Prueba Técnica Backend Developer

Caso de uso: aplicación de pacientes médicos.

Objetivo: Desarrollar una API REST con PHP para la gestión de pacientes de un hospital, el cual permitirá a los médicos del hospital realizar la búsqueda de un paciente, crear nuevos pacientes y agregar diagnósticos a los pacientes.

Tecnologías:

- PHP 8
- Framework Laravel v10
- MsSQL o en su defecto PostgreSQL.

Requisitos:

El arquitecto del proyecto ha proporcionado las siguientes especificaciones:

Un paciente tiene, al menos, los siguientes datos:

Field	**Description**	**Mandatory for creation?**	**Restrictions**
ID	Unique identifier. This field is generated automatically when a new patient is created.	Yes	N/A
Document	Identification document	Yes	String (20). Just numbers are allowed.
First name	Patient first name	Yes	String (255)
Last name	Patient last name	Yes	String (255)
Birth_date	Patient Birthday	Yes	String (30). It must be a date.
Email	Contact email	Yes	String (255). It must be a valid email.
Phone	Contact phone	Yes	String (20). Just numbers are allowed.
Genre	Patient Genre	Yes	String (30). Just “Male” and “Female” options are allowed.

Los diagnósticos tendrán los siguientes datos:

Field	Description	Mandatory for creation?	Restrictions
ID	Unique identifier. This field is generated automatically when a new diagnostic is created.	Yes	N/A
Name	Diagnostic name	Yes	String (255)
Description	Description	No	String (255)

La asignación de diagnósticos tendrá los siguientes datos:

Field	Description	Mandatory for creation?	Restrictions
ID	Unique identifier. This field is generated automatically when a new diagnostic is created.	Yes	N/A
Patient	Patient Identifier	Yes	Number
Diagnostic	Diagnostic Identifier	Yes	Number
Observation	Diagnostic observations	No	String (255)
Creation	Creation Date	Yes	String (255) It must be a datetime.

- Debe desarrollar un método para registrar al paciente. Debes tener en cuenta las siguientes reglas:
    - Se deben validar las restricciones mencionadas en la tabla de datos del paciente.
    - Si el paciente ya ha sido registrado, se debe retornar un mensaje de error.
    - Si el paciente se ha registrado correctamente, se debe retornar un código de respuesta exitoso con la información completa del paciente (esto incluye la identificación del paciente).
    - Si no se pudo registrar al paciente, debe devolver un código de respuesta HTTP 419. El cuerpo de la respuesta debe incluir un mensaje de error con una descripción adecuada al negocio.
- Debe desarrollar un método para actualizar la información de un paciente específico. Aplicar las reglas del método anterior.

- Debe desarrollar un método para la asignación de diagnósticos a un paciente. Debes tener en cuenta las siguientes reglas:
    - Debe registrarse la fecha en la que el paciente recibió el diagnóstico.
    - Debe ser posible el ingreso de una observación.
- Se debe desarrollar un método para obtener el listado de pacientes registrados, por cada paciente se debe incluir su información y los diagnósticos recibidos. Dicho listado debe permitir la paginación de los resultados.
- Se debe desarrollar un método para la búsqueda de pacientes registrados por su nombre, apellido y número de documento.
- Se debe desarrollar un método para obtener el listado de los 5 diagnósticos más comunes en los pacientes en los últimos 6 meses.
- Se debe desarrollar un método para eliminar un paciente existente, si el paciente posee diagnósticos estos también deben ser eliminados.
- Debe crear las tablas usando una base de datos relacional..
- Se debe cargar la base de datos con datos de prueba.
- El api debe contener, al menos, las siguientes capas: controlador, servicio, acceso a datos (modelo).

Entregables:

- Repositorio público en GitHub con el código fuente completo de la api.
- Diagrama de la estructura de base de datos (puede incluirse en el repositorio).
- Instrucciones claras sobre cómo configurar y ejecutar la aplicación localmente.
- Documentación de la api en Postman o Swagger. (Excluyente)
- Url de la aplicación desplegada en un servidor gratuito (Opcional).

Consideraciones:

- Se alienta la implementación de características adicionales que demuestren habilidades creativas y conocimientos avanzados.

- Ten en cuenta que evaluaremos la calidad de tu código, la estructura del proyecto, el cumplimiento de los requisitos, entregables y la implementación de buenas prácticas de programación.