

Marcos de Desarrollo Web

Sesión 11



Universidad
Tecnológica
del Perú

¿Tienen alguna consulta o duda sobre la clase previa?



Logro de la sesión

Al finalizar la sesión, el estudiante construye un proyecto web con conexión a una base de datos relacional empleando la biblioteca Spring Validator para realizar validaciones de consistencia de datos.

¿Qué es la validación de datos? ¿Cómo se implementa en proyectos web?



Add Person

Name

Full Name



size must be between 5 and 2147483647
must not be empty

Email

hello

must be a well-formed email address

Age

1

must be greater than or equal to 18

Add User

¿Cuál es la importancia de validar los datos ingresados por los usuarios?



Universidad
Tecnológica
del Perú

Contenido

Spring validator

- Relaciones entre tablas
- Validación de datos
- Restricciones en las tablas



Spring validator

- A pesar de que Spring Boot permite una integración fluida con validadores personalizados, **Hibernate Validator** es el estándar predominante para llevar a cabo la validación.
- Para ello se debe agregar la siguiente dependencia al proyecto de Spring Boot:

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-validation</artifactId>  
</dependency>
```

Anotaciones de validación

- Las anotaciones de validación se utilizan para garantizar que los datos de entrada a las aplicaciones cumplan con ciertos requisitos antes de ser procesados.
- Estas anotaciones son parte del **Java Bean Validation API (JSR 380)**, que Spring integra para proporcionar validación automática y sencilla en aplicaciones **Spring Boot** y **MVC**.

@NotNull

@NotEmpty

@NotBlank

@Size

@Min

@Max

@Pattern

@Email

@Past

@Future

@Valid

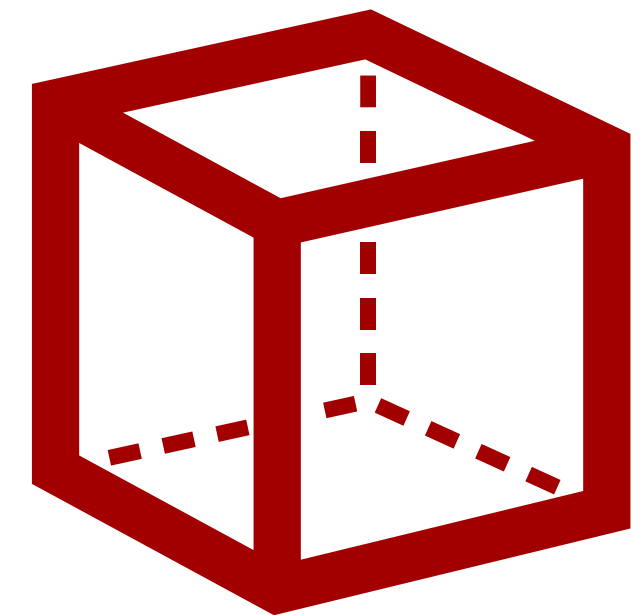


Universidad
Tecnológica
del Perú

Funcionamiento de la validación en Spring

Uso en Entidades

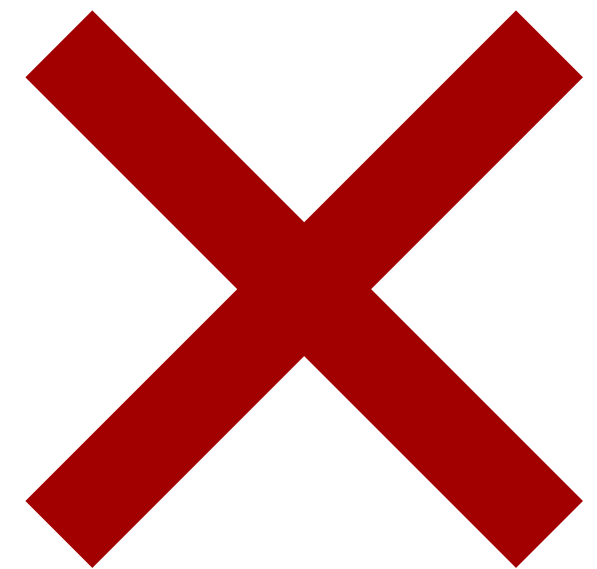
- Las anotaciones se colocan directamente en los campos de las entidades o DTOs (Data Transfer Objects).



Funcionamiento de la validación en Spring

Manejo de Errores

- Si los datos no cumplen con las restricciones de validación, se generan errores que pueden ser manejados y devueltos al cliente para corrección.



Funcionamiento de la validación en Spring

Activación de la Validación

- En Spring Boot y Spring MVC, la validación se activa automáticamente utilizando las anotaciones `@Valid` o `@Validated` en los métodos del controlador que reciben estas entidades como parámetros.



Relaciones entre tablas

- **@ManyToOne** y **@OneToMany**: Estas anotaciones suelen utilizar claves foráneas para establecer las relaciones entre las tablas.
- **@OneToOne**: Utiliza claves primarias o foráneas para vincular una tabla directamente con otra, garantizando que solo haya una relación única.
