

## 1. Crear una aplicación básica con un saludo

- **Descripción:** Crear una aplicación que muestre un saludo en una vista al abrir la página principal.
  - **Instrucciones:**
    - Crear un controlador que maneje una petición GET a /saludo.
    - Pasar un mensaje desde el controlador a la vista usando un Model.
    - Mostrar el mensaje en una vista Thymeleaf.
- 

## 2. Página de cálculo de edad

- **Descripción:** Crear una aplicación que reciba un nombre y una edad desde la URL y muestre un mensaje personalizado.
  - **Instrucciones:**
    - Crear un controlador que capture parámetros desde la URL, como /calcular-edad?nombre=Juan&edad=25.
    - Enviar los datos capturados al modelo.
    - Mostrar un mensaje como: "Hola, Juan. Tu edad es 25 años." en la vista.
- 

## 3. Formulario para sumar dos números

- **Descripción:** Crear un formulario que permita al usuario ingresar dos números y ver la suma.
- **Instrucciones:**
  - Crear un formulario HTML en Thymeleaf que use el método POST.
  - Crear un controlador que reciba los datos del formulario usando @RequestParam.
  - Mostrar la suma en una nueva página.

#### 4. Página para listar usuarios

- **Descripción:** Crear una aplicación que muestre una lista de usuarios en una tabla.
  - **Instrucciones:**
    - Crear un controlador que simule una lista de usuarios (por ejemplo, con una lista de objetos).
    - Pasar la lista al modelo y mostrarla en la vista usando Thymeleaf.
- 

#### 5. Detalle de usuario

- **Descripción:** Mostrar los detalles de un usuario específico al hacer clic en su nombre en la lista del ejercicio 4.
  - **Instrucciones:**
    - Crear un controlador que reciba el ID del usuario en la URL (por ejemplo, /usuario/detalle/1).
    - Pasar los datos del usuario al modelo.
    - Mostrar los detalles del usuario en una nueva vista.
- 

#### 6. Página de búsqueda de usuarios

- **Descripción:** Crea una clase Usuario con dos atributos (username y email). y un método en esa clase que devuelva dos usuarios. Crea un controlador que devuelva una vista con un formulario con un campo de búsqueda donde si se le pasa un nombre de usuario que existe en la vista devuelva todos los datos del usuario (username y email) a la misma vista, caso contrario, un mensajes de “Usuario no encontrado”.

---

## 7. Validación de formularios

- **Descripción:** Crear un formulario que valide que los campos no estén vacíos y que la edad sea un número válido.
- **Instrucciones:**
  - Usar la anotación @Valid en el modelo del formulario.
  - Mostrar mensajes de error en la vista si los datos no son válidos.
  - Muestra los datos solo si la validación es exitosa.

## 8. Mostrar saludo personalizado

Crea una aplicación que reciba el nombre de una persona a través de la URL y muestre un saludo personalizado en una página HTML.

### Requisitos:

1. URL: /saludo?nombre=Juan
2. El controlador debe capturar el parámetro nombre y enviarlo a la vista.
3. En la vista, utiliza Thymeleaf para mostrar el saludo, por ejemplo: "¡Hola, Juan!".

## 9. Detalles del producto

Crea una aplicación que reciba el ID de un producto a través de la URL y muestre los detalles de dicho producto.

### Requisitos:

1. URL: /producto/{id}
2. El controlador debe capturar el parámetro id, buscar un producto simulado (puedes usar una lista o un mapa en memoria), y enviarlo a la vista.
3. La vista debe mostrar el nombre, precio y descripción del producto.

## 10. Cálculo de operaciones matemática

Crea una aplicación que reciba dos números y una operación a través de la URL, realice el cálculo y muestre el resultado en la página.

### Requisitos:

1. URL: /calcular?a=10&b=5&operacion=suma
2. El controlador debe capturar los parámetros a, b y operacion, realizar la operación correspondiente, y enviar el resultado a la vista.
3. En la vista, muestra algo como: "El resultado de la suma es: 15".

### Operaciones soportadas:

- suma
- resta
- multiplicación
- división

## 11. Filtrar lista de usuarios

Crea una aplicación que reciba un parámetro en la URL para filtrar usuarios por nombre o correo.

### Requisitos:

1. URL: /usuarios?filtro=Juan
2. El controlador debe capturar el parámetro filtro, buscar en una lista de usuarios simulada, y enviar los resultados a la vista.
3. En la vista, utiliza Thymeleaf para iterar sobre los resultados y mostrarlos.

## 12. Seguimiento de pedidos

Crea una aplicación que reciba un número de pedido como parte de la URL y muestre su estado. Crea un mapa de pedidos con clave numeroPedido y valor un estado. Crear tres pedidos uno con estado “Enviado” otro “En proceso” y otro “Entegado”

Requisitos:

1. URL: /pedido/{numero}
2. El controlador debe capturar el valor de numero usando @PathVariable, buscar el estado del pedido en un mapa simulado, y enviarlo a la vista.
3. La vista debe mostrar algo como: "El estado de tu pedido 12345 es: Enviado".

