**Documentación y pasos para TESTING**

**1. Configura la Base de Datos para Pruebas**

Laravel utiliza un archivo de configuración llamado **.env** para definir las credenciales de la base de datos. Sin embargo, durante las pruebas, Laravel utiliza el archivo **.env.testing** si existe. Si no se debe crear un archivo **.env.testing**.

Comando para crear archivo: cp .env .env.testing

Luego, se edita el archivo **.env.testing** para asegurarte de que la base de datos de pruebas esté configurada correctamente. Por ejemplo:

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=workwavedatabasefinal\_testing

DB\_USERNAME=root

DB\_PASSWORD=

Se debe asegurar de crear y de la existencia de la base de datos de pruebas **workwavedatabasefinal\_testing** exista en tu servidor MySQL. Si no existe, créala manualmente:

ConsultaSQL: CREATE DATABASE workwavedatabasefinal\_testing;

**2. Ejecuta las Migraciones para la Base de Datos de Pruebas**

Antes de ejecutar las pruebas, se deben asegurar de que las tablas necesarias estén creadas en la base de datos de pruebas. Ejecuta las migraciones con el entorno de pruebas:

Esto creará las tablas en la base de datos **workwavedatabasefinal\_testing.**

Comando: **php artisan migrate --env=testing**

**3. Verifica el Archivo phpunit.xml**

Laravel utiliza el archivo phpunit.xml para configurar las pruebas. Asegúrate de que el entorno de pruebas esté configurado correctamente en este archivo. Busca la línea que define el entorno y asegúrate de que sea testing:

<server name="APP\_ENV" value="testing"/>

También verifica que la variable **DB\_CONNECTION** esté configurada correctamente:

<server name="DB\_CONNECTION" value="mysql"/>

**4. Corrige el Uso de RefreshDatabase**

El trait RefreshDatabase asegura que la base de datos se limpie antes de cada prueba. Sin embargo, si las migraciones no están configuradas correctamente, puede causar problemas. Asegúrate de que todas las migraciones estén al día y que el trait esté correctamente implementado en tus pruebas:

**5. Ejecuta las Pruebas**

Después de realizar los pasos anteriores, ejecuta las pruebas:

Comando para ejecutar todos los test: php artisan test

Comando para ejecutar test individual:

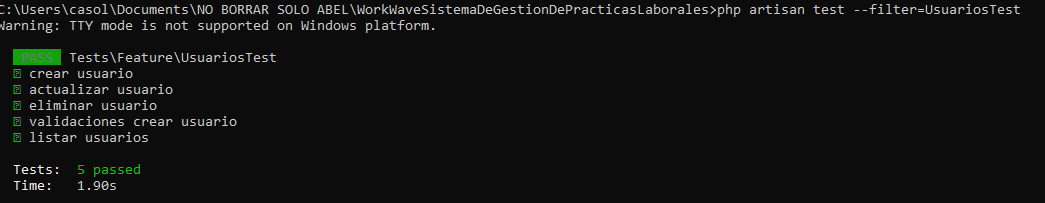
php artisan test --filter=UsuariosTest

php artisan test --filter=OfertasTest

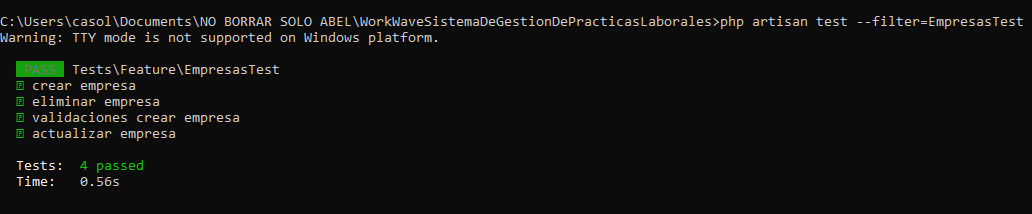
php artisan test --filter=EmpresasTest

**Resultados de TESTING**

**php artisan test --filter=UsuariosTest**

****

php artisan test --filter=EmpresasTest

****

php artisan test --filter=OfertasTest