

# Laboratorio 2: Applets y Servlets

Sistemas Distribuidos  
EU Informática Segovia  
Universidad de Valladolid

Mario Villacorta García

# Índice global

## 1. Applets

1.1. Taxonomía de clases

1.2. Ciclo de vida

1.3. Limitaciones

1.4. Implementación

## 2. Servlets

2.1. Servlets y JSPs

2.2. Taxonomía de clases

2.3. Ciclo de vida

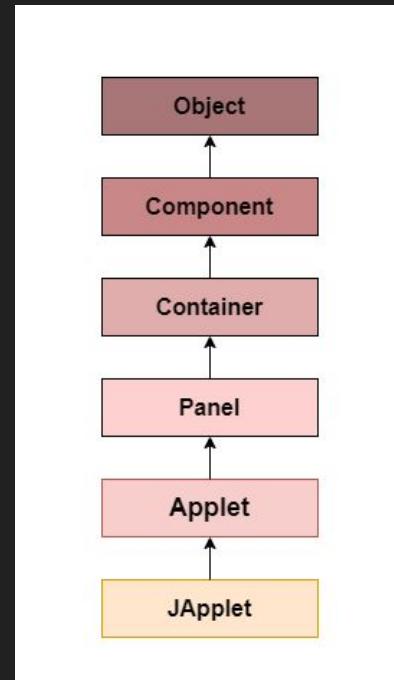
2.4. Implementación

Sistemas Distribuidos  
EU Informática Segovia  
Universidad de Valladolid

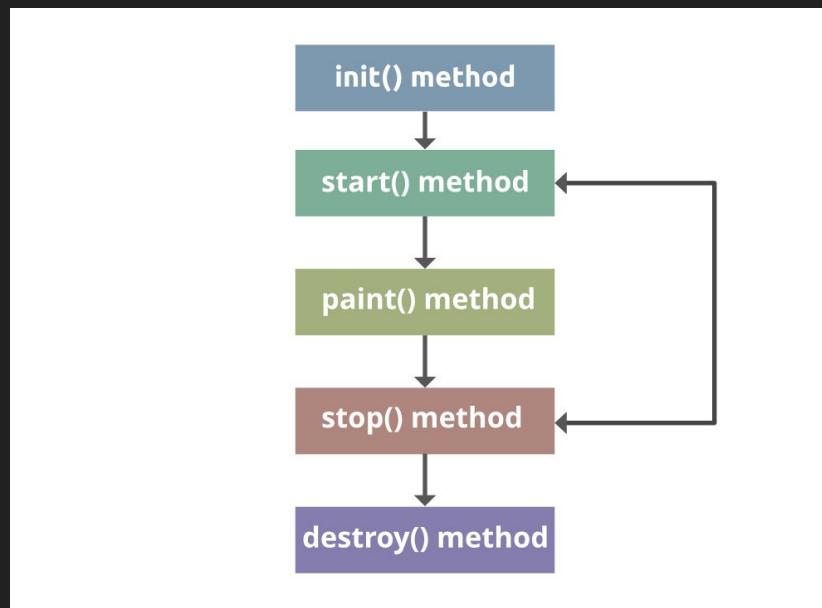
# 1. Applets

- **Applet:** aplicación Java que permite ejecutar código embedido dentro de una página web en el lado del cliente.
- Procedimiento
  - La llamada a un applet se inserta en HTML con la etiqueta `<applet>`.
  - El navegador detecta la llamada, descarga el código del applet del servidor.
  - La máquina virtual de Java (JVM) del navegador lanza su ejecución.
- Ejemplo de **código móvil**.
- Actualmente están en desuso por cuestiones de seguridad.

## 1.1. Taxonomía de clases



## 1.2. Ciclo de vida



## 1.2. Ciclo de vida

- ***public void init()***
  - Los applets no tienen método *main*. Comienzan su ejecución en *init*.
  - Sirve para inicializar los elementos que va a utilizar (variables, hilos, recuperar parámetros pasados inicialmente...).
- ***public void start()***
  - Ejecutado cuando el navegador muestra la página que está insertando.
- ***public void paint (Graphics g)***
  - Muestra el contenido visual del applet.
- ***public void stop()***
  - Para el applet. Se ejecuta cuando el navegador se minimiza o el usuario cambia de página.
- ***public void destroy()***
  - Elimina de memoria los objetos creados durante la ejecución. Cuando el navegador se cierra.

## 1.3. Limitaciones

- Applets **cargados a través de la red**
  - No puede **leer o escribir** ficheros en el cliente.
  - No puede hacer **conexiones de red**, excepto con el servidor origen.
  - No puede **arrancar ningún programa** en el cliente.
- Applets **locales**: sin limitaciones.

