

# TAREA01

## Cruz Vargas Uziel Vidal



## Actividad

- Programar un Servlet que muestre la siguiente vista y devuelva otra vista mostrando los campos capturados. Ver el programa de ejemplo visto en clase Forma\_de\_Compra.

### Datos del crédito

Fecha

05/02/2003



Moneda

☐ Soles (S/.)

☒ Dólares (US\$)

Monto

0.03

Periodo (meses)

1

Cuota

0.01

TEA (%)

0.01

Fecha de vencimiento

19/09/2025



ACEPTAR

Figura 1. Formulario

## Campos capturados

Campo	Valor
Fecha	2003-02-05
Moneda	Dólares (US\$)
Monto	0.03
Periodo	1
Cuota	0.01
TEA (%)	0.01
Fecha vencimiento	2025-09-19

[← Capturar de nuevo](#)

Figura 2. Resultado

## Explicación

Un **Servlet** es un componente de Java que se ejecuta en el servidor de aplicaciones (en este caso Tomcat) y permite procesar peticiones HTTP (GET, POST, etc.).

En este proyecto se utiliza el servlet Capturar, que está anotado con `@WebServlet` para quedar accesible en la URL `/Capturar`.

1. El usuario llena el formulario en `index.jsp` y envía la información con método **POST** al servlet.
2. En el servlet, el método `doPost(HttpServletRequest request, HttpServletResponse response)` se ejecuta y recoge los parámetros enviados con `request.getParameter("nombreCampo")`.
3. Esos valores se guardan en un objeto tipo `Map` y se colocan como atributo en la petición (`request.setAttribute("datos", datos);`).
4. Finalmente, el servlet usa un **RequestDispatcher** para redirigir la ejecución hacia `resultado.jsp`

# Dificultades

---

Tuve problemas al principio con la instalación del servidor TomCat 9.0. Al correr mi proyecto me pedía credenciales que no tenía, por lo que tuve que editar en `tomcat-users.xml` los usuarios con roles `manager-gui` y `manager-script`.

También tuve BUILD FAILURE porque NetBeans incluía la versión obsoleta del plugin `maven-war-plugin`. Actualicé el `pom.xml` para la versión 3.3.2 y añadí la dependencia correcta `javax.servlet.api`.