Dulces Recetas Documentación Técnica

@Moisés Muñoz Aranda

Índice

Credenciales del proyecto Laravel

Usuario: moi

Contraseña: moi

Credenciales de aplicación web

http://moisesmuar.com.es/recetas

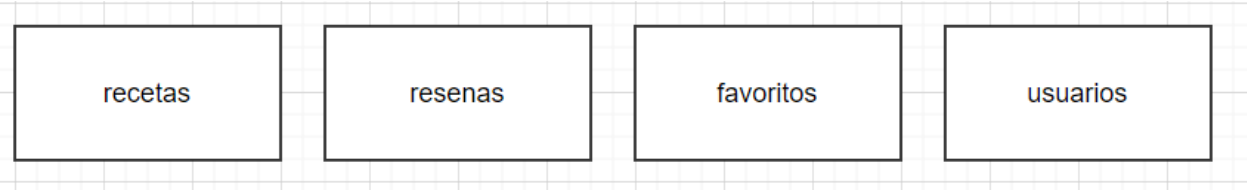
moi@moi.com // 123123123

Política versionado de ramas

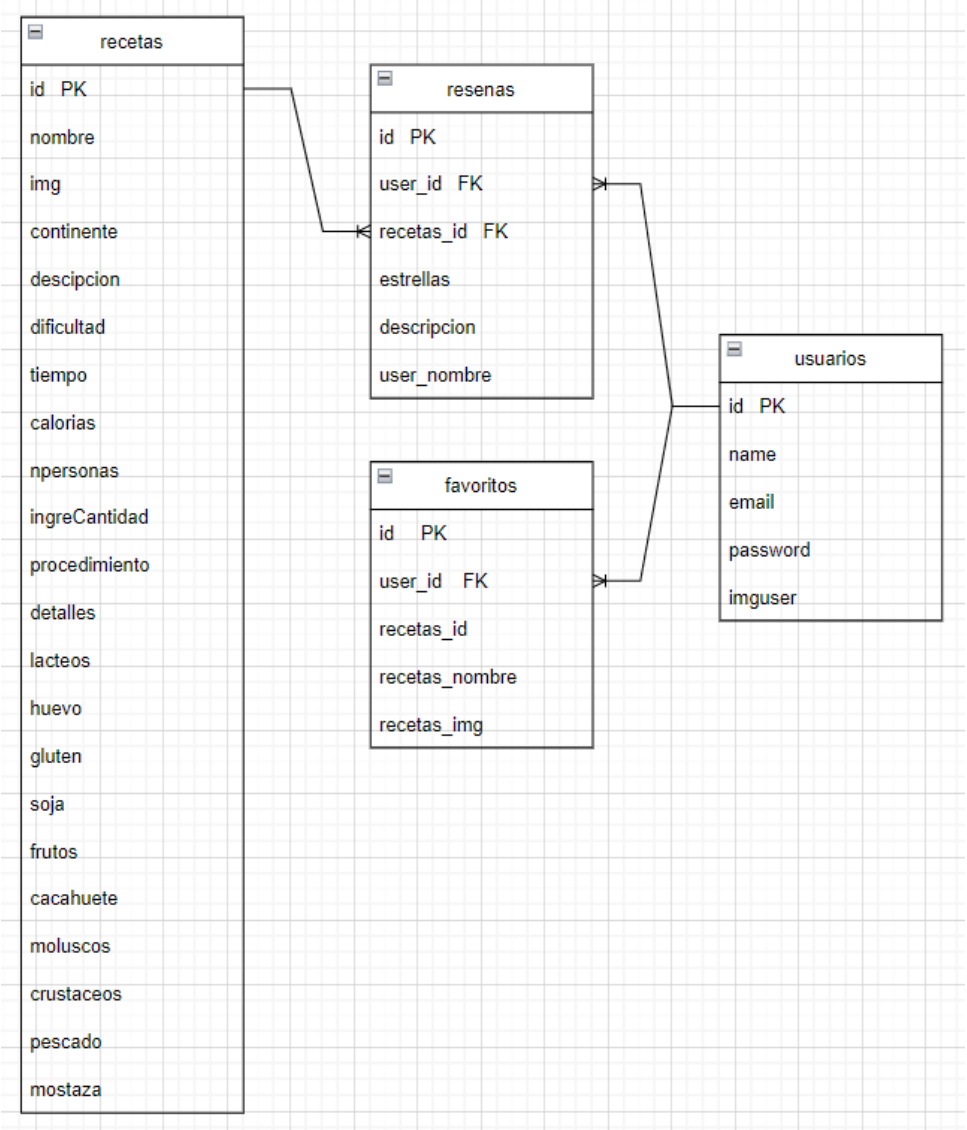
- Para el proyecto Dulces Recetas seguiré solo una rama de desarrollo, pues este entregable será una primera versión, realizada en el tiempo reducido del que se dispone.

