

Paso 1: Identificar el servicio existente

Revisa el servicio que ya tienes en tu proyecto y toma nota de los métodos que incluye.

Paso 2: Crear la interfaz del servicio

Crea una interfaz que defina los métodos que tu servicio debe implementar. Esta interfaz funcionará como un contrato que todas las implementaciones deben cumplir.

Paso 3: Implementar la interfaz en el servicio

Actualiza tu servicio existente para que implemente la interfaz creada. Asegúrate de agregar la anotación `@Service` para que Spring lo registre como un bean. Renombra la clase como `UsuarioServiceImpl` para dejar claro que es una implementación.

Paso 4: Inyectar el servicio en el controlador

Cambia la inyección del servicio en tu controlador para usar la interfaz en lugar de la clase concreta. Usa la anotación `@Autowired` para que Spring se encargue de inyectar la implementación.

Paso 5: Agregar un nuevo método al servicio

Agrega un nuevo método a la interfaz para validar datos provenientes del formulario, y luego implementa este método en la clase que implementa la interfaz. Define las reglas de validación necesarias (por ejemplo, campos obligatorios o validaciones específicas).

Paso 6: Usar el nuevo método en el controlador

Modifica el controlador para que llame al nuevo método de validación cuando procese datos de un formulario. Asegúrate de manejar el resultado de la validación mostrando un mensaje adecuado en la vista.

Paso 7: Implementar la Lógica de Validación

1. En el servicio:

- En la implementación de UsuarioService, define las validaciones:
 - El nombre no puede estar vacío.
 - El correo debe contener un @.
 - La edad debe ser mayor a 0.
- Si las validaciones son exitosas, agrega el usuario.

2. En el controlador:

- Si el servicio retorna false, muestra un mensaje de error en la vista usando Thymeleaf.
- Si las validaciones pasan, redirige a una página de éxito.