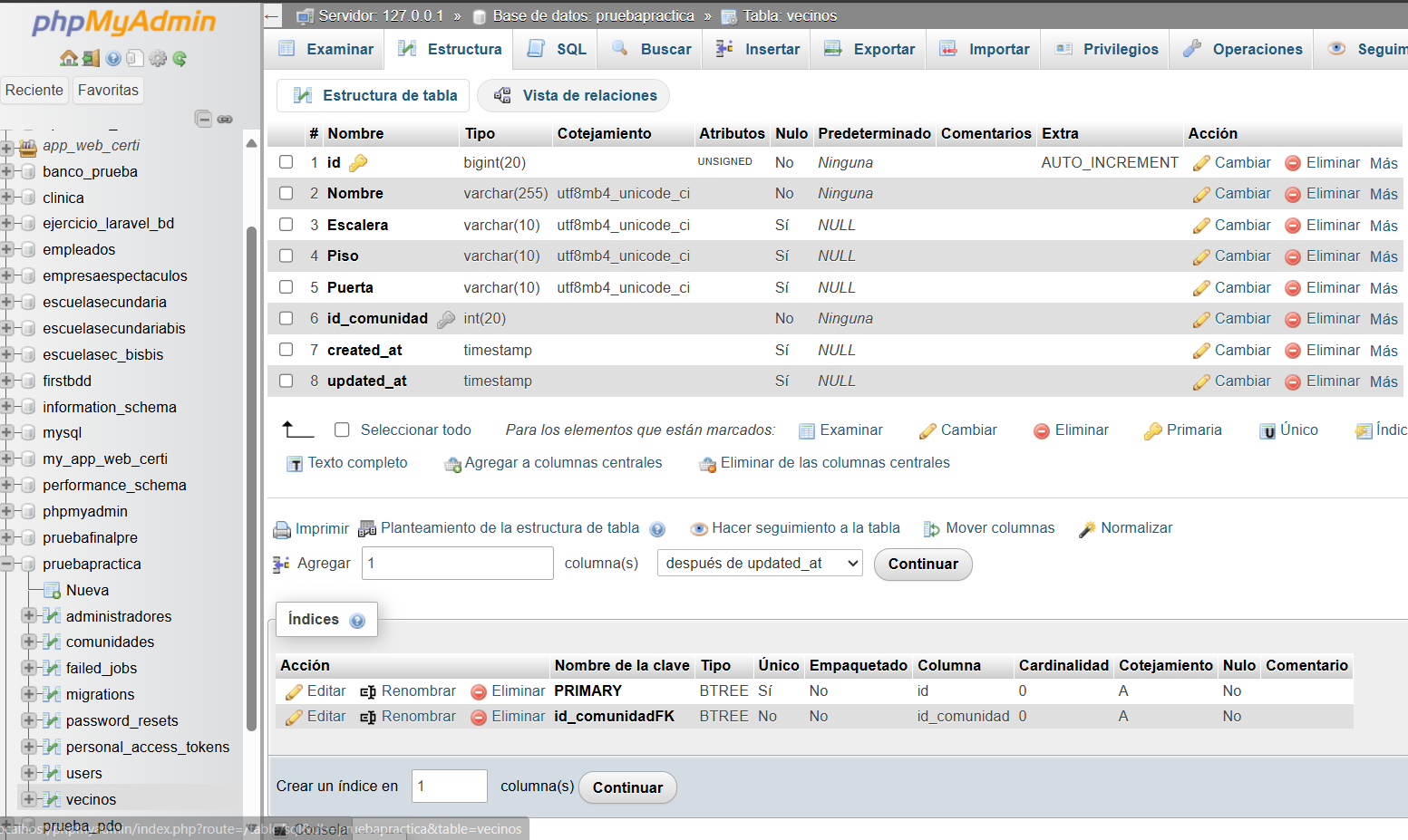
**5.1 migración**

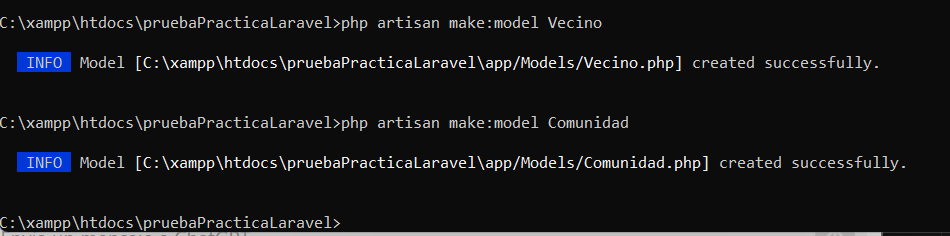
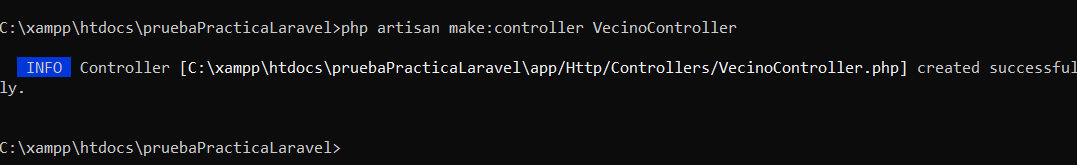
* + 1. **Definir la estructura de la tabla**

