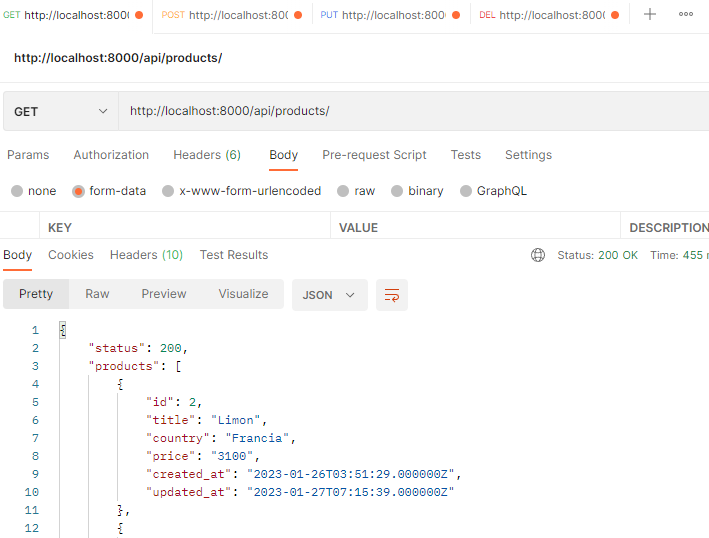
# PRUEBA TÉCNICA PARA DESARROLLADOR

## Puntos de la prueba técnica:

1. Crear una base de datos de 3 tablas (no importa el motor de BD).
2. Crear un API Rest en PHP usando el framework Laravel 5.
3. El API debe exponer un servicio para hacer CRUD a una tabla de la base de datos.
4. El FrontEnd debe estar creado en Angular.

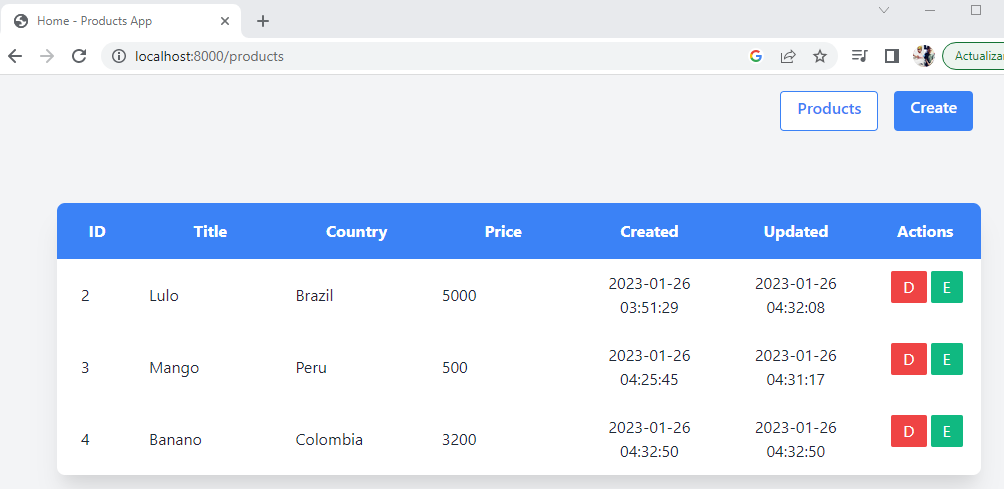
R/ A continuación explico el trabajo realizado:

* Se utilizo SQLite como gestor de base de datos, para que sea mas practica y ligera la revisión.
* Se realizo la creación del API Rest en formato JSON (GET,POST,PUT,DELETE), para la cual se cargó la evidencia correspondiente en el proyecto “proyectoAPIREST”



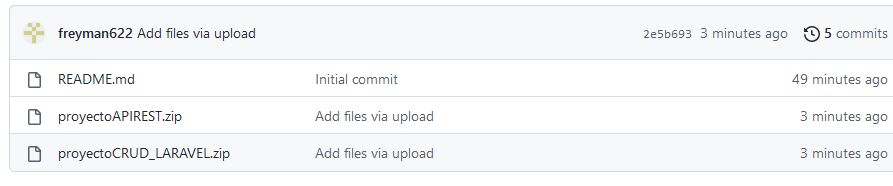
* Por temas de tiempo por mi trabajo actual y el tiempo dedicado inicial perdido intentando configurar el ambiente por el tema de las versiones de Laravel 5 y con php5 ya obsoleto y sin soporte, no alcance a realizar el FrontEnd en Angular para la parte GUI, **PERO** sin embargo se logró realizar esta GUI del CRUD en Laravel, aprovechando el ambiente configurado exitosamente con una versión Laravel más actualizada.

Para la cual se cargó la evidencia correspondiente en el proyecto “proyectoCRUD\_LARAVEL”



* Se envían proyectos en repositorio GitHub para su posterior descarga.

<https://github.com/freyman622/PruebaValley.git>



## Consideraciones.

1. Excelente orden y documentación del código.
2. Las tablas de la base de datos deben de estar relacionadas.
3. El API REST no tendrá formularios HTML, solo se encargará de exponer los servicios necesarios para hacer CRUD a una de las tablas.
4. Las respuestas del API REST deben ser en formato JSON.
5. La tabla a la que se le realizará CRUD debe tener al menos una llave foranea.
6. La aplicación web no debe utilizar JQuery, las peticiones deben hacerse como lo especifica AngularJS ($http) o Angular (HttpClient).
7. La aplicación web debe contar con todas las validaciones necesarias para realizar el CRUD.

## Plus

Los siguientes puntos no son obligatorios, pero se tendrán en cuenta si son implementados:

1. Diseño de la interfaz gráfica.
2. Documentación del código.
3. Utilización correcta de los métodos HTTP.
4. Utilización de un repositorio GIT.

## Tiempo

* 4 días a partir de recibir el correo.