Modelo BD y diagrama de clases.

Modelo Conceptual



Modelo Lógico



Modelo Físico

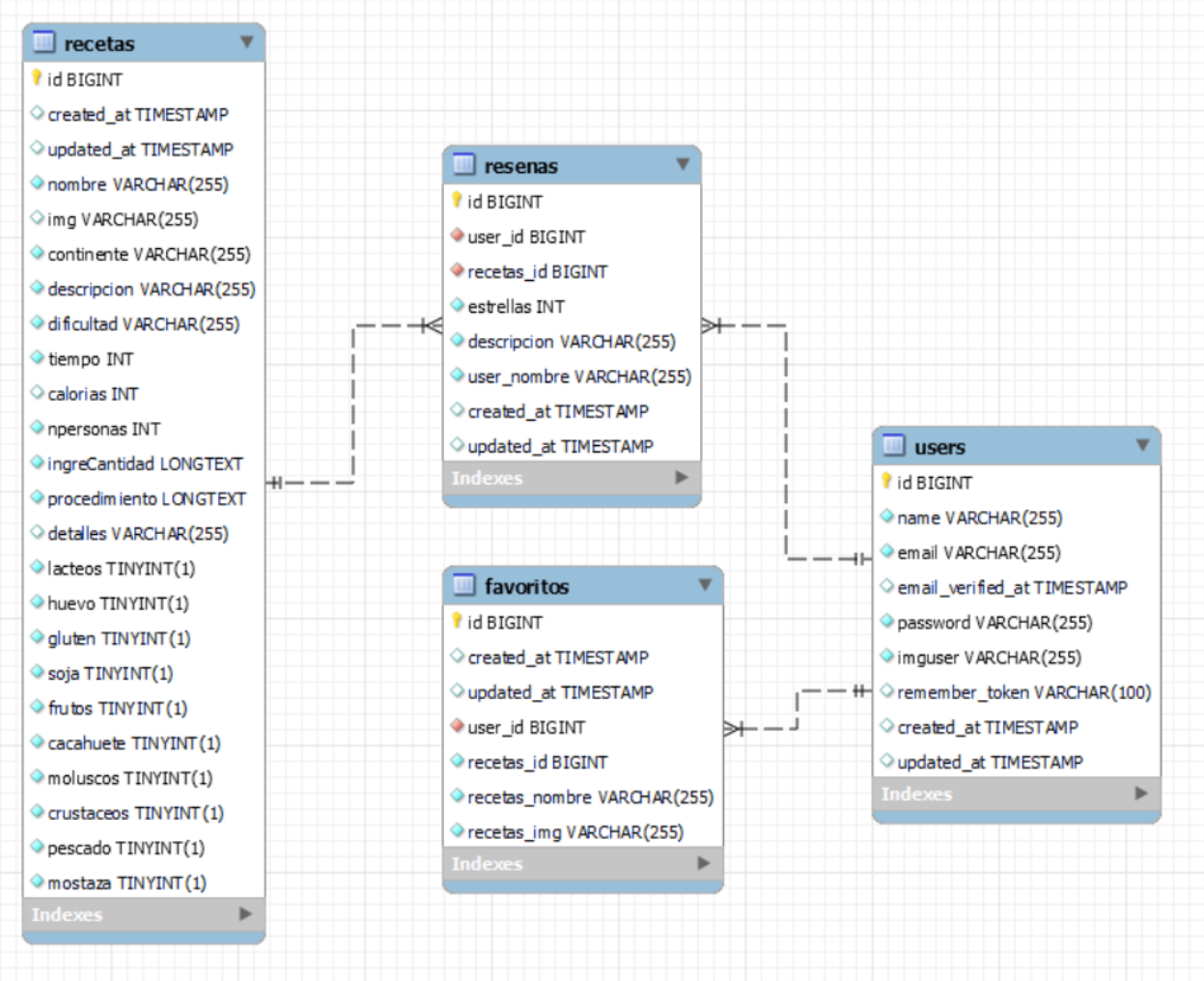
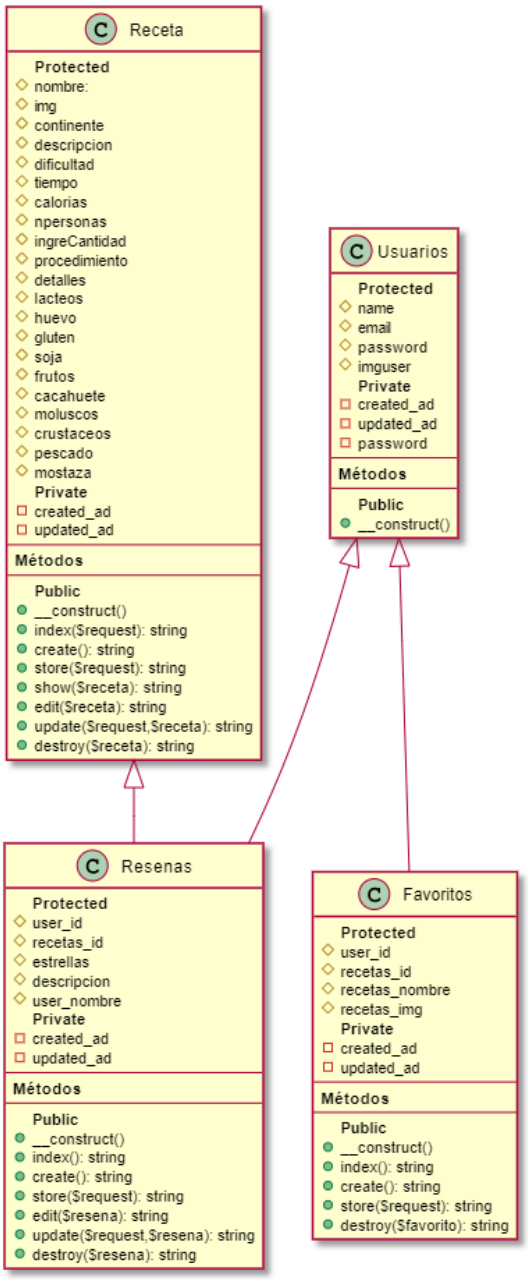


