

# Despliegue de aplicaciones en el servidor de aplicaciones

Servlets

# Servlets

Un servlet amplía las funcionalidades de un servidor ofreciendo un **servicio específico** dentro de un marco de trabajo bien definido (contexto o aplicación).

Es un **pequeño código Java** (normalmente una sola clase) que proporciona un servicio específico.

Cuando el servidor recibe un tipo particular de solicitud, invoca el servlet asociado, le pasa los detalles de la petición y un objeto *response* para devolver el **resultado**.




# Servlets

Todos los servlets implementan la interfaz ***javax.servlet.Servlet*** directa o indirectamente (servlets HTTP o JSP). Esta interfaz incluye los siguientes métodos:

**init()**: define cualquier **código de inicialización** que debería ejecutar cuando se carga el servlet en memoria (cuando se invoca).

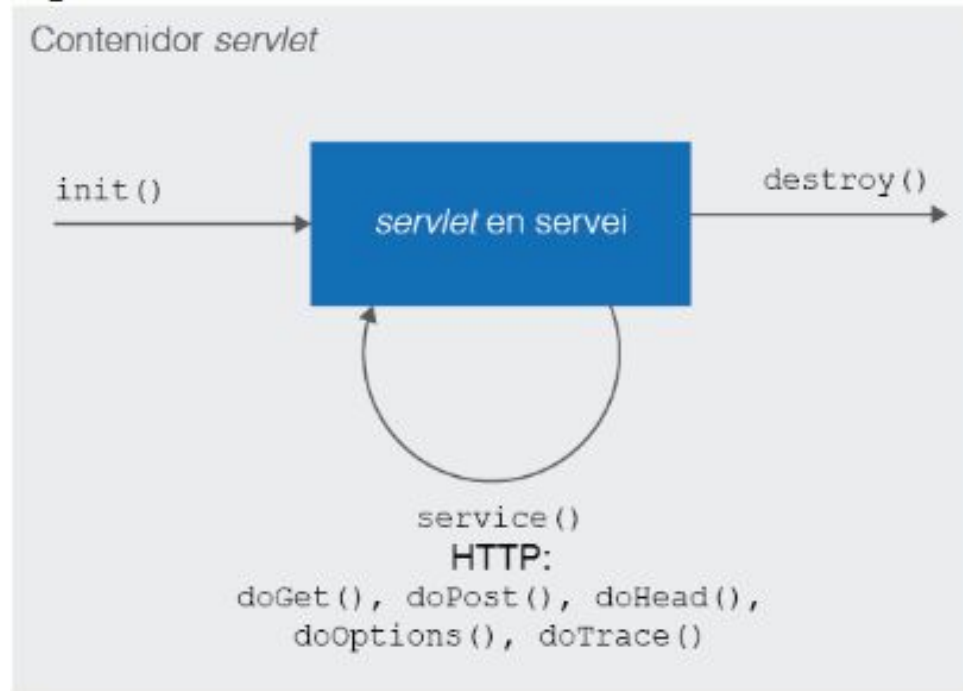
**service()**: es el método principal, llamado cuando el servlet recibe una solicitud de servicio. Contiene la **lógica de procesamiento** (el código a ejecutar).

**destroy()**: define cualquier **código de limpieza** requerido antes de eliminar el servlet de la memoria (unload, no significa que se elimine del servidor).



