

Tecnologías en el Servidor

Java - Servlets

Curso 2020/21

Javier Albert Seguí

Indice

Arquitectura Web - Recordatorio

- La Arquitectura de las aplicaciones Web suelen presentar un esquema de tres niveles:
 - El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor web que es el responsable de dar a los datos un formato adecuado.
 - El segundo nivel está referido habitualmente a algún tipo de programa o script – lógica de negocio.
 - Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución.

Java EE

- Java Platform Enterprise Edition (antes J2EE, ahora Jakarta EE) es un estándar para el desarrollo de aplicaciones empresariales (portables, robustas, escalables y seguras) usando tecnología Java.
- Jakarta EE es un conjunto de especificaciones, **no es un producto**. Los productos que cumplen con la especificación son realizados por terceras empresas u organizaciones.
- Toda la información acerca de esta especificación la podemos encontrar en <https://eclipse-ee4j.github.io/jakartaee-platform/>

Jakarta EE

- Ahora la versión de esta especificación es la v8 publicada en septiembre/17
- Incluye:
 - Jakarta WebSocket
 - Jakarta Servlet
 - Jakarta Server Page
 - Jakarta JSON Processing
 - Jakarta Standard Tag Library
 - Jakarta Server Faces
 - Jakarta RESTful Web Services

Tecnologías Java

- En JAVA tenemos dos tecnologías distintas para el Desarrollo de aplicaciones en un entorno Web:
 - Servlets
 - JSP
- Aunque distintas, están íntimamente ligadas, dado que la tecnología JSP al final se transforma en un servlet que son las aplicaciones que se ejecutan sobre el servidor de aplicaciones.

