



Introducción a Java Servlets y JSP

Sesión Conceptual 2





Inicio

{desafío}
latam_



15 minutos

- Entender el concepto de multiThreading.
- Conocer las reglas para servlets concurrentes.
- Diferenciar cookie y sesión.

Objetivo



Desarrollo

{desafío}
latam_



150 minutos

/* Manejo de sesiones */

¿Qué son las cookies?

Pequeños archivos de datos creados y enviados por un sitio web y almacenados en el navegador del usuario.

Formada por una clave que le da nombre y un valor asociado.

No puede borrar ni leer información del ordenador de los internautas.

Reconoce una combinación de computador-navegador-usuario.

cookies	
ebay	
Nombre	ebay
Contenido	%5Esb%3D%23%5Ejs%3D1%5E
Dominio	.ebay.com
Ruta	/
Enviar para	Cualquier tipo de conexión
Accesible para secuencia de comandos	Si
Creada	martes, 14 de abril de 2020 a las 23:38:42
Caduca	Al finalizar la sesión de navegación

Cookies en el navegador

¿Cuál es la utilidad de las cookies?

Las aplicaciones web tienen que utilizar una serie de técnicas que permitan mantener el estado entre cliente y servidor en el transcurso del tiempo.

Ventajas y desventajas de las cookies

Ventajas	Desventajas
Navegación rápida, sencilla y personalizada para el usuario.	Sólo pueden almacenar texto.
Conocer los hábitos de navegación del usuario.	Son dependientes del navegador, esto significa que si el cliente deshabilita las cookies, ya no será posible utilizarlas.
Recordar preferencias de uso.	No se puede almacenar más de 4 Kb de información.

Aspectos legales en el uso de cookies

(RGPD)

Reglamento General de Protección de Datos.



Políticas relacionadas al uso de cookies.



Política de cookies transparente.

Resumen y responsabilidad por las cookies en la web.

Solicitud de consentimiento a través de un ícono informativo.

Posibilidad de retirar el consentimiento en cualquier momento.

Renovación de consentimiento.

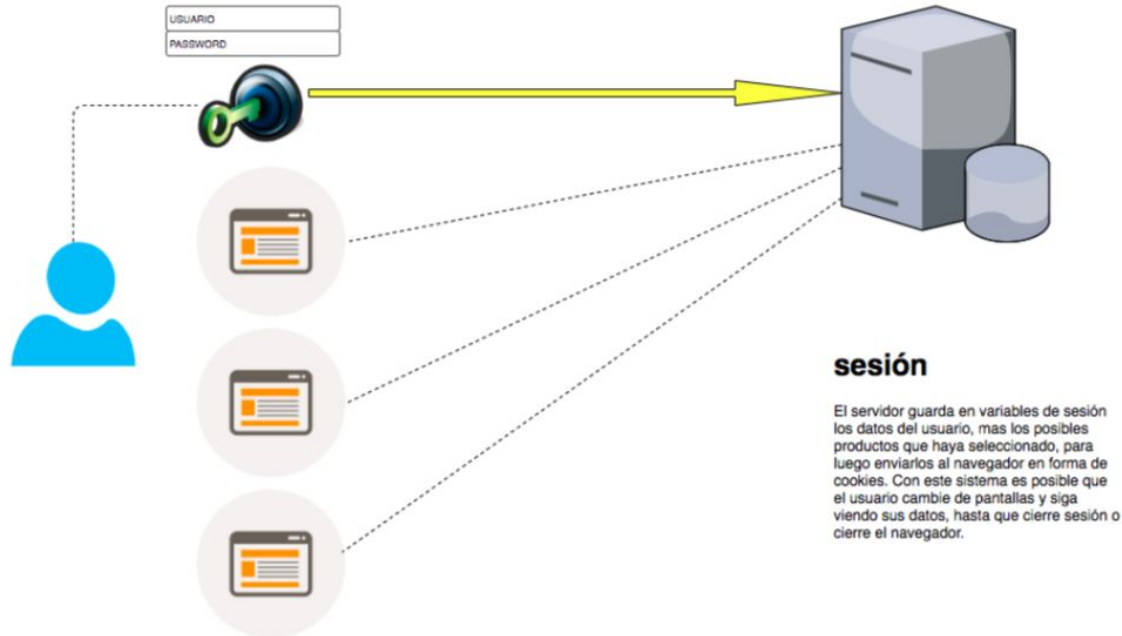
Los consentimientos deben ser almacenados.