- **@ManyToMany**: Utiliza una tabla intermedia (también llamada tabla de unión) para gestionar la relación entre las dos entidades.

Validación de datos

@NotEmpty

- **Descripción:** Asegura que el campo no sea **null** y que, si es un String, Collection, Map o Array, **no esté vacío**.
- **Uso:** Utilizado principalmente para cadenas y colecciones que no deben estar vacías.

Validación de datos

@NotNull

- **Descripción:** Asegura que el campo no sea **null**.
- **Uso:** Se utiliza para campos que deben contener un valor, pero pueden ser cadenas vacías, listas vacías, etc.

Validación de datos

@NotBlank

- **Descripción:** Similar a **@NotEmpty**, pero específicamente para cadenas de texto. Asegura que la cadena no sea **null**, no esté vacía y no contenga solo espacios en blanco.
- **Uso:** Ideal para validar entradas de texto donde los espacios en blanco no son válidos.

Validación de datos

@Size

- **Descripción:** Asegura que la longitud de una cadena, la cantidad de elementos en una colección o la cantidad de elementos en un array esté dentro de un rango específico.
- **Parámetros:** min y max para especificar el rango permitido.
- **Ejemplo:** @Size(min = 5, max = 15)

Validación de datos

@Min y @Max

- **Descripción:** Asegura que el valor numérico sea mayor o igual (@Min) o menor o igual (@Max) al valor especificado.
- **Uso:** Se aplica a campos numéricos para validar rangos específicos.
- **Ejemplo:** @Min(18) @Max(65)

Validación de datos

@Pattern

- **Descripción:** Asegura que el campo coincide con una expresión regular especificada.
- **Parámetros:** regexp para definir la expresión regular y flags para modificar su comportamiento.
- **Ejemplo:** `@Pattern(regexp = "^[a-zA-Z0-9]+$")`

Validación de datos

@**Future** y @**Past**

- **Descripción:** Asegura que una fecha sea en el futuro (@**Future**) o en el pasado (@**Past**).
- **Uso:** Se aplica a campos de tipo Date o LocalDate.

Validación de datos

@Email

- **Descripción:** Asegura que el campo contenga una dirección de correo electrónico válida.
- **Uso:** Utilizado para validar direcciones de correo electrónico.

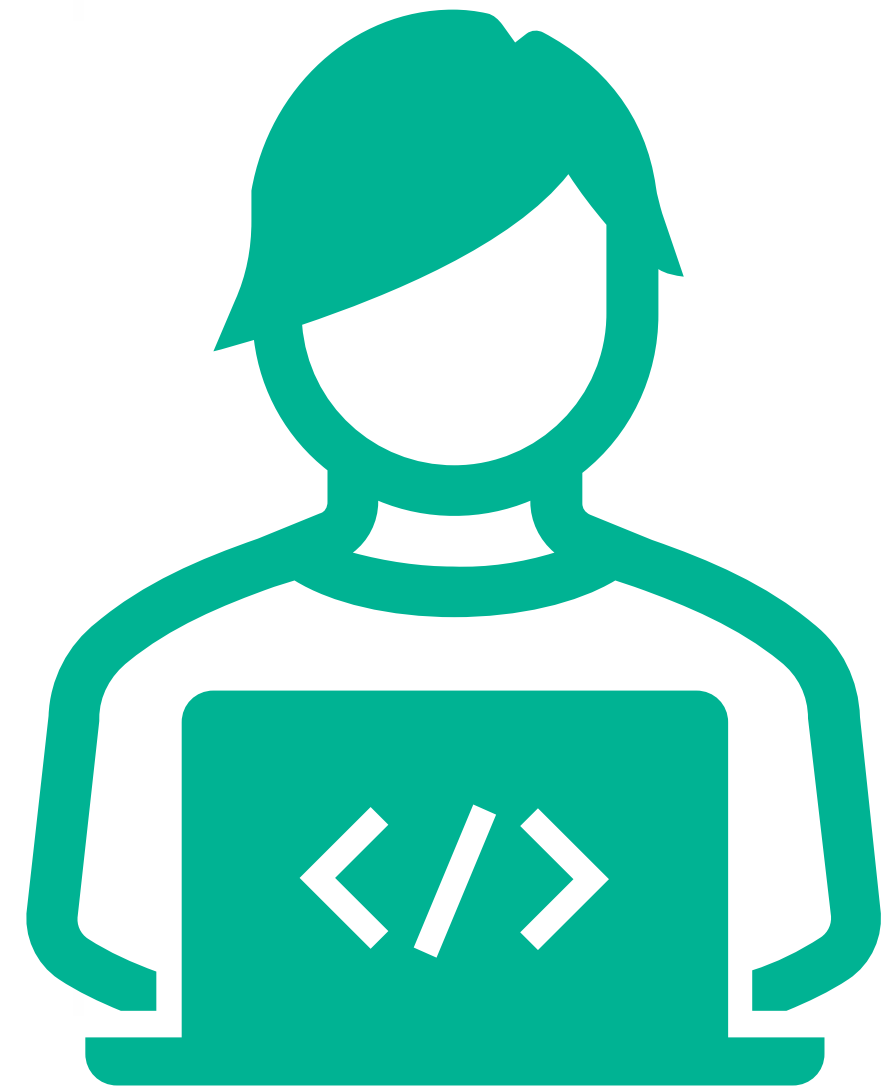
Relación entre anotaciones y base de datos

- Las anotaciones **@ManyToOne**, **@OneToMany**, **@OneToOne**, y **@ManyToMany** se utilizan para definir las relaciones entre entidades en un modelo de datos relacional.
- Estas anotaciones son parte de la **Java Persistence API (JPA)** y ayudan a mapear las relaciones de base de datos en objetos Java.



Programando

- Construir un sitio web empleando **validación** de campos al insertar o actualizar valores en una tabla.



¿Tienen alguna consulta o duda?



Actividad



Resolver la actividad planteada en la plataforma.

Cierre

¿Qué hemos aprendido hoy?



Elaboramos nuestras conclusiones sobre el tema tratado



**Universidad
Tecnológica
del Perú**