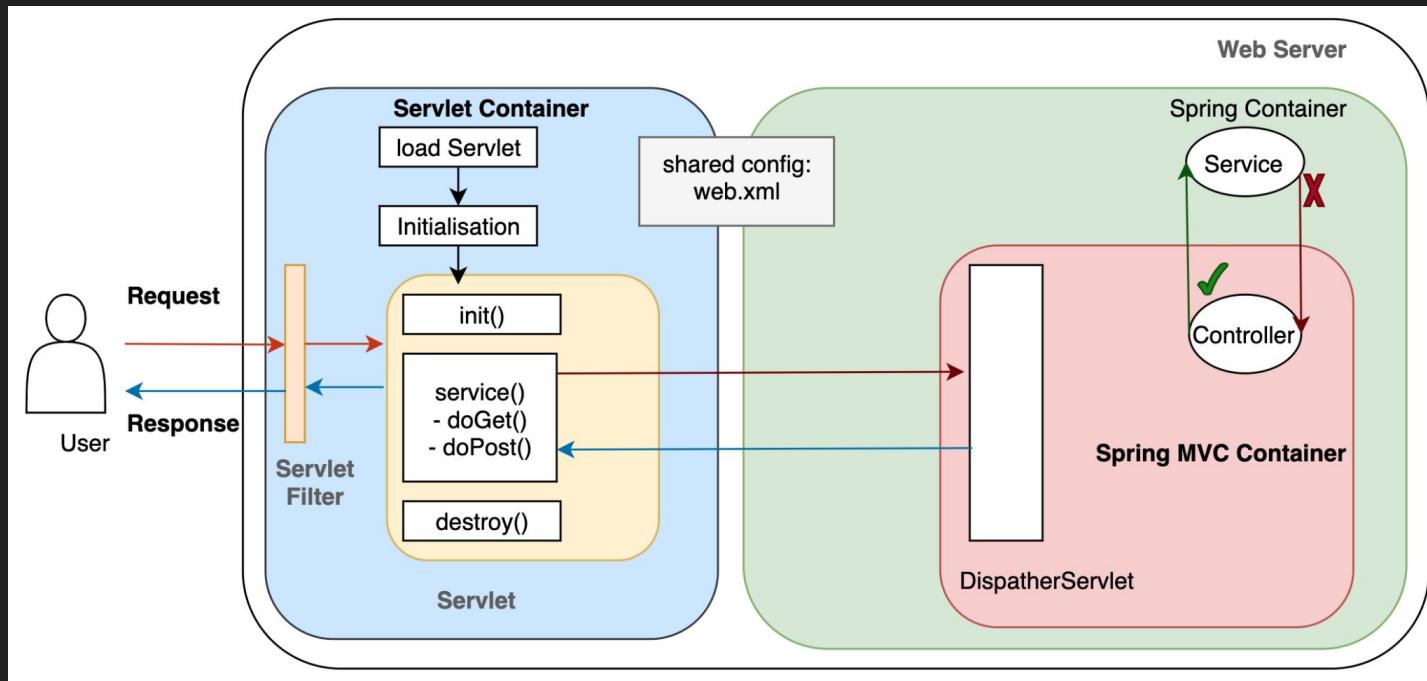
## 1.4. Implementación

- *import javax.swing.\**
- Extender de *JApplet* e implementar los métodos de ciclo de vida.
- Crear elementos de Java Swing y añadir al applet mediante método *add*.
- Utilizamos *appletviewer* para ejecutar.

## 2. Servlets

- **Servlet:** aplicación Java que se ejecuta en el lado del servidor para dar respuesta a una petición HTTP.
- Procedimiento
  - Servidor instancia *servlet* y espera peticiones HTTP del cliente.
  - Cliente emite petición HTTP con **parámetros** necesarios.
  - Servidor ejecuta *servlet* y genera respuesta a petición en forma de **datos o contenido dinámico**.
- Sustituyen a *Common Gateway Interface* (CGI).
- Estándar de facto dentro del mundo Java para interacción HTTP en modelo petición-respuesta.