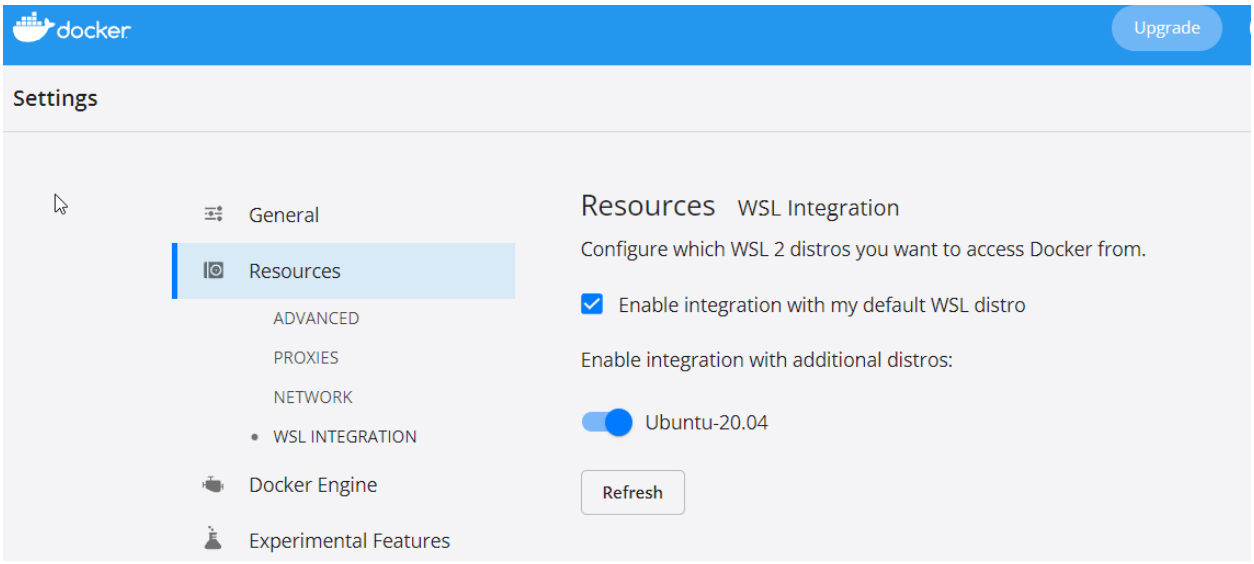
Diagrama de Clases



Comandos y requerimientos - Creación proyecto

Para instalar una aplicación Laravel en una máquina con Windows, instalaremos Docker Desktop y el subsistema de windows para Linux WSL2.

Habilitamos desde que distribución en WSL podremos acceder a Docker.



Comandos

- Creamos el proyecto laravel.

```
curl -s "https://laravel.build/dulcesRecetas?with=mysql" | bash
 // verificar en .env variables entorno BD
cd dulcesRecetas/

sail up   (ver si hay error )
sail down
sail up -d

  # usuarios login/registro
sail composer require laravel/breeze --dev
sail php artisan breeze:install

sail php artisan breeze:install react

 # Instala dependencias de js - compila todo (webpack)(hace un mix) y lo hace legible como web compatibiliza con navegadores
sail npm install && sail npm run dev
sail npm run dev

sail composer require inertiajs/inertia-laravel

 # creaccion de clase : modelo controlador
sail php artisan make:model Receta -mcr
sail php artisan make:model Favorito -mcr
sail php artisan make:model Resena -mcr
```

```
sail php artisan make:model Filtro -mcr
sail php artisan make:controller UserController

 # Migrar tablas a base datos
sail artisan migrate

// borramos bd y migramos de nuevo
sail artisan migrate:reset
sail artisan migrate
sail npm run dev   // ejecutamos al editar componentes


sail npm run watch  // ejecutamos para compilar automaticamente al editar js
sail php artisan route:cache // para cachear nueva rutas

 # ver todas la rutas del proyecto
sail php artisan route:list
```

Clases

Recetas

Controller

- Se requiere de autenticación del usuario para consultar una receta.
- index() componente que mostrará todas las recetas y sus reseñas (valoraciones)
- create() componente create formulario
- store() valida datos y los registra en base de datos
- show() componente ilustrador de una receta
- edit() componente formulario de edición
- update() valida datos actualizados y y los registra en base de datos
- destroy() borra un receta y sus registros de reseñas y favoritos
-

Model

- Definimos campos que podrán ser consultados y editados por Eloquent en el array

$fillable

- Añadimos clase para poder usarla en la función de relacción
- definimos relaciones con otros modelos (relaciones entre tablas)

```php
public function resenas(){
    return $this->hasMany( Resena::class );
}
```

Guardar y tratar imágenes

- Ejecutar dentro del proyecto para instalar librería

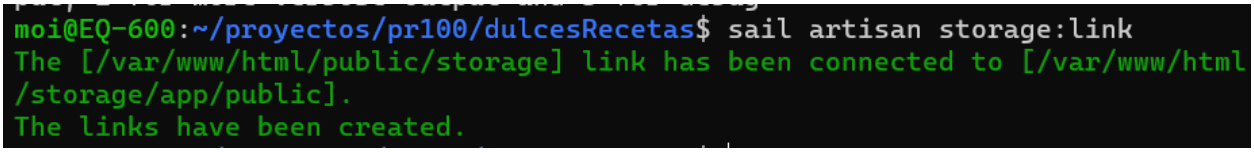
```
sail composer require intervention/image
```

