Prueba entrevista FullStack PHP

Se requiere hacer 3 Servicios Restful donde :

1. Se registren Customers.
2. Se consulten Customer por dni o email.
3. Eliminar lógicamente el customer del sistema.
4. Cada servicio retornará a demás de la información solicitada un success true si se ejecuto correctamente y si no un false.

Indicaciones a seguir :

0. Se debe hacer un servicio de autenticación que retorne cuando el inicio de sesión sea exitoso un token con tiempo de vida (el token no debe repetirse y conformado de email, fecha y hora del inicio de sesión y un random de 200 a 500 todo encriptado en SHA1), este token deberá ser usado en todos los otros servicios a hacer y validando que si esta vencido no pueda ser usado y no de acceso al servicio. (Debe crear las tablas necesarias para que esta función se cumpla.)

1. Al registrar, asociar la commune y region del cliente, hacer todas las validaciones pertinentes. ejm no permitir registro de customer en commune y regiones que no estén relacionadas o no existan.

2. La consulta debe hacerse solo con customer activos (A), no con desactivo (I) o eliminados (trash), adicionalmente deberá retornar name, last_name, address (de no tener address retorna null en el campo), description region y commune. Realizar validaciones pertinentes.

3. El customer a eliminar debe de estar activo (A) o desactivo (I). En el caso de estar ya eliminado (trash) retornar “Registro no existe”. Hacer validaciones pertinentes.

4. El servicio debe de solo poder ser ejecutado en POST (Inserción) DELETE (Eliminación de registros) GET (Consulta de información).

5. Debe de ser desarrollado en php en el framework **Lumen** o **Laravel** (Ultimas versiones) utilizando todas las herramientas que estas entregan para su correcto funcionamiento.

6. El código debe de estar protegido para sql Injection y con un key de autenticación (Debe implementarse a través de middlewares) Toda validación previa (campos obligatorios, o validaciones de que la información que pasan exista debe hacerse a través de middlewares).

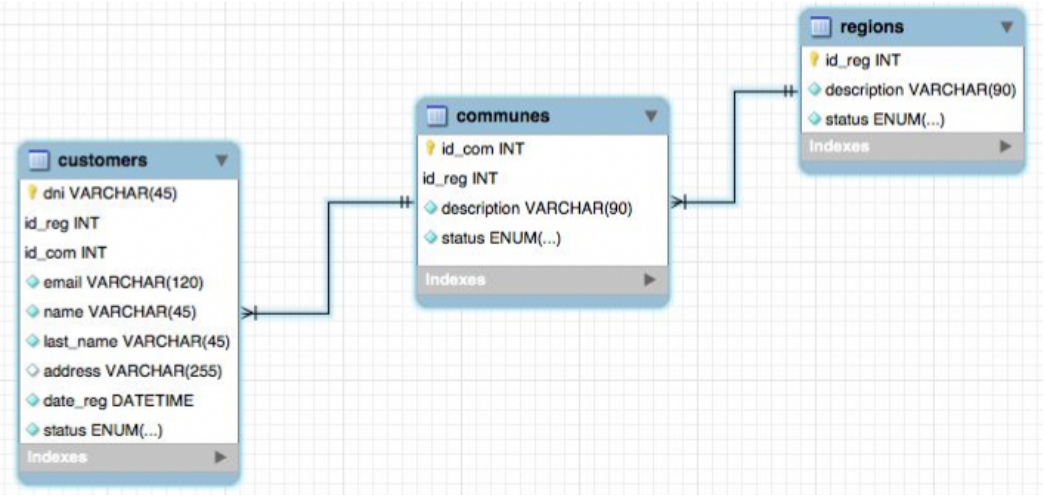
7. Documentación del servicio para su uso y para el desarrollador (Definición de servicios, metodos, pasos para su instalación, configuración y requerimientos mínimos) todo esto en el archivo ReadMe del proyecto.

8. Debe manejar logs de entrada y salida de información (puede estar en BD o en archivos de texto plano), así mismo, indicar de que IP proviene la información.

9. La plataforma debe tener un parámetro en el .env donde si se pasa a producción deja de guardar los logs de salida y solo guardar los de entrada.
Este parámetro APP_DEBUG siempre debe estar en False.

10. Al terminar el proyecto se debe de subir a un github publico y copiar el repo y pasarlo de respuesta en el correo donde te llego este documento.

Modelo básico de la BD a usa.



SQL Básico a Utilizar

```
---------------------
-- Schema mydb
---------------------
  DROP SCHEMA IF EXISTS `mydb` ;

 ---------------------
-- Schema mydb
 ---------------------


CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 COLLATE
utf8_general_ci ;

USE `mydb` ;

---------------------
-- Table `mydb`.`regions`
---------------------


  DROP TABLE IF EXISTS `mydb`.`regions` ;

CREATE TABLE IF NOT EXISTS `mydb`.`regions` (
`id_reg` INT NOT NULL AUTO_INCREMENT COMMENT '', `description` VARCHAR(90)
NOT NULL COMMENT '',
`status` ENUM('A', 'I', 'trash') NOT NULL DEFAULT 'A' COMMENT '', PRIMARY KEY
(`id_reg`) COMMENT '')
ENGINE = MyISAM;

---------------------
-- Table `mydb`.`communes`
---------------------


  DROP TABLE IF EXISTS `mydb`.`communes` ;

CREATE TABLE IF NOT EXISTS `mydb`.`communes` (
`id_com` INT NOT NULL AUTO_INCREMENT COMMENT '', `id_reg` INT NOT NULL
COMMENT '',
`description` VARCHAR(90) NOT NULL COMMENT '',
`status` ENUM('A', 'I', 'trash') NOT NULL DEFAULT 'A' COMMENT '', PRIMARY KEY
(`id_com`, `id_reg`) COMMENT '',
INDEX `fk_communes_region_idx` (`id_reg` ASC) COMMENT '')
ENGINE = MyISAM;

---------------------
-- Table `mydb`.`customers`
---------------------
```

```
DROP TABLE IF EXISTS `mydb`.`customers` ;

CREATE TABLE IF NOT EXISTS `mydb`.`customers` (
`dni` VARCHAR(45) NOT NULL COMMENT 'Documento de Identidad',
`id_reg` INT NOT NULL COMMENT '',
`id_com` INT NOT NULL COMMENT '',
`email` VARCHAR(120) NOT NULL COMMENT 'Correo Electrónico',
`name` VARCHAR(45) NOT NULL COMMENT 'Nombre',
`last_name` VARCHAR(45) NOT NULL COMMENT 'Apellido',
`address` VARCHAR(255) NULL COMMENT 'Dirección',
`date_reg` DATETIME NOT NULL COMMENT 'Fecha y hora del registro',
`status` ENUM('A', 'I', 'trash') NOT NULL DEFAULT 'A' COMMENT 'estado del registro:\nA
: Activo\nI : Desactivo\ntrash : Registro eliminado',
PRIMARY KEY (`dni`, `id_reg`, `id_com`) COMMENT '',
INDEX `fk_customers_communes1_idx` (`id_com` ASC, `id_reg` ASC) COMMENT '',
UNIQUE INDEX `email_UNIQUE` (`email` ASC) COMMENT '')
ENGINE = MyISAM;
```