## 13. Formulario de registro de usuario

Crea una aplicación que permita registrar usuarios a través de un formulario.

Requisitos:

1. Crea una página con un formulario HTML que capture:
  - Nombre
  - Correo electrónico
  - Contraseña
2. Envía los datos al endpoint POST /registro.
3. El controlador debe procesar los datos y mostrarlos en una página de confirmación.
4. La vista de confirmación debe mostrar: "Usuario [nombre] registrado con el correo [correo]".

## **14. Cálculo de suma**

Crea una aplicación que reciba dos números a través de un formulario y muestre su suma.

### **Requisitos:**

1. Crea una página con un formulario que capture:
  - Número 1
  - Número 2
2. Envía los datos al endpoint POST /sumar.
3. El controlador debe calcular la suma y enviarla a la vista.
4. La vista debe mostrar algo como: "La suma de 5 y 10 es 15".

## **17.Publicar comentarios**

Crea una aplicación que permita a los usuarios publicar comentarios y mostrarlos en una lista.

### **Requisitos:**

1. Crea una página con un formulario que capture:
  - Nombre del usuario
  - Texto del comentario
2. Envía los datos al endpoint POST /comentario.
3. El controlador debe guardar los comentarios en una lista simulada en memoria.
4. En la misma página, muestra la lista de comentarios ya publicados.

## 18: Conversión de temperatura

Crea una aplicación que permita convertir temperaturas de Celsius a Fahrenheit.

Requisitos:

1. Crea una página con un formulario que capture:
  - Temperatura en Celsius
2. Envía los datos al endpoint POST /convertir.
3. El controlador debe convertir la temperatura usando la fórmula:  $F = C * 9/5 + 32$  y enviar el resultado a la vista.
4. La vista debe mostrar algo como: "30°C equivalen a 86°F".

## 19: Encuesta de satisfacción

Crea una aplicación que permita a los usuarios responder una encuesta de satisfacción.

Requisitos:

1. Crea una página con un formulario que capture:
  - Nombre del usuario
  - Pregunta: "¿Qué te pareció el servicio?" (usa botones de opción: Excelente, Bueno, Regular, Malo)
2. Envía los datos al endpoint POST /encuesta.
3. El controlador debe mostrar un mensaje de agradecimiento y el resumen de la respuesta en la vista.

## 20: Búsqueda de productos

Crea una aplicación que permita buscar productos por nombre.

Requisitos:

1. Crea una página con un formulario que capture:
  - Texto para buscar el producto

2. Envía los datos al endpoint POST /buscar.
3. El controlador debe buscar los productos que coincidan con el texto en una lista simulada en memoria.
4. La vista debe mostrar los resultados de la búsqueda en una tabla.

## **21: Login de usuario**

Crea una aplicación que permita a los usuarios iniciar sesión.

### **Requisitos:**

1. Crea una página con un formulario que capture:
  - Correo electrónico
  - Contraseña
2. Envía los datos al endpoint POST /login.
3. El controlador debe validar los datos (puedes usar un mapa en memoria con correos y contraseñas simulados).
4. Si los datos son correctos, muestra: "Bienvenido [nombre]".
5. Si los datos son incorrectos, muestra un mensaje de error en la misma página.

## **22: Formulario de contacto**

Crea una aplicación que permita a los usuarios enviar mensajes a través de un formulario de contacto.

### **Requisitos:**

1. Crea una página con un formulario que capture:
  - Nombre
  - Correo electrónico
  - Mensaje
2. Envía los datos al endpoint POST /contacto.



3. El controlador debe guardar los datos en una lista simulada en memoria y mostrar un mensaje de confirmación: "Gracias por contactarnos, [nombre]. Pronto responderemos tu mensaje".

### **23: Calculadora de IMC**

Crea una aplicación que permita a los usuarios calcular su Índice de Masa Corporal (IMC).

#### **Requisitos:**

1. Crea una página con un formulario que capture:
  - Peso en kilogramos
  - Altura en metros
2. Envía los datos al endpoint POST /imc.
3. El controlador debe calcular el IMC usando la fórmula:  $IMC = peso / (altura * altura)$  y clasificarlo (Bajo peso, Normal, Sobrepeso, Obesidad).
4. La vista debe mostrar el resultado del cálculo y la clasificación.

### **24. Registro de tareas**

Crea una aplicación que permita a los usuarios registrar tareas y marcarlas como completadas.

#### **Requisitos:**

1. Crea una página con un formulario que capture:

- Descripción de la tarea
2. Envía los datos al endpoint POST /tareas.
  3. El controlador debe agregar la tarea a una lista en memoria y mostrarla en una tabla en la misma página.
  4. Agrega un botón para marcar una tarea como completada (esto puede redirigir a otro endpoint POST o DELETE).