## 2. Servlets



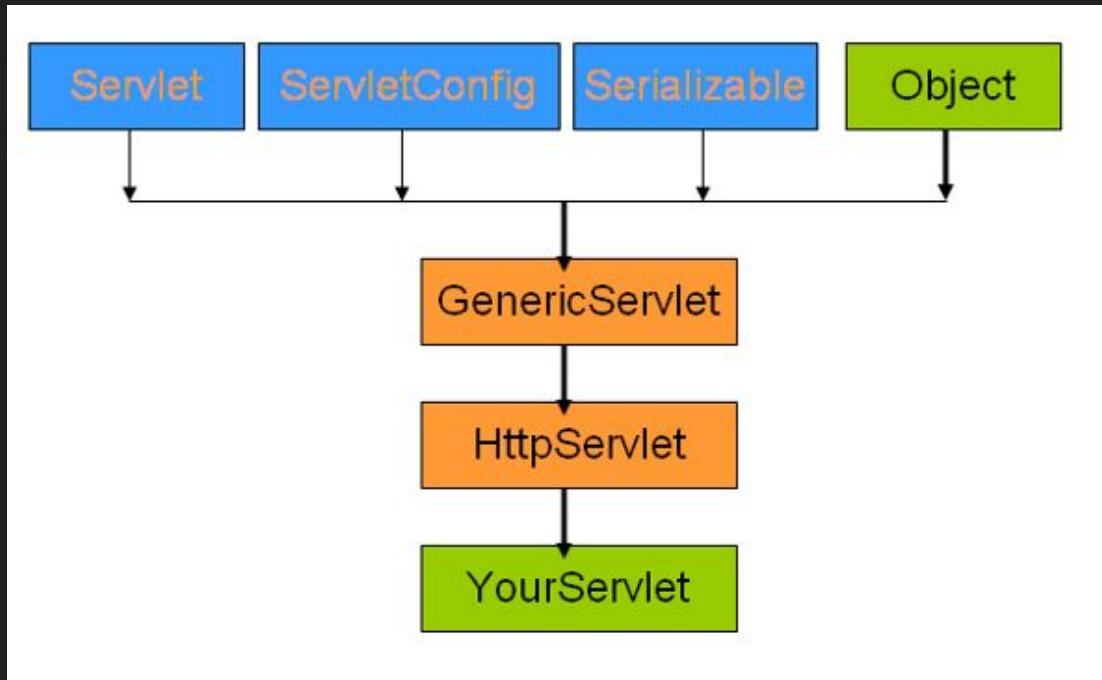
## 2.1. Servlets y JSPs

- **JSP:** JavaServer Pages, tecnología de creación de web apps de forma dinámica desde el lado del servidor. Trabaja sobre servlets
- Procedimiento
  - Definición de una JSP (archivo con extensión .jsp) que es HTML con código Java embedido.
  - Cliente realiza petición de JSP al servidor.
  - Servidor compila JSP y devuelve una página HTML con el contenido creado de forma dinámica.
- Suele trabajar de forma conjunta con *Servlets* en los proyectos.