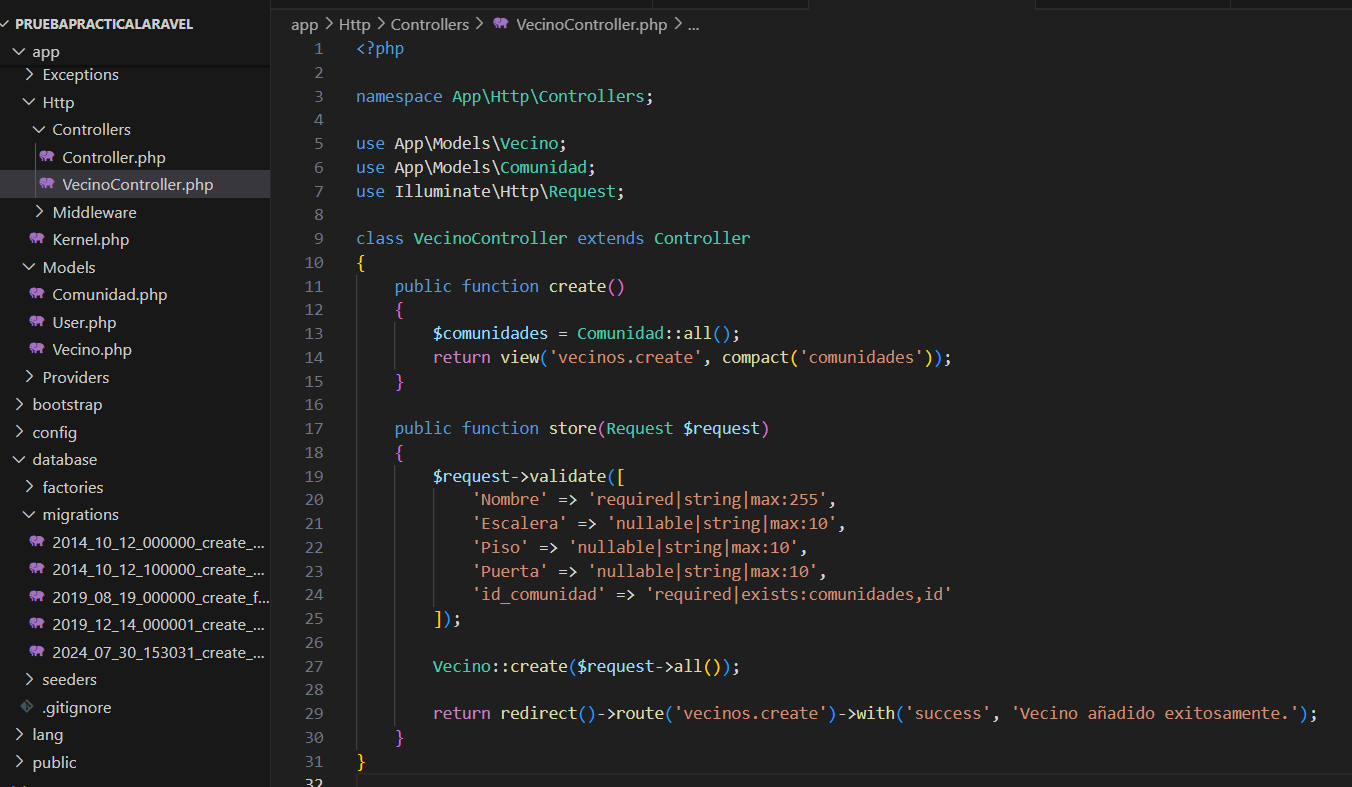
**5.1.2 configurar .env**



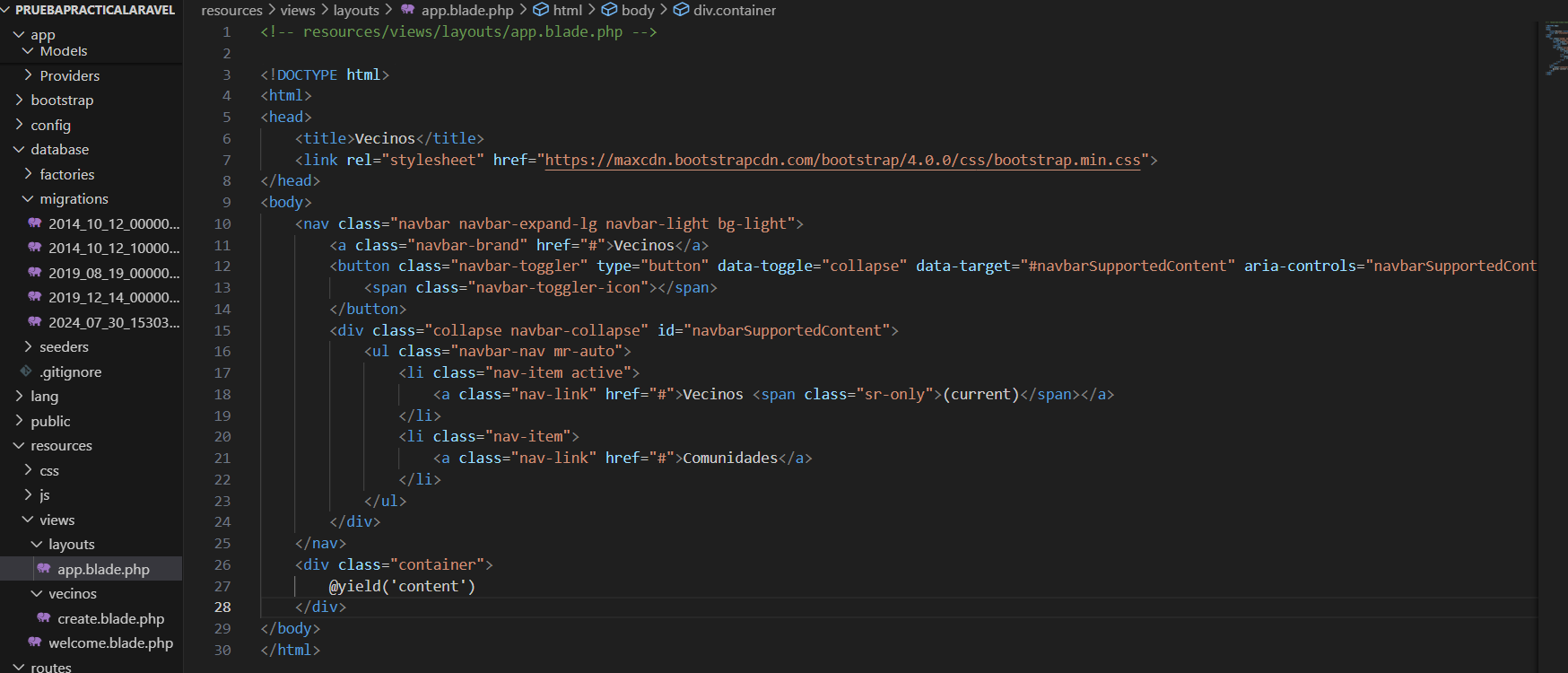
* + 1. **Tabla creada**

