



Recibe una cálida:

# ¡Bienvenida!

---

Te estábamos esperando 😊 

# » Conociendo los Servlets

---

**Plan formativo:** Desarrollo de Aplicaciones Full Stack Java Trainee V2.0

# HOJA DE RUTA

¿Cuáles **skill** conforman el programa?



# REPASO CLASE ANTERIOR

En la clase anterior trabajamos :

- ✓ Manejo de parámetros de GET request



# LEARNING PATHWAY

¿Sobre qué temas trabajaremos?

5.3

Start! 🏁

**Conociendo los  
Servlets**

Controlando parámetros de  
un POST request

POST request

POST request

# OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?



*Aprender a controlar parámetros de un POST request*



# › Controlando parámetros de un POST request



# Controlando parámetros de un POST request



En el caso de recibir los parámetros desde un formulario, por ejemplo, se utiliza prácticamente la misma estructura que el método `doGet()`, pero en este caso se denomina `doPost()`.



Este método **solo está accesible desde los formularios**. Se envían los parámetros de forma **implícita** junto a la página, es decir, al pasar los parámetros, nosotros no vemos reflejado en ningún sitio qué parámetros son y cuál es su valor (están ocultos).







# Controlando parámetros de un POST request



**Veamos un ejemplo de cómo controlar los parámetros de una solicitud POST en un servlet:**

Supongamos que tienes un servlet llamado MainServlet que maneja una solicitud POST con dos parámetros, "nombre" y "edad". El servlet mostrará estos parámetros en la respuesta al cliente.



```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    // Obtenemos el valor del parámetro "nombre" de la solicitud POST
    String nombre = request.getParameter("nombre");

    // Obtenemos el valor del parámetro "edad" de la solicitud POST
    String edad = request.getParameter("edad");

    // Procesamos y validamos los parámetros según sea necesario

    // Creamos la respuesta para el cliente
    response.setContentType("text/html;charset=UTF-8");
    response.getWriter().println("<html><body>");
    response.getWriter().println("<h1>¡Hola, " + nombre + "!</h1>");
    response.getWriter().println("<p>Tu edad es: " + edad + " años.</p>");
    response.getWriter().println("</body></html>");
}
```



# Controlando parámetros de un POST request



Para verlo funcionando, vamos a modificar el servlet GET con el que habíamos trabajado antes para añadirle un formulario en donde solicitamos un parámetro al que llamaremos “nombre”.

La generación del formulario, la realizaremos en el método **doGet**, porque es donde vendrá, cuando llamemos a la página desde una URL.

Haremos que el formulario vuelva por POST, por lo que reservamos el método **doPost** para recibir los parámetros que el formulario envíe.

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    // Obtenemos el valor del parámetro "nombre" de la solicitud POST
    String nombre = request.getParameter("nombre");

    // Obtenemos el valor del parámetro "edad" de la solicitud POST
    String edad = request.getParameter("edad");

    // Procesamos y validamos los parámetros según sea necesario

    // Creamos la respuesta para el cliente
    response.setContentType("text/html;charset=UTF-8");
    response.getWriter().println("<html><body>");
    response.getWriter().println("<h1>¡Hola, " + nombre + "!</h1>");
    response.getWriter().println("<p>Tu edad es: " + edad + " años.</p>");
    response.getWriter().println("</body></html>");
}
```



# Controlando parámetros de un POST request

En la línea 39, enviamos el inicio del formulario, en donde indicamos que vamos a utilizar el método **POST**, ya que así, dejaremos el método GET para crear el formulario, y en **POST** lo recibiremos y procesaremos.

El indicar "ACTION=#" hace que el formulario se envíe a la misma página que lo creo, si no, deberíamos haber indicado la URL en donde se debía enviar.

Las restantes líneas crean un label, una caja de input (que tiene el nombre del parámetro que queremos recibir!!!) y un botón para enviar.

```
27-  
28-  
29-  
30-  
31-  
32-  
33-  
34-  
35-  
36-  
37-  
38-  
39-  
40-  
41-  
42-  
43-  
44-  
45-  
46-  
/**  
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse  
 * response)  
 */  
protected void doGet(HttpServletRequest request, HttpServletResponse response)  
    throws ServletException, IOException {  
    response.setContentType("text/html");  
    PrintWriter out = response.getWriter();  
    out.println("<HTML>");  
    out.println("<HEAD><TITLE>Hola Mundo!</TITLE></HEAD>");  
    out.println("<BODY>");  
    out.println("Bienvenido a mi primera página Güev!");  
    out.println("<FORM METHOD=POST ACTION=#>");  
    out.println("<LABEL FOR=\"nombre\">Indique el nombre</LABEL>");  
    out.println("<INPUT NAME= \"nombre\"/>");  
    out.println("<INPUT TYPE= \"SUBMIT\"/>");  
    out.println("</FORM>");  
    out.println("</BODY></HTML>");  
}
```



# Controlando parámetros de un POST request



Para recibir el parámetro 'nombre', llamaremos a `getParameter` en el método `doPost`, y luego indicamos que nos muestre la información en consola.