- Añadir al controlador para uso de clase Image

```
use Intervention\Image\Facades\Image;
```

- Queremos subir la imagen al servidor en sistema de archivos /var/www/html/storage y guardaremos ruta en la base de datos

```
( -h info o ayuda para el comando )
$ sail artisan storage:link -h
( vincula /var/ww/html/storage con /var/www/storage/app/public )
$ sail artisan storage:link
```



- Imagen por defecto al crear un receta

```
cp /mnt/c/Users/MOI/Desktop/dulcesRecetas/default.png storage/app/public/images/
cd storage/app/public/images/
chmod 644 default.png
cd -
```

- Método store creación link imagen recibida

Guardamos $path en base de datos.

```
// guardar imagen servidor e insertar receta en bd
$path = Storage::disk('public')->putFile('images', $request['img']);
```

```
$path = asset('storage/' . $path);
```

Reseñas

Controller

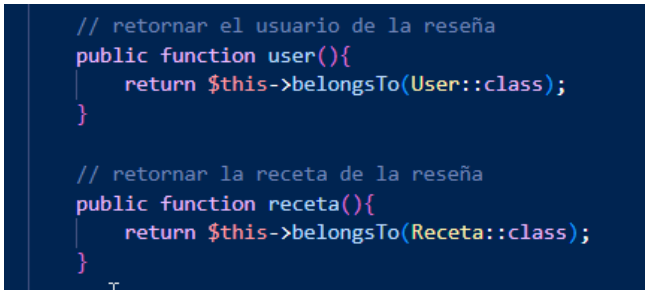
- Se requiere de autenticación del usuario para consultar una reseñas.
- index() componente que mostrará la reseñas (valoraciones) del usuario
- create() componente create formulario
- store() valida datos y los registra en base de datos
- edit() componente formulario de edición de reseña
- update() valida datos actualizados y y los registra en base de datos
- destroy() borra la reseña

Model

- Definimos campos que podrán ser consultados y editados por Eloquent en el array

$fillable

- Añadimos clase para poder usarla en la función de relacción
- definimos relaciones con otros modelos (relaciones entre tablas)

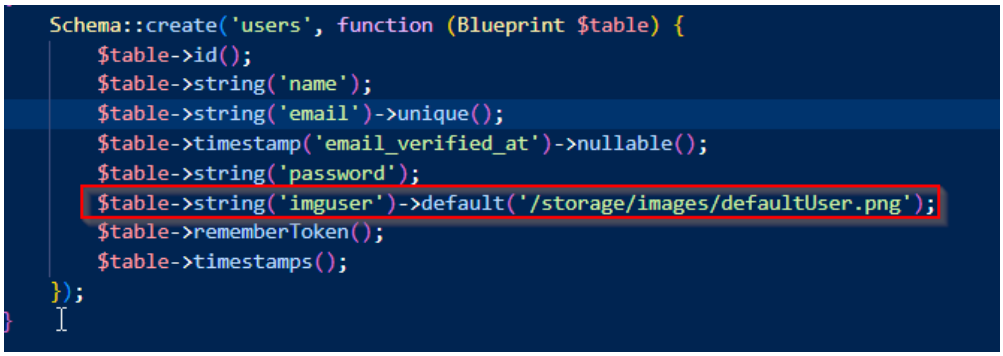


Usuarios

- Uso de paquete para **Laravel Breeze** permitiendo añadir el sistema de autenticación

Imagen por defecto usuario

- Añadimos la imagen dentro del proyecto en la ruta /resources/img/



Favoritos

Controller

- Se requiere de autenticación del usuario para consultar favoritos.
- index() componente que mostrará los favoritos del usuario
- create() componente create formulario
- store() valida datos y los registra en base de datos
- destroy() borra el favorito

