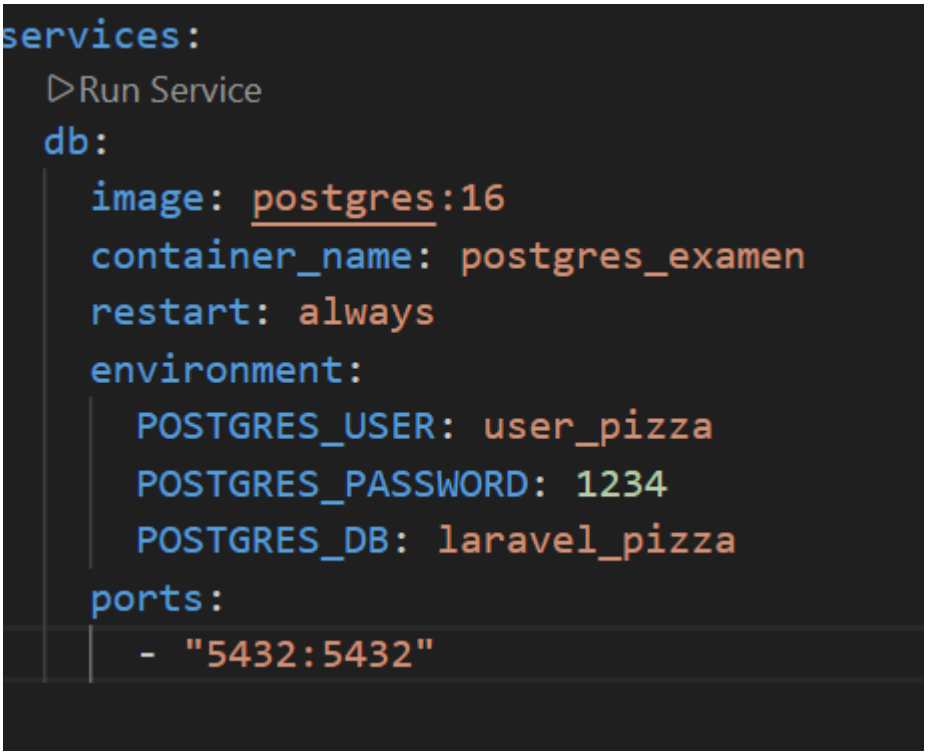
Documentacion

Miguel Angel Santiago Barriga

2ºDaw Examen Final Despliegue de aplicaciones web

Local

Para el local solo tenemos que crear un contenedor docker con la base de datos y rellenar las credenciales en el .env



docker-compose de la base de datos local

```
{
  "status": true,
  "message": "Operación finalizada",
  "data": [
    {
      "id": 1,
      "name": "margarita",
      "price": 10.5,
      "created_at": "2026-02-05T18:37:24.000000Z",
      "updated_at": "2026-02-05T18:37:24.000000Z"
    },
    {
      "id": 3,
      "name": "Pepperoni",
      "price": 10.5,
      "created_at": "2026-02-05T18:37:24.000000Z",
      "updated_at": "2026-02-05T18:37:24.000000Z"
    },
    {
      "id": 2,
      "name": "Funghi",
      "price": 11,
      "created_at": "2026-02-05T18:37:24.000000Z",
      "updated_at": "2026-02-05T18:37:24.000000Z"
    }
  ]
}
```


Salida del primer endpoint

Dev

Para el dev necesitamos un contenedor para la base de datos y otro para la build de la aplicación. También configuraremos un Dockerfile con los comandos necesarios así como un script en bash para instalar las dependencias y las migraciones de la base de datos

```yaml
services:
  ▷Run Service
  db:
    image: postgres:16
    container_name: postgres_examen
    restart: always
    environment:
      POSTGRES_USER: user_pizza
      POSTGRES_PASSWORD: 1234
      POSTGRES_DB: laravel_pizza
    ports:
      - "5432:5432"

  ▷Run Service
  app:
    build: .
    container_name: "examen-despliegue"
    restart: always
    volumes:
      - .:/var/www/html
      - /var/www/html/vendor
      - /var/www/html/node_modules
    ports:
      - "8080:80"
    environment:
      - APP_NAME=Laravel
      - APP_DEBUG=true
      - APP_LOG=stdrr

      - APP_KEY=base64:m/psNCDbBA10PxoZK9zZBxrF3MiBTtM0Qq2qsBDTVpk=

      - DB_HOST=db
      - DB_CONNECTION=pgsql
      - DB_USERNAME=user_pizza
      - DB_PASSWORD=1234
      - DB_DATABASE=laravel_pizza
      - DB_PORT=5432
```

Docker-compose dev, base de datos y build

```
#!/usr/bin/env bash
echo "Running composer"
composer global require hirak/prestissimo
composer install --no-dev --working-dir=/var/www/html

echo "Caching config..."
php artisan config:cache

echo "Caching routes..."
php artisan route:cache

echo "Running migrations..."
php artisan migrate --force --seed
```