Cookies y sesiones

- Una sesión es una serie de comunicaciones entre cliente y servidor, en la que se realiza un intercambio de información.
- Su finalización ocurre:
 - Cuando se abandona el sitio web.
 - Después de un tiempo de inactividad.
 - Cuando se cierra o reinicia el navegador.

Sesiones y servlets

Implementan su propio mecanismo de manejo de sesiones.



Ejemplo sencillo del manejo de sesión y el almacenamiento de variables asociados a un usuario.

```
public class EntradasConcierto{
    private String titularEntrada;
    private String nombreArtista;
    private String ubicacion;
    private Date fechaConcierto;
    public String getTitularEntrada() {
        return titularEntrada;
    }
    public void setTitularEntrada(String titularEntrada) {
        this.titularEntrada = titularEntrada;
    }
    public String getNombreArtista() {
        return nombreArtista;
    }
    public void setNombreArtista(String nombreArtista) {
        this.nombreArtista = nombreArtista;
    }
    public String getUbicacion() {
        return ubicacion;
    }
}
```

```
public void setUbicacion(String ubicacion) {  
    this.ubicacion = ubicacion;  
}  
public Date getFechaConcierto() {  
    return fechaConcierto;  
}  
public void setFechaConcierto(Date fechaConcierto) {  
    this.fechaConcierto = fechaConcierto;  
}  
}
```

```
@WebServlet("/creaSesion")
public class EjemploSesion extends HttpServlet {
    private static final long serialVersionUID = 1L;
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws IOException{
        String usuario = request.getParameter("nombre");
        EntradaConcierto entrada = new
EntradasConcierto();
        HttpSession sesion = request.getSession(true);
        PrintWriter writer = response.getWriter();
        entrada.setNombreArtista("Iron Maiden");
        entrada.setFechaConcierto(new Date());
        entrada.setUbicacion("Cancha");
        sesion.setAttribute("datosCompra", entrada);
        writer.println("<html><body>");
        writer.println("<h1>Entradas para
"+entradas.getNombreArtista()+"</h1>");
        writer.println("<h1>A nombre de
"+entradas.getTitularEntrada()+"</h1>");
        writer.println("</body></html>");
    }
}
```

Vamos a crear el
primer servlet

- Al llamar al servlet mediante la dirección:
 - <http://localhost:8080/proyectoSesiones/creaSesion?nombre=Juan%20Soto>
- En el servlet ocurre lo siguiente:
 - El servlet recibe el request mediante el parámetro HttpServletRequest request.
 - Obtenemos el valor que fue enviado por el usuario y lo guardamos en la variable usuario.
 - Creamos un objeto entrada de la clase EntradasConcierto.
 - Generamos la sesión mediante el request que llega desde el cliente.
 - Creamos atributos para dar valor al objeto entrada.

El código es el siguiente:

Este servlet se encarga de capturar los datos de la sesión que guardamos en la pantalla anterior.

```
@WebServlet("/obtieneSesion")
public class ObtieneSesion extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws IOException{
        HttpSession misesion = request.getSession();
        EntradasConcierto entradas =
(EntradasConcierto)misesion.getAttribute("datosCompra");
        PrintWriter writer = response.getWriter();
        writer.println("<html><body>");
        writer.println("<h1>Entradas para
"+entradas.getNombreArtista()+"</h1>");
        writer.println("<h1>A nombre de
"+entradas.getTitularEntrada()+"</h1>");
        writer.println("</body></html>");
    }
}
```

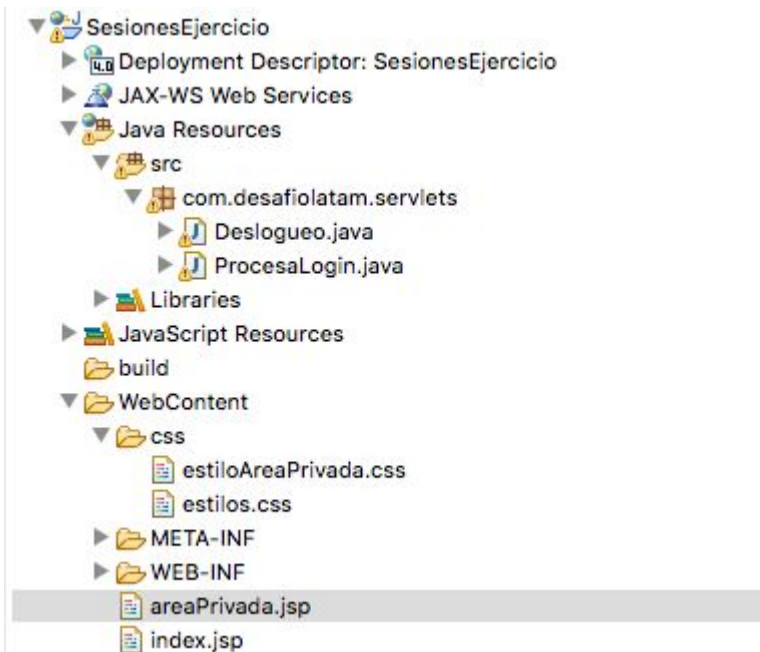
- Es tiempo de llamar al servlet obtieneSesion con la url:
 - <http://localhost:8080/proyectoSesiones/obtieneSesion>