# Servlets - Ciclo de vida

Ciclo de vida de un servlets cargado y en ejecución

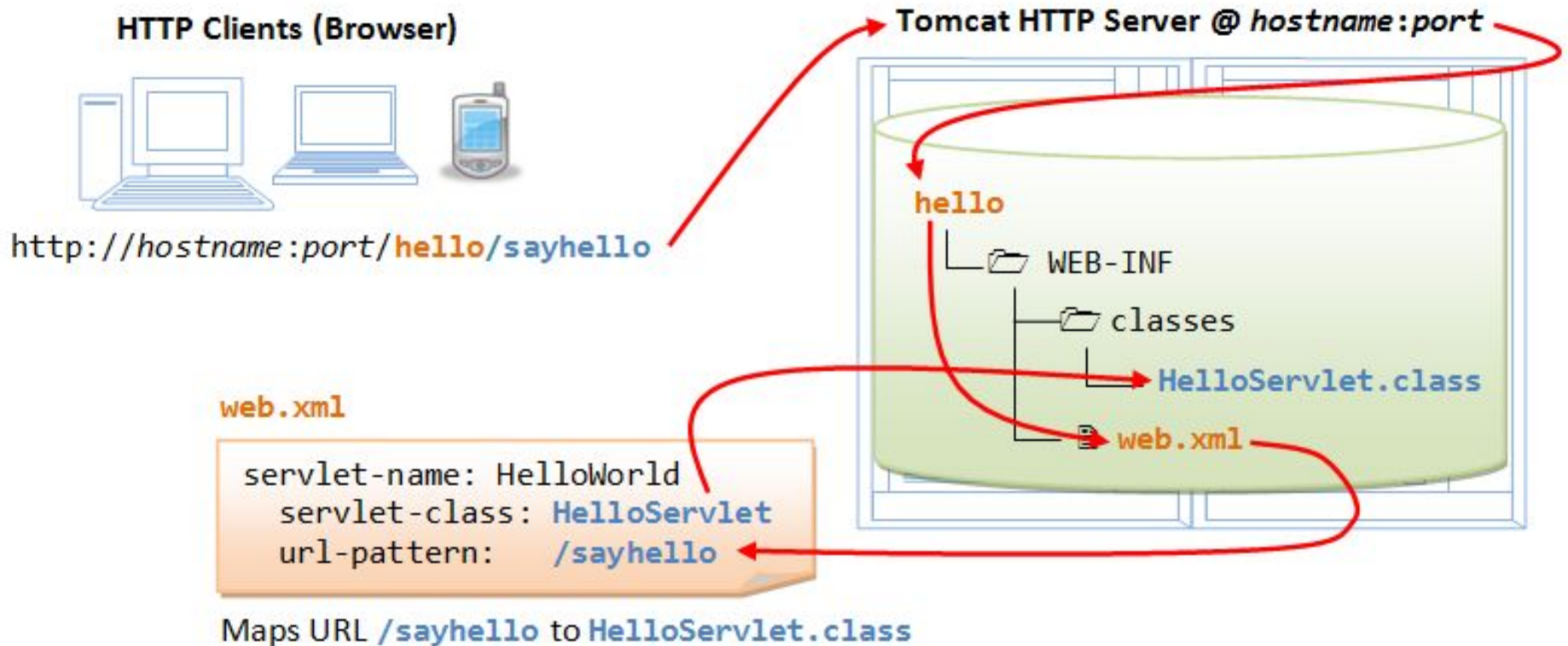


# Servlets - Ciclo de vida

Durante el tiempo de vida de un servlet, los métodos *init()* y *destroy()* sólo son invocados **una vez**, mientras que el método *service()* es invocado **muchas veces** (cada vez que se haga una solicitud para ejecutar el applet).



# Servlets - Proceso de Ejecución



# Servlets - Creación

**Recordatorio:** imprescindible tener la biblioteca de JDK instalada y configurada.

Definir una clase que herede de **HttpServlet** (implementamos la interfaz ***javax.servlet.Servlet*** para HTTP).

Sobrescribir los métodos “**doGet**” o “**doPost**” para manejar solicitudes.



# Servlets - Creación - Estructura

Generar la siguiente estructura de carpetas y ficheros:

HolaMundo/

└─ WEB-INF/

│ └─ classes

│ └─ web.xml

└─ HolaMundoServlet.java

Carpeta raíz

Carpeta privada

Carpeta (aquí irán las clases compiladas)

Fichero (descriptor de despliegue)

Fichero con código java (mover una vez compilado)





# Servlets - Creación - HolaMundoServlet.java

```
import java.io.IOException;  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;
```



# Servlets - Creación - HolaMundoServlet.java

```
@WebServlet("/hola")

public class HolaMundoServlet extends HttpServlet {

    private static final long serialVersionUID = 1L;

    @Override

    protected void doGet(HttpServletRequest request, HttpServletResponse response)

        throws ServletException, IOException {

        response.setContentType("text/html");

        response.getWriter().println("<h1>¡Hola Mundo!</h1>");

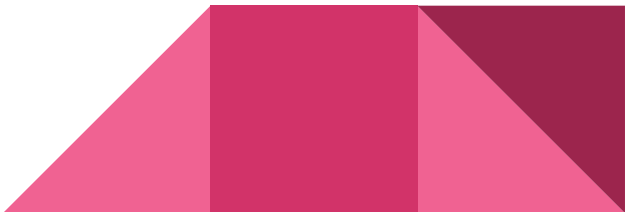
    }

}
```



# Servlets - Creación - web.xml

```
<?xml version="1.0" encoding="UTF-8"?>  
<web-app xmlns="http://java.sun.com/xml/ns/javaee" version="3.0">  
  <servlet>  
    <servlet-name>HolaMundoServlet</servlet-name>  
    <servlet-class>HolaMundoServlet</servlet-class>  
  </servlet>  
  <servlet-mapping>  
    <servlet-name>HolaMundoServlet</servlet-name>  
    <url-pattern>/hola</url-pattern>  
  </servlet-mapping>  
</web-app>
```



# Servlets - Compilación

Compilar mediante javac:

```
javac -d HolaMundo/WEB-INF/classes HolaMundoServlet.java
```

Esto generará el archivo **.class** dentro de **WEB-INF/classes**



# Servlets - Empaquetado

Se empaqueta toda la estructura en un archivo **.war** usando **jar**:

```
jar -cvf HolaMundo.war -C HolaMundo .
```


(HolaMundo siendo la carpeta raiz del proyecto)

Esto creará el archivo HolaMundo.war, que contiene toda la estructura necesaria para poder ser desplegado en Tomcat.



# Compilación VS Empaquetado: javac VS jar

Los comandos javac y jar son herramientas de línea de comandos del **JDK** (Java Development Kit) que cumplen funciones distintas en el ciclo de desarrollo de aplicaciones Java.

- **javac** (Java Compiler): Se usa para **compilar** archivos de código fuente Java (.java) en archivos de bytecode Java (.class). Este paso traduce el código humano legible a un **formato entendible para la Máquina Virtual de Java (JVM)**.
  - **jar** (Java Archive Tool): Se usa para **empaquetar** múltiples archivos (como .class, recursos como imágenes o configuraciones) en un archivo comprimido .jar o .war. Estos archivos son usados para desplegar aplicaciones de Java.
- 

# Servlets - Despliegue

Dos formas para desplegar:

- **Copiar el archivo** HolaMundo.war **al directorio webapps** de Tomcat en el servidor. Tomcat desplegará automáticamente el archivo (recomendable reiniciar).
- **Subir mediante manager-app** (aplicación que viene por defecto) el archivo HolaMundo.war, siguiendo los pasos indicados.

Si todo el proceso ha ido bien, desde la URL: <http://localhost:8080/HolaMundo/hola>

Se mostrará la página creada con el mensaje “¡Hola Mundo!”

