

# Desafío Backend

El Colegio de Magia y Hechicería Hogwarts requiere diseñar su sistema de inscripciones para el próximo periodo académico.

Para tal fin, se requiere exponer una API REST, construida usando el framework .NetCore(> 3.0).

Deben exponerse los endpoints necesarios para soportar las siguientes operaciones:

- +Enviar Solicitud de Ingreso
- +Actualizar solicitud de Ingreso
- +Consultar todas las solicitudes enviadas
- +Eliminar la solicitud de ingreso.

La solicitud de ingreso debe indicar como mínimo los siguientes datos del aspirante:

- +Nombre (Solo letras, máximo 20 caracteres)
- +Apellido (Solo letras, máximo 20 caracteres)
- +Identificación (Solo números, máximo 10 dígitos)
- +Edad (Solo números, máximo 2 dígitos)
- +Casa a la que aspira pertenecer (Solo 4 posibles opciones: "Gryffindor", "Hufflepuff", "Ravenclaw", "Slytherin")

Cualquier solicitud de adición o edición que incumpla los criterios anteriormente descritos, debe ser descartada.

En lo referente a la persistencia de la información, el criterio/tecnología a emplear para almacenarla es libre(Memoria, InMemoryDb(SqlLite, EFNetCore ), txt files... )

Con el fin de probar la funcionalidad debe adjuntarse un documento explicando como consumir los servicios con un cliente HTTP

o en su defecto un plan de pruebas con alguna de las herramientas disponibles para tal fin (Postman, Jmeter...).

Para facilitar la extensión de la funcionalidad en un futuro cercano, apreciamos y valoramos el uso de la siguiente "magia":

- +Uso de patrón de arquitectura: Capas.

- +Uso de patrones de diseño (Creacionales, estructurales).

- +Se sugiere el uso de inyección de dependencias para comunicar las múltiples capas.

- +Se recomienda considerar el uso de un middleware para la implementación del estándar OpenAPI(swagger).

Algunos Mantras

- +Explícito es mejor que implícito.

- +Simple es mejor que complejo

- +Complejo es mejor que complicado.

- +Plano es mejor que anidado.

- +Espaciado es mejor que denso.

- +La legibilidad es importante.

El resultado es importante, se sugiere tener en cuenta que la solución presentada este pensada para ser fácilmente desplegada o ejecutada en cualquier ambiente sin mayores dependencias.