Script

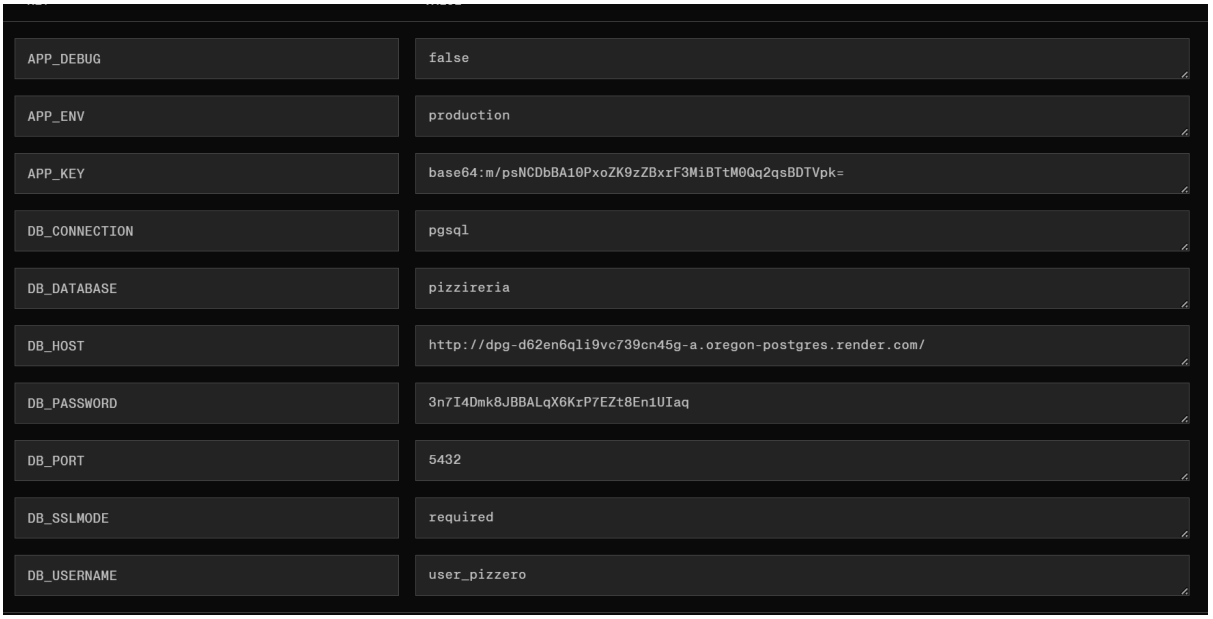
```
GET|HEAD  / ..........
GET|HEAD  api/calcular-importe/{id} ..........
POST      api/createOrder ..........
GET|HEAD  api/find/{id} ..........
GET|HEAD  api/hello ..........
GET|HEAD  api/pizza ..........
PUT       api/update ..........
GET|HEAD  api/user ..........
GET|HEAD  sanctum/csrf-cookie ..........
GET|HEAD  storage/{path} ..........
GET|HEAD  up ..........
```

Endpoints

Problemas: Cuidado con el puerto que postgre se pone nervioso si no es 5432 o eso ha parecido. No se muestran los endpoint en el navegador.

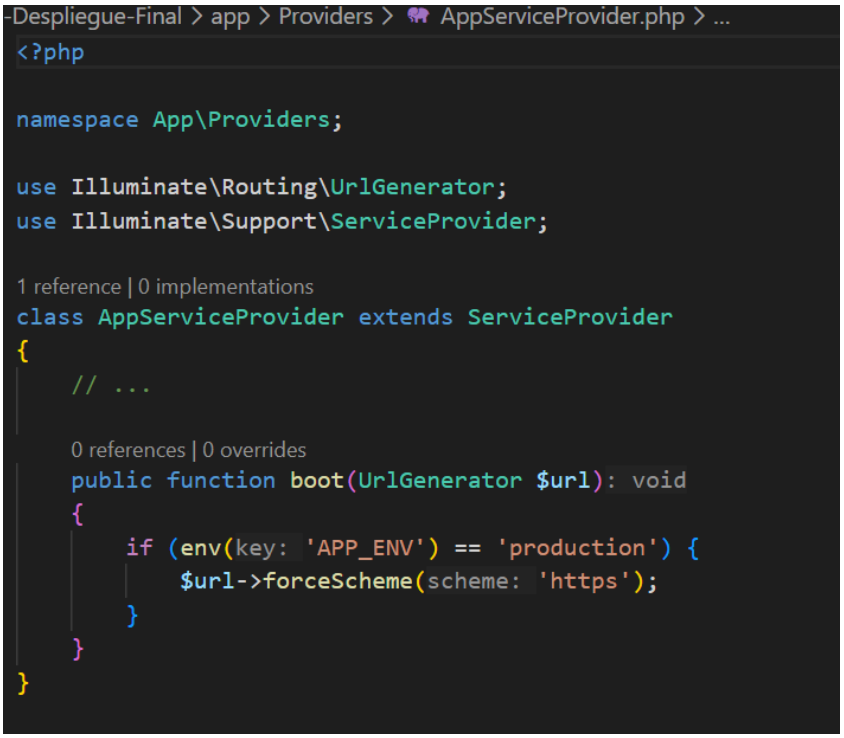
Render con PGSQL

Lo único que necesitamos añadir es forzar el uso de https en el código de AppServiceProvider. En render necesitamos añadir las credenciales de la base de datos y algunas variables de entorno de laravel.