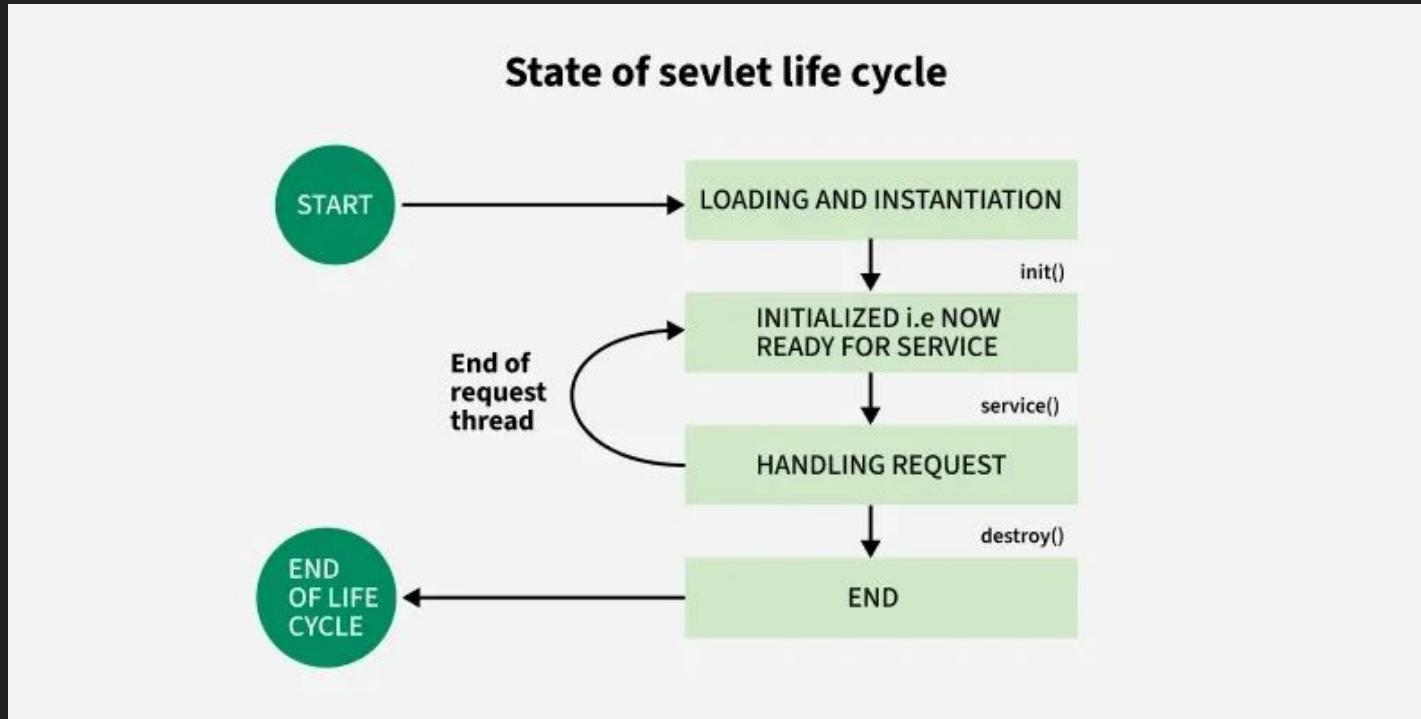
## 2.1. Servlets y JSPs

Features	JSP	Servlet
<b>Code Length</b>	Jsp required less code	Servlets required more code
<b>Ease of Use</b>	Jsp is simple to use	Servlet is more complex to use
<b>Dynamic Content</b>	Easily embedded in HTML	Requires HTML generation in code
<b>Page Maintenance</b>	Easier to maintain	More challenging

## 2.2. Taxonomía de clases



## 2.3. Ciclo de vida



## 2.3. Ciclo de vida

- ***void init(ServletConfig config) throws ServletException***
  - Inicialización del servlet con su setup, variables, conexión a BBDD, etc.
- ***void service(ServletRequest req, ServletResponse res) throws ServletException, IOException***
  - Manejo de la petición HTTP.
  - Definición del método HTTP utilizado (GET, POST, PUT, DELETE).
- ***public void destroy()***
  - Limpieza del servlet.

## 2.4. Implementación

- Elegir tecnología de contenedor de servlets: Glassfish o **Apache Tomcat**.
- *import jakarta.servlet.\** si Jakarta EE 9 o *javax.servlet.\**
- Crear clase que herede de *HTTPServlet*.
- Definir formato de respuesta (HTML, JSON, etc.) mediante *content-type*.
- Implementar métodos de manejo de peticiones HTTP.

## 2.4. Implementación

Method	Description
<code>void doDelete(HttpServletRequest req,                   HttpServletResponse res) throws IOException, ServletException</code>	Handles an HTTP DELETE request.
<code>void doGet(HttpServletRequest req,               HttpServletResponse res) throws IOException, ServletException</code>	Handles an HTTP GET request.
<code>void doHead(HttpServletRequest req,               HttpServletResponse res) throws IOException,           ServletException</code>	Handles an HTTP HEAD request.
<code>void doOptions(HttpServletRequest req,               HttpServletResponse res) throws IOException, ServletException</code>	Handles an HTTP OPTIONS request.
<code>void doPost(HttpServletRequest req,               HttpServletResponse res) throws IOException, ServletException</code>	Handles an HTTP POST request.
<code>void doPut(HttpServletRequest req,               HttpServletResponse res) throws IOException, ServletException</code>	Handles an HTTP PUT request.
<code>void doTrace(HttpServletRequest req,               HttpServletResponse res) throws IOException, ServletException</code>	Handles an HTTP TRACE request.
<code>long getLastModified(HttpServletRequest req)</code>	Returns the time (in milliseconds since midnight, January 1, 1970, GMT) when the requested resource was last modified.
<code>void service(HttpServletRequest req,               HttpServletResponse res) throws IOException, ServletException</code>	Called by the server when an HTTP request arrives for this servlet. The arguments provide access to the HTTP request and response, respectively.

# Bibliografía y referencias

Groussard, T. (2003). Los applets. En Java 7: Los fundamentos del lenguaje Java (Cap. 2). Eni.

GeeksforGeeks. (2025, July 23). Life cycle of Java applet. GeeksforGeeks. Retrieved October 31, 2025, from <https://www.geeksforgeeks.org/java/life-cycle-of-java-applet/>

Schildt, H. (2003). The History and Evolution of Java. En Java: The Complete Reference Eleventh Edition (Cap. 1). Mc Graw Hill.

Schildt, H. (2003). Introducing Servlets. En Java: The Complete Reference Eleventh Edition (Cap. 35). Mc Graw Hill.

javaskool.com. (2015, Jan 19). Servlet Programming [ A Dynamic web Component ]. javaskool.com. Retrieved October 31, 2025, from <https://www.javaskool.com/servlet-programming/>

GeeksforGeeks. (2025, Aug 01). Introduction to JSP. GeeksforGeeks. Retrieved October 31, 2025, from <https://www.geeksforgeeks.org/advance-java/introduction-to-jsp/>

GeeksforGeeks. (2025, Oct 03). Life Cycle of a Servlet. GeeksforGeeks. Retrieved October 31, 2025, from <https://www.geeksforgeeks.org/java/life-cycle-of-a-servlet/>