

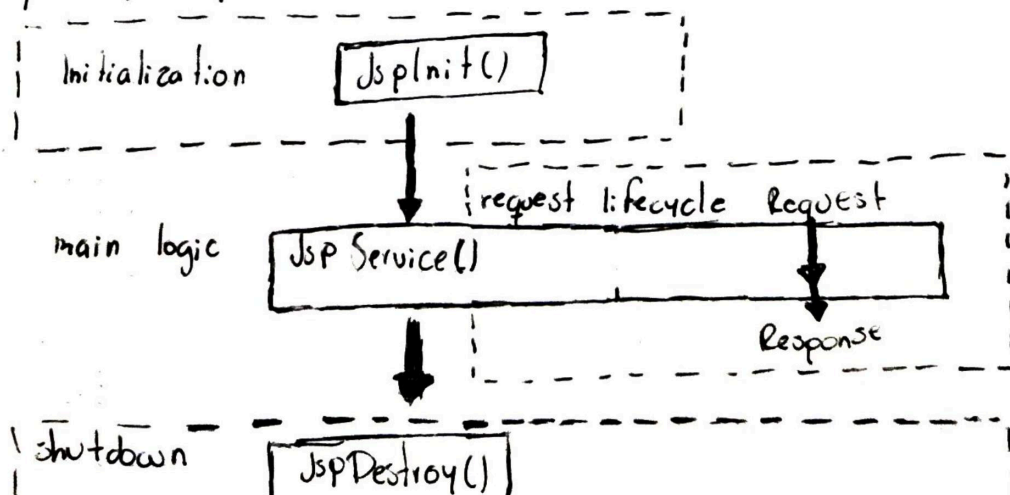
## Ciclo de vida de JSP

JSP es el ciclo de vida desde la creación hasta la destrucción, similar al ciclo de vida del Servlet, excepto que el ciclo de vida JSP también incluye un archivo JSP se compila en un Servlet

El siguiente es el ciclo de vida de JSP por etapas:

- **Compilar Fase:**  
Servlet archivos de origen de compilación  
Contenedor generador clase servlet
- **Fase de inicialización**  
Carga y correspondiente clase servlet JSP, crea una instancia y llamar a su método de inicialización
- **Fase de ejecución**  
los llamados de servicio al servlet y JSP correspondiente al método
- **Etapas de destrucción**  
Destrucción llamada al método y la correspondiente instancia de servlet JSP, entonces destruida instancia de Servlet

Etapas principales del ciclo de vida de JSP



## Compilación JSP

Cuando un navegador solicita una página JSP, motor JSP primero tendrá que comprobar si para compilar el archivo si este archivo no está compilado, o compilado fue cambiado después de la última vez

El proceso de compilación consta de tres pasos

- Analizar archivos JSP.
- El archivo JSP en el servlet.
- Servlet compilado.

## JSP INICIACIÓN

Después de que el contenedor fue cargado el archivo JSP, se llama al método `JspInit()` para la solicitud antes de proporcionar cualquier servicio. Si necesita realizar tareas de inicialización método de replicación JSP personalizada `JspInit()`

```
public void JspInit() {  
    // código de inicialización  
}
```

## JSP EJECUCIÓN

Esta etapa se describe el ciclo de vida de JSP de todas las interacciones asociadas con la solicitud hasta que se destruye

Cuando la página JSP inicialización se ha completado, el motor JSP método `JSPService()` llama

- `JSPService()` método requiere un objeto `HttpServletRequest` y un objeto `HttpServletResponse` como parámetro

```
Void _jspService(HttpServletRequest request,  
                  HttpServletResponse)
```

```
{
```

```
// código de procesamiento del lado del servidor
```

-jspService método() se llama una vez para cada solicitud y se encarga de generar la respuesta correspondiente y es responsable de generar respuestas a los siete métodos HTTP, como GET, POST, DELETE y así sucesivamente