APP_DEBUG	false
APP_ENV	production
APP_KEY	base64:m/psNCDbBA10PxoZK9zZBxrF3MiBTtM0Qq2qsBDTVpk=
DB_CONNECTION	pgsql
DB_DATABASE	pizzireria
DB_HOST	http://dpg-d62en6qli9vc739cn45g-a.oregon-postgres.render.com/
DB_PASSWORD	3n7I4Dmk8JBBALqX6KrP7EZt8En1UIaq
DB_PORT	5432
DB_SSLMODE	required
DB_USERNAME	user_pizzero

Variables de entorno en Render
(He cambiado el host, que en la imagen sale con una / de mas)



-Despliegue-Final > app > Providers > AppServiceProvider.php > ...

```php
<?php

namespace App\Providers;

use Illuminate\Routing\UrlGenerator;
use Illuminate\Support\ServiceProvider;

1 reference | 0 implementations
class AppServiceProvider extends ServiceProvider
{
    // ...

    0 references | 0 overrides
    public function boot(UrlGenerator $url): void
    {
        if (env(key: 'APP_ENV') == 'production') {
            $url->forceScheme(scheme: 'https');
        }
    }
}
```

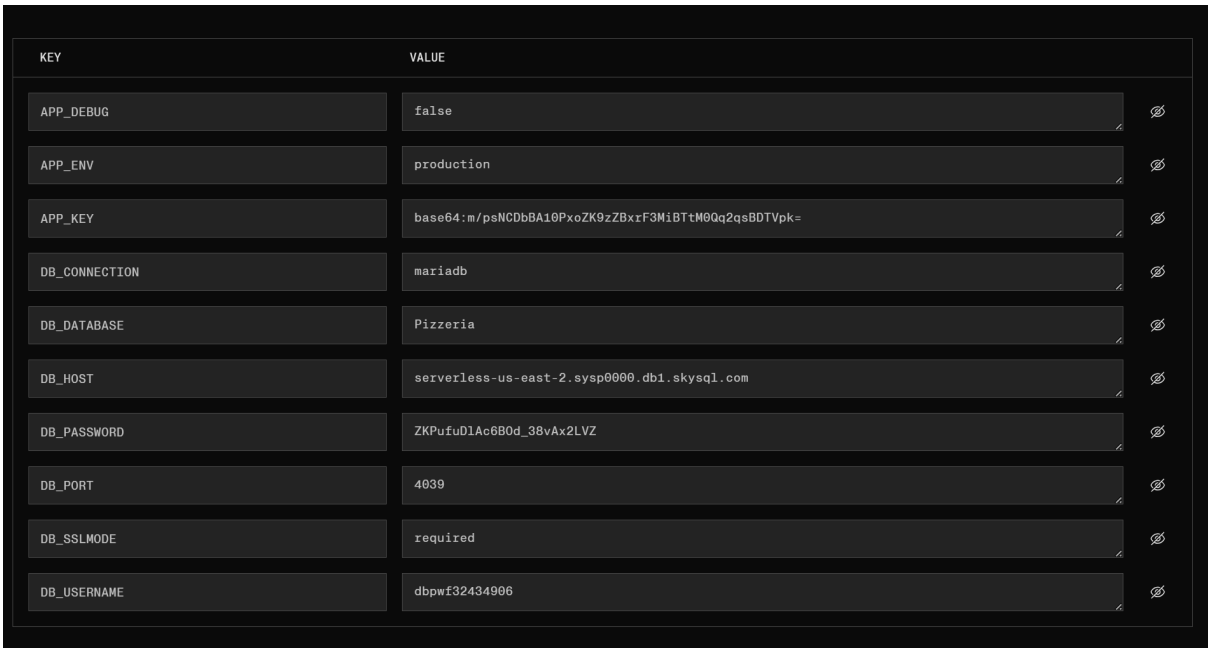
AppProvider cambiado

Url externa de la base de datos:
postgresql://user_pizzero:3n7l4Dmk8JBBALqX6KrP7EZt8En1Ulaq@dpg-d62en6qli9vc739cn45g-a.oregon-postgres.render.com/pizzireria

Pagina a visitar en render

Render con MySql

En la app de laravel no tenemos que hacer más cambios, solo enlazarla con MariaDb cambiando las variables de entorno y crear manualmente una base de datos con un cliente. (Web con Mysql)



Variables de entorno en render