**ProductoDto**

El ejemplo que proporciona es una implementación de un DTO (Objeto de Transferencia de Datos) llamado ProductoDto. El DTO es una clase utilizada para transferir datos entre diferentes capas de una aplicación, en este caso entre la capa de controladores y la capa de servicios.

La clase ProductoDto tiene una serie de atributos, cada uno de los cuales representa un campo de datos que se puede transferir. Estos atributos se han anotado con diferentes anotaciones que proporcionan información sobre la validación y las restricciones de cada campo.

Las anotaciones @Getter y @Setter se utilizan para generar automáticamente los métodos getter y setter para cada atributo. Esto permite acceder y modificar los valores de los atributos desde otras partes de la aplicación.

Las anotaciones @NotBlank y @Size se utilizan para validar la entrada de datos para los campos "nombre", "descripcion" y "direccion". La anotación @NotBlank asegura que estos campos no estén vacíos, mientras que la anotación @Size establece los requisitos de longitud mínima y máxima para cada campo.

La clase también tiene una lista de objetos de tipo Imagen, una lista de objetos de tipo Long, dos campos de tipo Long para id\_Ciudad e id\_Categoria y una lista de objetos de tipo Política. Estos campos también tienen anotaciones que establecen requisitos de validación y restricciones para cada campo.

En resumen, la clase ProductoDto es un ejemplo de DTO utilizado para transferir datos entre diferentes capas de una aplicación. Cada campo de datos en el DTO tiene anotaciones que establecen requisitos de validación y restricciones para asegurar que los datos sean precisos y válidos.

**PuntuacionDto**

En este ejemplo se muestra la implementación de un DTO llamado PuntuacionDto, que se utiliza para transferir datos relacionados con la puntuación de un producto realizado por un usuario.

La clase tiene tres atributos, cada uno de los cuales representa un campo de datos que se puede transferir. Los atributos "id\_Producto" y "id\_Usuario" son de tipo Long y están anotados con la anotación @NotNull, lo que significa que no pueden ser nulos.

El tercer atributo, "puntuacion", es también de tipo Long y tiene restricciones adicionales establecidas mediante las anotaciones @Min y @Max. La anotación @Min establece que la puntuación mínima debe ser 1, mientras que la anotación @Max establece que la puntuación máxima debe ser 5.

La clase PuntuacionDto utiliza las anotaciones @Getter y @Setter para generar automáticamente los métodos getter y setter para cada atributo.

En resumen, la clase PuntuacionDto es un ejemplo de DTO utilizado para transferir datos relacionados con la puntuación de un producto realizado por un usuario. Cada campo de datos en el DTO tiene anotaciones que establecen requisitos de validación y restricciones para asegurar que los datos sean precisos y válidos.

**ReservaDto**

En este ejemplo se muestra la implementación de un DTO llamado ReservaDto, que se utiliza para transferir datos relacionados con las reservas de un producto por parte de un cliente.

La clase tiene seis atributos, cada uno de los cuales representa un campo de datos que se puede transferir. Los atributos "hora\_Comienzo" y "hora\_Final" son de tipo LocalTime y representan el inicio y fin de la reserva. El atributo "hora\_Comienzo" está anotado con @NotNull, lo que significa que no puede ser nulo, mientras que el atributo "hora\_Final" es opcional y puede ser nulo.

Los atributos "fecha\_Inicial" y "fecha\_Final" son de tipo LocalDate y representan la fecha de inicio y fin de la reserva. Estos atributos también están anotados con @NotNull, lo que significa que no pueden ser nulos.

Los atributos "id\_Producto" y "id\_Cliente" son de tipo Long y representan los identificadores del producto y del cliente que realizó la reserva. Estos atributos también están anotados con @NotNull, lo que significa que no pueden ser nulos.

La clase ReservaDto utiliza las anotaciones @Getter y @Setter para generar automáticamente los métodos getter y setter para cada atributo.

En resumen, la clase ReservaDto es un ejemplo de DTO utilizado para transferir datos relacionados con las reservas de un producto por parte de un cliente. Cada campo de datos en el DTO tiene anotaciones que establecen requisitos de validación y restricciones para asegurar que los datos sean precisos y válidos.

**UsuarioDto**

En este ejemplo se muestra la implementación de un DTO llamado UsuarioDto, que se utiliza para transferir datos relacionados con un usuario.

La clase tiene seis atributos, cada uno de los cuales representa un campo de datos que se puede transferir. Los atributos "nombre", "apellido", "email" y "password" son de tipo String y representan el nombre, apellido, correo electrónico y contraseña del usuario. Cada uno de estos atributos está anotado con varias anotaciones de validación, como @Size, @NotBlank y @Email para asegurarse de que los datos sean precisos y válidos.

Los atributos "rolId" e "id\_Ciudad" son de tipo Long y representan el identificador del rol y la ciudad del usuario. Estos atributos también están anotados con @NotNull, lo que significa que no pueden ser nulos.

La clase UsuarioDto utiliza las anotaciones @Getter y @Setter para generar automáticamente los métodos getter y setter para cada atributo.

En resumen, la clase UsuarioDto es un ejemplo de DTO utilizado para transferir datos relacionados con un usuario. Cada campo de datos en el DTO tiene anotaciones que establecen requisitos de validación y restricciones para asegurar que los datos sean precisos y válidos.