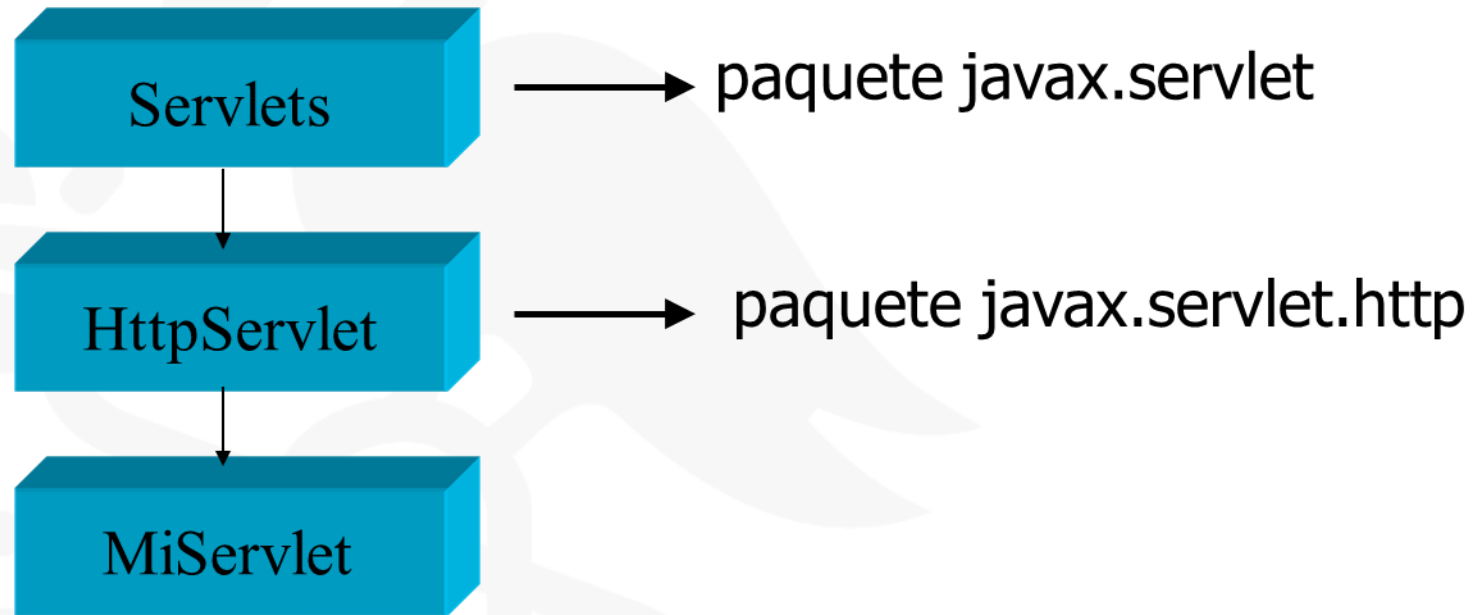
Servlets

- Se puede definir un **Servlet** como un programa JAVA que se ejecuta en un entorno distribuido en red, como un servidor web, y que recibe y responde a las peticiones de un cliente a través del protocolo HTTP.
- Son la contraparte a tecnologías como ASP.net y PHP
- Los **Servlets** son un reemplazo efectivo para los CGI en los servidores que los soporten ya que proporcionan una forma de generar documentos dinámicos utilizando las ventajas de la programación en Java como conexión a base de datos, manejo de peticiones concurrentes, programación distribuida, etc.
 - Por ejemplo, un servlet podría ser responsable de procesar los datos desde un formulario en HTML como registrar la transacción, actualizar una base de datos, contactar algún sistema remoto y retornar un documento dinámico o redirigir a otro servlet u alguna otra cosa.

Servlets

- Los servlets son independientes del servidor y de la Plataforma en que se ejecuta.
- En el servidor tendremos que tener una maquina virtual JAVA, además de las librerías necesarias para el soporte de los servlets.

Servlets - API



Servlets - API

- Hay dos paquetes dentro del API de Servlets

- javax.servlet
- javax.servlet.http

- **javax.servlet**

- Interfaz Servlet
- Interfaz ServletRequest
- Interfaz ServletResponse
- Interfaz ServletConfig
- Interfaz ServletContext
- Interfaz SingleThreadModel
- Clase GenericServlet

javax.servlet.http

Interfaz HttpServlet

Interfaz HttpServletRequest

Interfaz HttpServletResponse

Interfaz HttpServletConfig

Servlets - API

○ Clase HttpServlet

- Extiende de la clase GenericServlet y proporciona una implementación de la interfaz Servlet mucho más específica para el protocolo HTTP.
Esta es la clase que extienden la mayoría de los servlets que hay en la actualidad.
- Métodos (throws ServletException, IOException)
 - Gestionan el servicio
 - GET/ POST service (HttpServletRequest req, HttpServletResponse res)
 - GET/ POST processRequest (HttpServletRequest req, HttpServletResponse res) → NetBeans
 - Implementan operaciones propias de HTTP
 - GET doGet (HttpServletRequest req, HttpServletResponse res)
 - POST doPost (HttpServletRequest req, HttpServletResponse res)

Servlet - API

○ `javax.servlet.http.HttpServletRequest`

- Una interfaz que encapsula la funcionalidad de las peticiones que hace el cliente.
- **IMPORTANTE:** Uno de los métodos de esta interfaz es el método `getParameter (String Name)` utilizado en la mayoría de los servlets que recuperan valores de los de los formularios HTML rellenos por los clientes de las aplicaciones.
- Un objeto `HttpServletRequest` se pasa como parámetro al método **service** de la clase `HttpServlet`.