## **25. Selección de país y ciudad**

Crea una aplicación que permita seleccionar un país y, basado en esa selección, elegir una ciudad.

### **Requisitos:**

1. Crea una página con un formulario que contenga:
  - Un desplegable para seleccionar el país.
  - Un segundo desplegable que se actualice dinámicamente con las ciudades disponibles para el país seleccionado (puedes simular esto con datos predefinidos en el backend).
2. Envía los datos al endpoint POST /seleccion.
3. Muestra en la vista algo como: "Has seleccionado el país [país] y la ciudad [ciudad]".

## **26. Selección de categoría de producto**

Crea una aplicación que permita seleccionar una categoría de producto y muestre los productos de esa categoría.

**Requisitos:**

1. Crea un formulario con un desplegable para seleccionar la categoría.
2. Envía la selección al endpoint POST /categoria.
3. En la vista de resultados, muestra una lista de productos disponibles en esa categoría.

**28: Encuesta de preferencia**

Crea una encuesta que permita seleccionar una opción entre varias categorías.

**Requisitos:**

1. Crea un formulario con un desplegable que contenga opciones como:
  - "Cine"
  - "Deportes"
  - "Tecnología"
2. Envía la selección al endpoint POST /encuesta.
3. Muestra en la vista algo como: "Gracias por participar. Tu categoría favorita es: [categoría]".

**27: Selección de hobbies**

Crea una aplicación que permita a los usuarios seleccionar sus hobbies favoritos.

**Requisitos:**

1. Crea un formulario que incluya varios hobbies con checkboxes:
  - "Leer"
  - "Deportes"
  - "Viajar"
  - "Cocinar"
2. Envía los datos al endpoint POST /hobbies.
3. El controlador debe procesar los datos seleccionados y mostrarlos en una vista.
4. Muestra en la vista algo como: "Tus hobbies seleccionados son: [lista de hobbies]".

## **29: Suscripción a boletines**

Crea una aplicación que permita a los usuarios suscribirse a varios boletines.

### **Requisitos:**

1. Crea un formulario con opciones de checkboxes como:
  - "Noticias"
  - "Promociones"
  - "Eventos"
2. Envía los datos al endpoint POST /suscripcion.
3. El controlador debe procesar las opciones seleccionadas y mostrar un mensaje como: "Te has suscrito a: [lista de boletines]".

## **30: Selección de ingredientes para una pizza**

Crea una aplicación que permita a los usuarios seleccionar los ingredientes de una pizza.

### **Requisitos:**

1. Crea un formulario con checkboxes para ingredientes como:
  - "Queso"

- "Pepperoni"
  - "Champiñones"
  - "Jalapeños"
2. Envía los datos al endpoint POST /pizza.
  3. Muestra en la vista algo como: "Tu pizza tendrá los siguientes ingredientes: [lista de ingredientes]".

### **32: Selección de método de pago**

Crea una aplicación que permita a los usuarios seleccionar su método de pago preferido.

#### **Requisitos:**

1. Crea un formulario con radio buttons para seleccionar:
  - "Tarjeta de crédito"
  - "PayPal"
  - "Transferencia bancaria"
2. Envía los datos al endpoint POST /pago.
3. Muestra en la vista: "Has seleccionado pagar con: [método de pago]"

### **31: Selección de tipo de usuario**

Crea una aplicación que permita a los usuarios seleccionar su tipo de perfil.

#### **Requisitos:**

1. Crea un formulario con radio buttons para elegir:

- "Usuario individual"
  - "Empresa"
2. Envía los datos al endpoint POST /tipoUsuario.
  3. Muestra en la vista algo como: "Has seleccionado el perfil: [tipo de usuario]".

### 33: Registro de usuario con validación

Crea un formulario para registrar un usuario con los siguientes campos:

1. **Nombre** (obligatorio, al menos 3 caracteres).
2. **Correo electrónico** (obligatorio, formato válido).
3. **Contraseña** (obligatorio, al menos 8 caracteres).

#### Requisitos:

1. Implementa la validación utilizando anotaciones de Bean Validation, como @NotBlank, @Email, y @Size.
2. Si hay errores, muéstralos en el formulario con Thymeleaf.
3. Procesa los datos si son válidos y muestra un mensaje de éxito.

### 34: Encuesta de opinión

Crea un formulario donde los usuarios puedan responder a una encuesta con los siguientes campos:

1. **Edad** (obligatorio, debe ser un número entre 18 y 100).

2. **Calificación del servicio** (radio buttons: "Excelente", "Bueno", "Regular", "Malo", obligatorio).
3. **Comentarios adicionales** (opcional, máximo 200 caracteres).

**Requisitos:**

1. Valida que la edad esté dentro del rango permitido.
2. Si faltan datos obligatorios, muestra un mensaje de error.
3. Si los datos son válidos, procesa la encuesta y muestra un mensaje de agradecimiento.