

TAREA02

Cruz Vargas Uziel Vidal

Noriega Rodríguez Marcos

Julian



Actividad

- Programar una aplicación web que muestre el siguiente formulario mediante una vista y devuelva otra mostrando los datos capturados, utilizando el framework de Spring y almacenando los datos capturados en una base de datos, por ejemplo MySQL. Ver el programa de ejemplo visto en clase FormularioDBSpring3MVC.

Datos del crédito

Fecha <input type="text" value="07/10/2025"/>	Moneda <input checked="" type="radio"/> Soles (S/.) <input type="radio"/> Dólares (US\$)	Monto <input type="text" value="0.01"/>
Periodo (meses) <input type="text" value="1"/>	Cuota <input type="text" value="0.01"/>	TEA (%) <input type="text" value="0.02"/>
Fecha de vencimiento <input type="text" value="21/10/2025"/>		
<input type="button" value="ACEPTAR"/>		

Figura 1. Formulario

Campos capturados

Campo	Valor
ID	33
Fecha	2025-10-07
Moneda	Soles (S/.)
Monto	0.01
Periodo (meses)	1
Cuota	0.01
TEA (%)	0.02
Fecha vencimiento	2025-10-21

[← Capturar de nuevo](#) | [Ver todos](#)

Figura 2. Resultado

Créditos almacenados

ID	Fecha	Moneda	Monto	Periodo
1	2025-10-07	Soles (S/.)	0.01	1
33	2025-10-07	Soles (S/.)	0.01	1

[← Volver](#)

Figura 3. Datos capturados en la BD.

Explicación

Esta tarea desarrolló la misma aplicación de capturar datos de crédito, pero utilizando el framework SpringBoot. Este marco de trabajo nos simplifica la creación de aplicaciones web en Java al integrar los componentes de servidor embebido, la gestión de dependencias, el enrutamiento y la persistencia de datos.

El flujo es el siguiente:

1. El usuario accede al formulario en index.html, desarrollado con Thymeleaf y completa los campos de crédito.
2. Al presionar Aceptar se envía la información mediante el método POST al controlador CreditoController.
3. El controlador recibe los datos a través de la notación @ModelAttribute, los encapsula en un objeto Crédito y los guarda en la base de datos mediante Spring Data JPA.
4. Una vez almacenado, el controlador reenvía al usuario hacia la vista resultado.html donde se muestran los datos capturados y confirmados desde la base de datos.

Así podemos implementar el proyecto con el patrón MVC. Spring Boot incorpora su propio servidor embebido (Tomcat interno), por lo que no fue necesario configurarlo manualmente, y además gestiona la base de datos H2 sin dependencias externas.

Dificultades

- La configuración inicial: buscar plugins para los diferentes IDEs que quisimos probar fue un reto divertido donde tuvimos que leer documentaciones, proyectos de ejemplo y tutoriales. Sin ayuda de algún wizard, buscar dependencias fue complicado hasta que pudimos agregarlas manualmente al pom.xml
- Hablando del pom.xml, tuvimos que buscar que las versiones de las dependencias fueras compatibles entre sí.

Después de resolver estas configuraciones, el proyecto pudo ejecutarse correctamente, mostrando el formulario, guardando la información en la base de datos y confirmando los datos en la vista de resultados.