Servlet - API

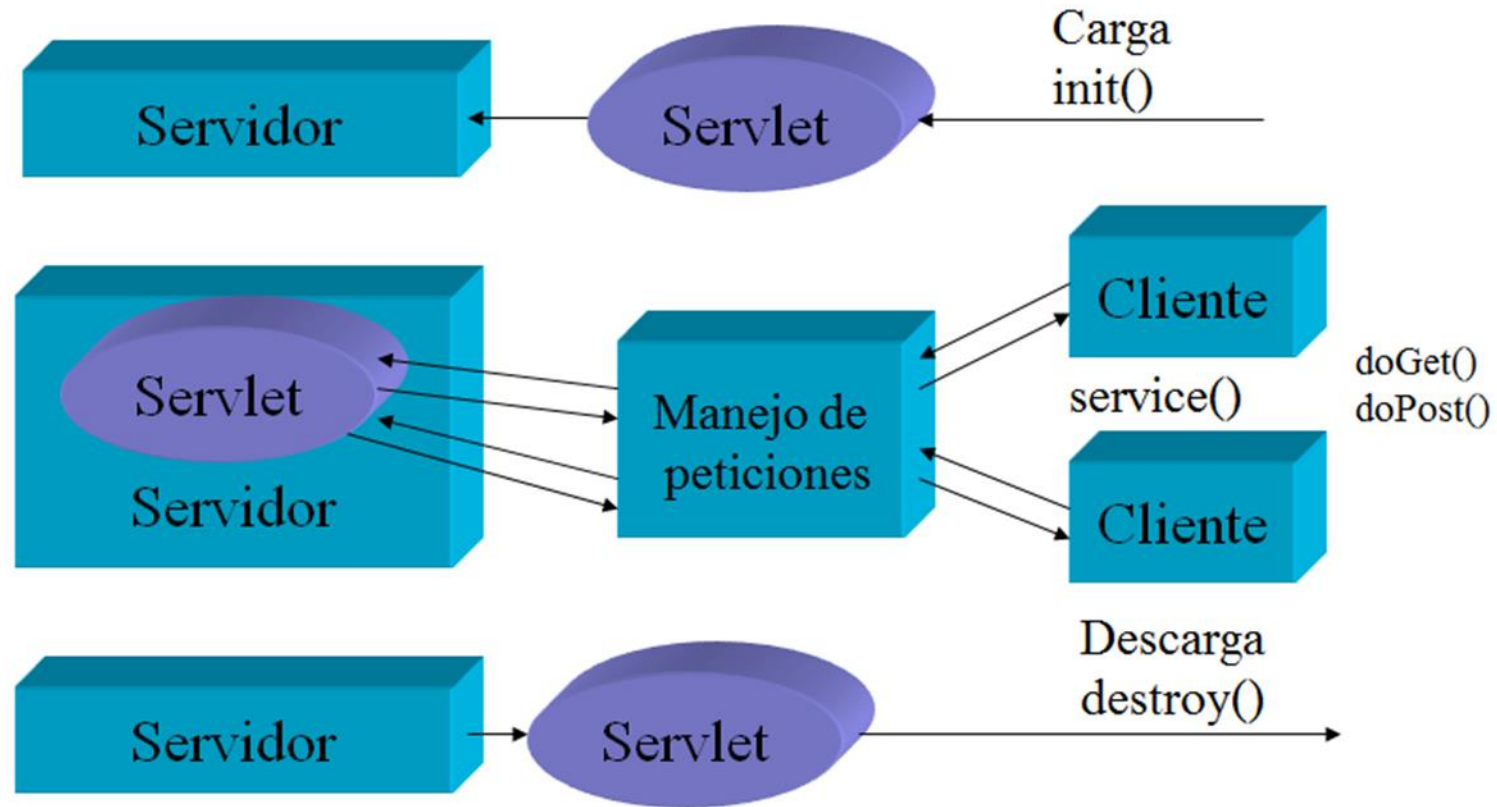
○ `javax.servlet.http.HttpServletResponse`

- Esta interfaz encapsula la funcionalidad de una respuesta HTTP, incluyendo el manejo de las cabeceras del propio protocolo.
- Interfaz que encapsula la funcionalidad de la respuesta MIME que será enviada al cliente.
- Un objeto `HttpServletResponse` se pasa como parámetro al método `service` de la clase `HttpServlet`.

Servlet – Ciclo de vida

- La interfaz que se utiliza para implementarlos define una serie de métodos para cada una de las etapas.
- El orden es el siguiente:
 - Cuando el servidor carga el Servlet invoca al método **init**.
 - Una vez ejecutado el método **init**, todas las peticiones serán atendidas por el método **service**.
 - Por último, cuando el servidor web descarga al Servlet se invocará al método **destroy**

Servlet – Ciclo de vida



Acceso a datos

- **JDBC**, Java DataBase Connectivity, es una API de Java que se usa para acceder a Bases de Datos, tanto locales como remotas.
- Existe una independencia del SGBDR: Si cambia el gestor se minimizan los cambios en la aplicación.
- Lenguaje de consulta SQL.
- El API JDBC (java.sql) está formada por cinco grupos:
 - Gestión de Drivers.
 - API para manejadores JDBC.
 - Excepciones.
 - API (trabajar con los datos)
 - Utilidades.

Gestión de Sesiones

- El protocolo HTTP **NO** posee la capacidad de almacenar estados, ¡es un protocolo sin estado!
- Posibles soluciones:
 - Cookies.
 - Añadir información en la URL
 - Usar campos ocultos de formularios (HIDDEN)
 - Empleo del objeto **HttpSession** del servlet.

Gestión de Sesiones

- Los servlets proporcionan una API denominada `HttpSession`.
- Una interfaz de alto nivel construida sobre cookies y la reescritura de las urls (de manera transparente para el desarrollador).
- Permite almacenar objetos de cualquier tipo.

Gestión de Sesiones

- Pasos para el manejo de sesiones con Servlets
 - Crear un objeto HttpSession: Usaremos el método getSession de HttpServletRequest
 - Añadir / Recuperar información:
 - `getAttribute("nombre_variable")`: devuelve un Object en caso de que la sesión tenga ya contenido en esa variable, null si no se ha utilizado nunca.
 - `setAttribute("nombre_variable",referencia)`: OJO si el objeto existe, se sobrescribe.
 - `getAttribute()`: retorna un tipo Enumeration con los nombres de todos los atributos establecidos en la sesión del usuario.