Model

- Definimos campos que podrán ser consultados y editados por Eloquent en el array

$fillable

- Añadimos clase para poder usarla en la función de relacción
- definimos relaciones con otros modelos (relaciones entre tablas)

Filtro

Controller

- index() muestra recetas filtradas por alérgenos enviados desde componente FiltroAlergenos.js

Subida a servidor

- Entrar al servidor

```
eval `ssh-agent -s`
ssh-add ~/.ssh/dulces.pem
ssh ubuntu@13.58.166.228
```

- Instalamos docker damos permisos para ejecutar.

```
 #install docker
https://docs.docker.com/engine/install/ubuntu/
    #Change the permissions of /var/run/docker.sock for the current user.
sudo chown $USER /var/run/docker.sock

git clone https://github.com/moimu/pr100.git
```

- Instalamos php composer y dependencias (ejecutar en directorio de la aplicación)

```
docker run --rm \
    -u "$(id -u):$(id -g)" \
    -v $(pwd):/var/www/html \
    -w /var/www/html \
    laravelsail/php81-composer:latest \
    composer install --ignore-platform-reqs
```

- Crear alias para uso fácil de laravel sail

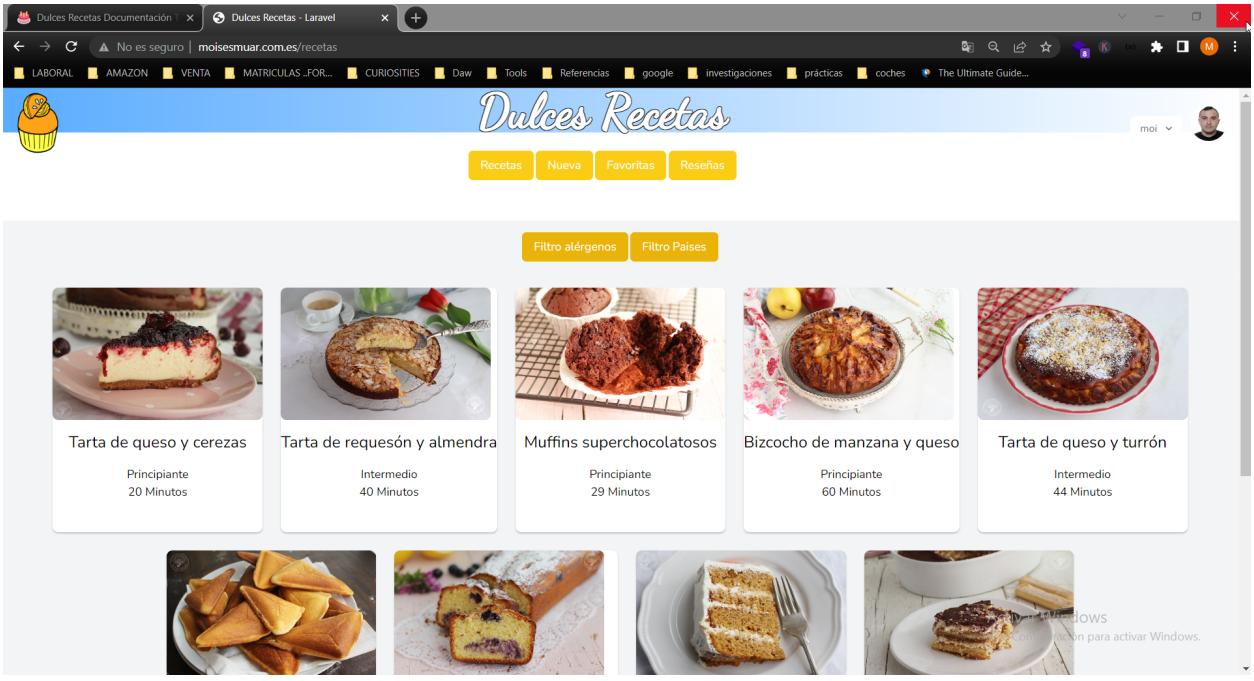
```
vim ~/.bashrc
alias sail='[ -f sail ] && bash sail || bash vendor/bin/sail'
source ~/.bashrc
```