```
52 protected void doPost(HttpServletRequest request, HttpServletResponse response)
53     throws ServletException, IOException {
54     String nombre = request.getParameter("nombre");
55     System.out.println(nombre);
56
57 }
```





# Controlando parámetros de un POST request

Si ahora intentamos lanzar el programa, veremos que en nuestro navegador aparece una caja de texto y un botón.

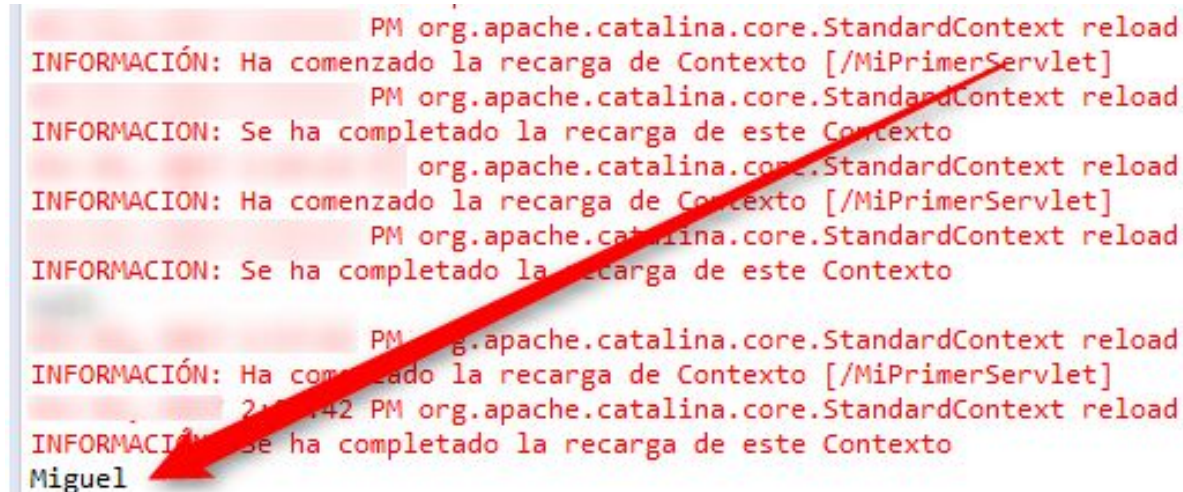




# Controlando parámetros de un POST request



Escribimos nuestro nombre y pulsamos el botón “Enviar consulta” y en la consola de Eclipse veremos el nombre



```
PM org.apache.catalina.core.StandardContext reload
INFORMACIÓN: Ha comenzado la recarga de Contexto [/MiPrimerServlet]
PM org.apache.catalina.core.StandardContext reload
INFORMACIÓN: Se ha completado la recarga de este Contexto
PM org.apache.catalina.core.StandardContext reload
INFORMACIÓN: Ha comenzado la recarga de Contexto [/MiPrimerServlet]
PM org.apache.catalina.core.StandardContext reload
INFORMACIÓN: Se ha completado la recarga de este Contexto
PM org.apache.catalina.core.StandardContext reload
INFORMACIÓN: Ha comenzado la recarga de Contexto [/MiPrimerServlet]
PM org.apache.catalina.core.StandardContext reload
INFORMACIÓN: Se ha completado la recarga de este Contexto
Miguel
```



# LIVE CODING

Ejemplo en vivo

## Primer método doPost():

*Vamos a crear un Servlet básico con los métodos vacíos para comenzar a implementar un método doPost:*

*1- Crear un servlet UserServlet con el método para procesar solicitudes GET que muestre un formulario para ingresar un nombre, un apellido y una contraseña.*

*2- Se debe validar que la contraseña sea '1234', si es correcta mostrar a través de un método doPost() los datos del usuario.*

*Si es incorrecta, mostrar un mensaje por consola que lo indique.*

**Tiempo: 30 minutos**



# Evaluación Integradora ✨

¿Listos para un nuevo desafío? En esta clase comenzamos a construir nuestro...

## Trabajo Integrador del Módulo 💪

Iremos completándolo progresivamente clase a clase.







# **Ejercicio N° 1**

# **POST request**



# POST request

## Manos a la obra: 🙌

En la clase 'UserServlet', implementar en el método doPost() en respuesta a los parámetros enviados por un formulario en doGet() .



## Consigna: 📝

- 1- En el proyecto AlkeWallet, trabajar sobre el servlet llamado UserServlet
- 2- Crear un formulario de login en doGet() y responder un saludo personalizado en un método doPost() con el nombre de usuario.

**Tiempo** 🕒: 30 minutos

○

# ¿Alguna consulta?

+



# RESUMEN

¿Qué logramos en esta clase?

- ✓ **Comprender cómo controlar los parámetros de un *POST* request**



# #WorkingTime

Continuemos ejercitando

¡Antes de cerrar la clase! Te invitamos a: 🙌🙌🙌

1. Repasar nuevamente la grabación de esta clase
2. Revisar el material compartido en la plataforma de Moodle (lo que se vio en clase y algún ejercicio adicional)
  - a. *Lectura Modulo 5, Lección 3: página 15*
3. Traer al próximo encuentro, todas tus dudas y consultas para verlas antes de iniciar nuevo tema.

# ¡Muchas Gracias!

Nos vemos en la próxima clase 🙌



Momento: ✚

# Time-out!

🕒 5 min.