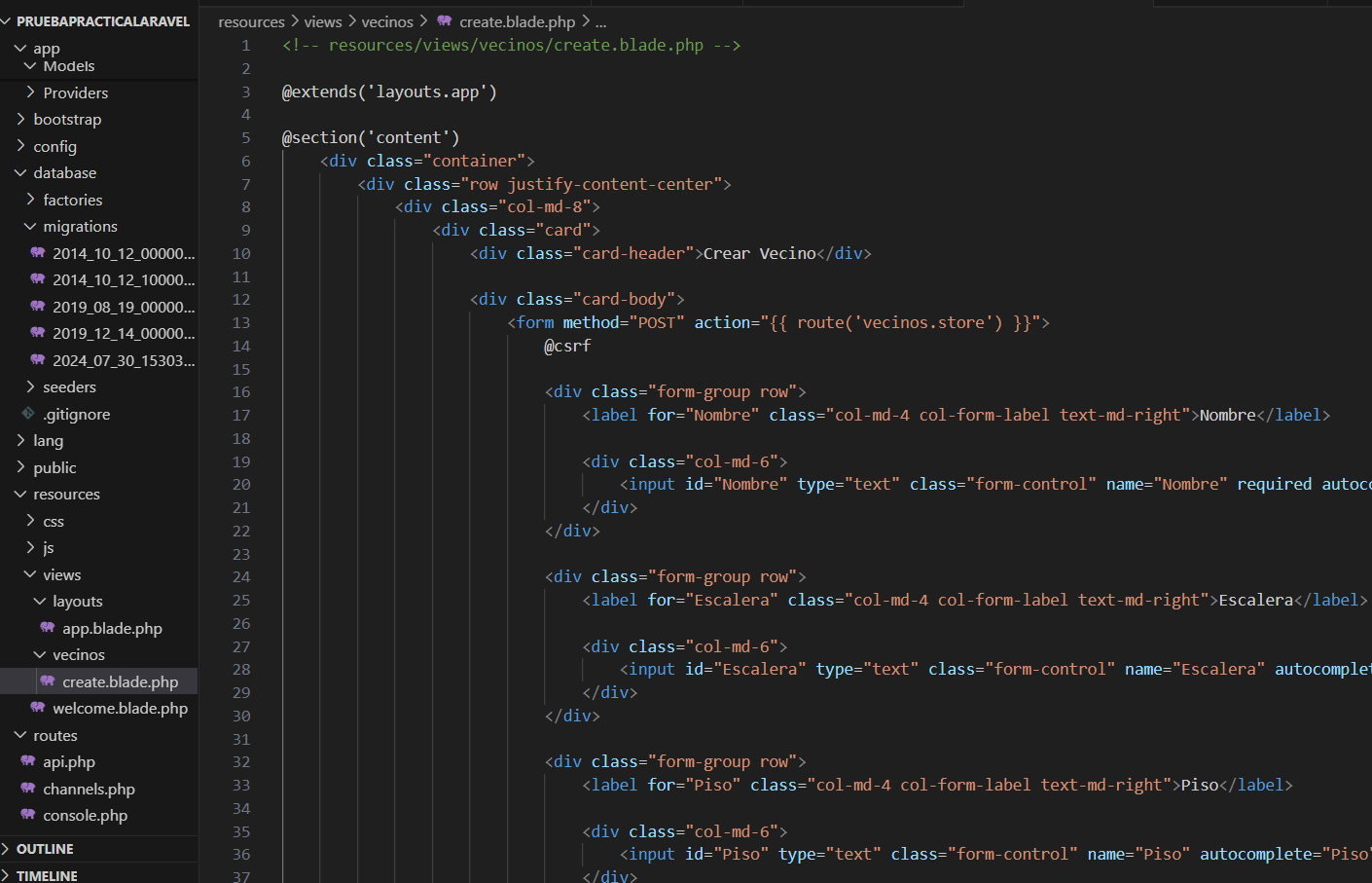


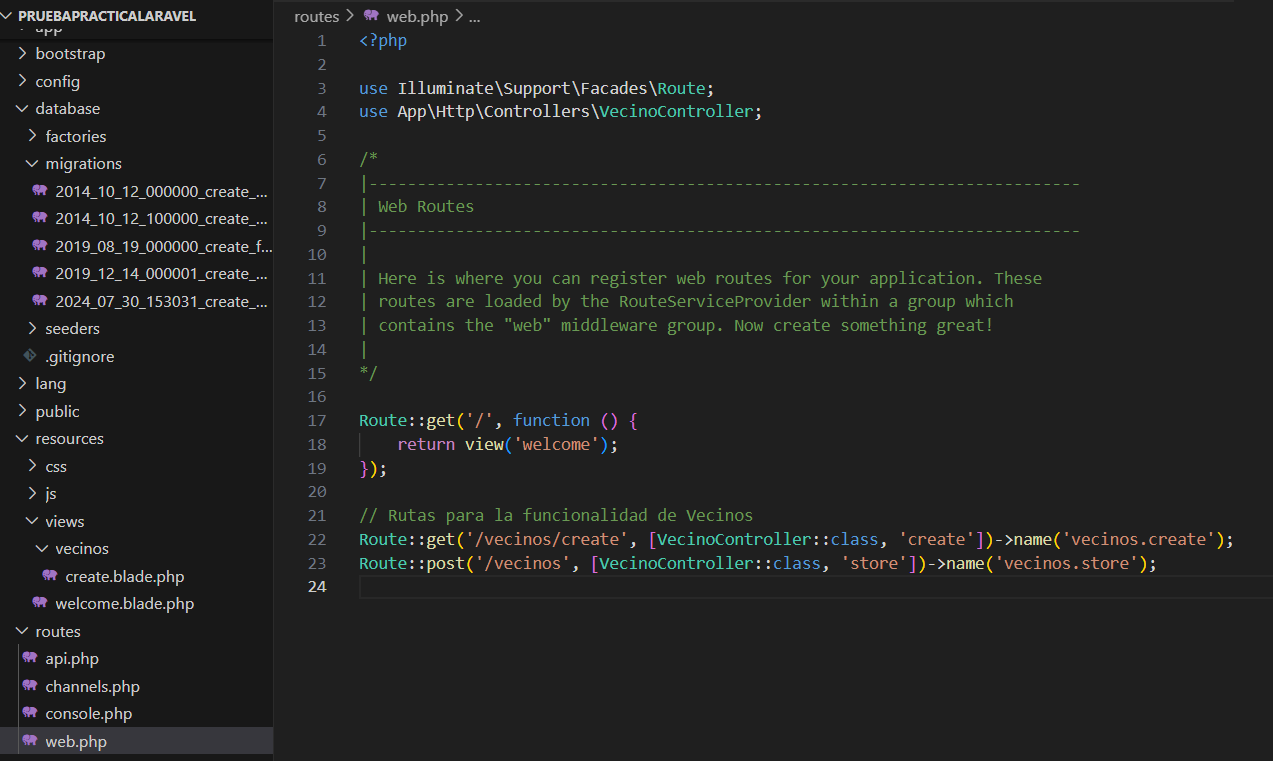
* 1. **crear modelos Comunidad, Vecino**
  2. **Controlador VecinoController**



1. **Crear la vista para el form Crear Vecino**

**resources/views/layout/ app.blade.php**

Resources/views/vecinos/create.blade.php

1. **Añadir las ruras para crear y almacenar vecinos (routes/web.php)**

**Vista creada (pena de vista)**