- ! Copiar el .env de local a servidor

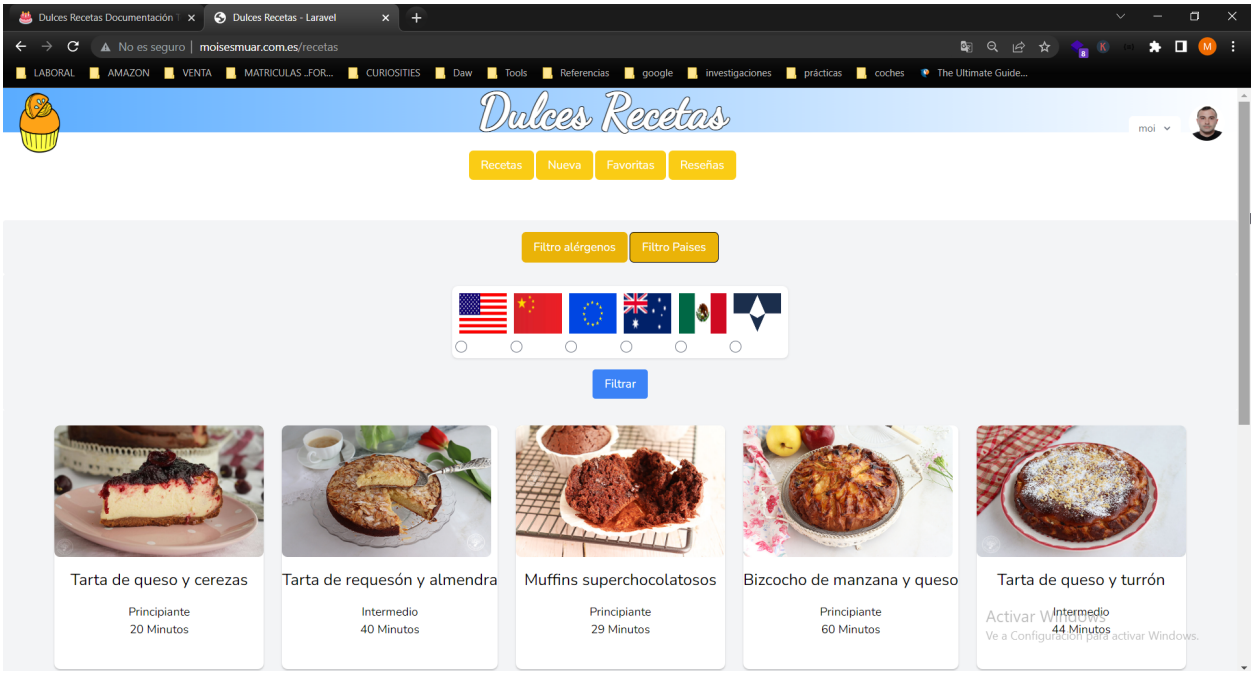
```
sail up -d

  # Clear the old boostrap/cache/compiled.php
php artisan clear-compiled
  # Recreate boostrap/cache/compiled.php
php artisan optimize
```

Asignamos Dominio a ip elástica.



- Con filtro de Países abierto



Referencias

https://raiolanetworks.es/?gclid=CjwKCAjw14uVBhBEEiwAaufYx-nhXzR-YQjaEP9lAe0pZDw-XwPys5hdf8qjFndsoDYeTbwoVqlO3xoCBT8QAvD_BwE

https://laravel.com/docs/9.x/sail

https://laravel.com/

https://laravel.com/docs/9.x#getting-started-on-windows

https://laravel.com/docs/9.x/sail

https://laravel.com/docs/9.x/migrations#available-column-types

https://laravel.com/docs/9.x/migrations#running-migrations

https://laravel.com/docs/9.x/controllers

https://laravel.com/docs/9.x/routing#route-parameters

https://laravel.com/docs/9.x/eloquent-relationships#one-to-many

https://laravel.com/docs/9.x/eloquent-relationships#one-to-many-inverse