



Programación Java

► Validaciones y Excepciones

// Práctica Individual

Objetivos

Tomando como base el proyecto “[ObtenerDiploma](#)” que se adjunta completamente implementado se deberán realizar las validaciones para los datos de entrada (request) y excepciones necesarias para las entradas y salidas, para controlar los diferentes escenarios posibles propuestos en cada ejercicio.



User Story

Historia de usuario narrada desde el punto de vista del personal administrativo del instituto educativo. El servicio (end-point) es expuesto desde el instituto educativo para ser consumido por cualquier persona autorizada de la misma que lo solicite.

- ✓ **US-0001:** Calcular el promedio de todas las notas obtenidas por un alumno teniendo en cuenta las materias cursadas. Elaborar un mensaje personalizado dependiendo del promedio.

Representación JSON

Se define el payload necesario para proceder con la carga, como así también la respuesta (response) en el formato JSON.

ENTRADA (PAYLOAD):

```
{
  "studentName": "Juan",
  "subjects":
    [
      {
        "name": "Matemática",
        "score": 10
      },
      {
        "name": "Física",
        "score": 7
      },
      {
        "name": "Química",
        "score": 6
      }
      ...
    ]
}
```

**SALIDA (RESPONSE):**

```
{
  "studentName": "Juan",
  "message": "El Alumno Juan ha obtenido un promedio de 7.6",
  "averageScore": 7.6,
  "subjects":
    [
      {
        "name": "Matemática",
        "score": 10
      },
      {
        "name": "Física",
        "score": 7
      },
      {
        "name": "Química",
        "score": 6
      }
      ...
    ]
}
```

¡Buena suerte!



Ejercicio 1

Validación de los campos de entrada

Se deberán realizar las validaciones necesarias en los campos referidos al **StudentDTO** y **SubjectDTO**, utilizando las anotaciones necesarias para validar las entradas.

Validaciones:

Atributo	Tipo	¿Obligatorio?	Validación	Mensaje de error
studentName	String	SI	<ul style="list-style-type: none">• Que el campo no esté vacío.• Que empiece con mayúscula.• Longitud máxima de 50 caracteres.	<ul style="list-style-type: none">• El nombre del alumno no puede estar vacío.• El nombre del alumno comenzar con mayúscula.• La longitud del nombre no puede superar los 50 caracteres.
subjects	List<SubjectDTO>	SI	<ul style="list-style-type: none">• Que la lista no esté vacía.	<ul style="list-style-type: none">• La lista no puede ser vacía.
name	String	SI	<ul style="list-style-type: none">• Que el campo no esté vacío.• Que empiece con mayúscula.• Longitud máxima de 30 caracteres.	<ul style="list-style-type: none">• El nombre de la materia no puede estar vacío.• El nombre de la materia debe comenzar con mayúscula.• La longitud del nombre no puede superar los 30 caracteres.
score	Double	SI	<ul style="list-style-type: none">• Que el campo no esté vacío.• Mínimo 0.0• Máximo 10.0	<ul style="list-style-type: none">• La nota no puede estar vacía.• La mínima nota es 0.0.• La máxima nota es 10.0.



Ejercicio 2

Excepciones de datos de entrada

Se deberán controlar las excepciones de datos de entrada utilizando las siguientes clases

ErrorDTO y ObtenerDiplomaExceptionController.



Ejercicio 3

Excepciones de respuesta

Se deberán controlar las excepciones de respuesta con las siguientes clases:

ErrorDTO y ObtenerDiplomaExceptionController.



Ejercicio 4

Extra Bonus

Se deberán controlar otras excepciones que usted crea necesarias, siempre manejandolas con las clases **ErrorDTO y ObtenerDiplomaExceptionController.**