- En Servlet EjemploSesion enviamos el nombre de la persona que está comprando la entrada:
 - <http://localhost:8080/proyectoSesiones/creaSesion?nombre=Juan%20Soto>
- Desde otro navegador, llamamos al servlet ObtieneSesion encargado de recuperar los valores que guardamos desde EjemploSesion.
 - <http://localhost:8080/proyectoSesiones/obtieneSesion>

Trabajo Práctico con Sesiones - Aplicación de variables de sesión en entorno de prueba

- Vamos a crear un pequeño sitio, donde tendremos un login que pedirá usuario y password mediante un formulario en un JSP.
- Con este ejemplo podremos ver como trabajan los JSP junto a los Servlets y las sesiones.



← → ⓘ localhost:8080/SesionesEjercicio/index.jsp ☆ 🕒 🔴 🔍 ⋮



login

password

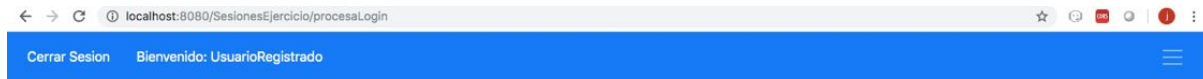
LOG IN

[Olvido la clave?](#)

{desafío}
latam_

Vista Login

Login correcto.



Area Privada

.Si estas viendo esta pagina, es por que estas logueado correctamente y el nombre de usuario esta guardado en la sesion

Generamos un objeto de tipo HttpSession.

```
<%  
    HttpSession misesion = request.getSession();  
    String nombre;  
    nombre = (String) misesion.getAttribute("Nombre");  
%>
```

Tenemos el valor de la sesión en la variable nombre.

```
Bienvenido: <%= nombre %>
```

Inicio de sesión.



/* Manejo de Cookies */

Entender funcionamiento de cookies

Vamos a crear un nuevo web dinamic proyect en eclipse.

Aplicar métodos básicos para manejo de cookies.

```
@WebServlet("/generadorCookie")
public class ManejoCookies extends HttpServlet {
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws IOException, ServletException{
        Cookie cookie = new Cookie("userjorge","viernes2019");
        cookie.setMaxAge(9000);
        cookie.setComment("Cookie guardada con propositos
educativos");
        response.addCookie(cookie);

        PrintWriter print = response.getWriter();
        print.println("<html><body>");
        print.println("<h2><i>Cookie guardada
correctamente:</i></h2><br>");
        print.println("Valor de la cookie: <strong>" +
cookie.getValue() + "</strong>");
        print.println("<br>");
        print.println("Tiempo de la duración de la cookie:
<strong>" + cookie.getMaxAge()+"</strong>");
        print.println("<br>");
        print.println("Comentario: <strong>" +
cookie.getComment() + "</strong>");
        print.println("</body></html>");
    }
}
```

Generar un constructor:

El constructor acepta valores de la clase String, pero por norma solo puede recibir valores alfanuméricos.

```
Public Cookie(java.lang.String name,java.lang.String value)
```

Cuenta con varios métodos que dan funcionalidad y permiten generar y obtener valores de la cookie creada.

setMaxAge(int expiración)	Setea el tiempo máximo de expiración de la cookie en segundos.
setComment(String comentario)	Añade un comentario que representa el propósito de la cookie.
addCookie(Cookie cookie)	Añade la cookie al response http, para que el servlet la envíe al navegador.
getValue()	Retorna el valor de la cookie.
getComment()	Retorna el valor del comentario.
getMaxAge()	Retorna el tiempo máximo de expiración.

Armar cuerpo de cookies para su recuperación

El segundo servlet se encargará de obtener los valores de la cookie que guardamos anteriormente.

Ciclo iterativo.

{desafío}
latam_

```
@WebServlet("/recuperaCookie")
public class RecuperaCookie extends HttpServlet {
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws IOException,
        ServletException{
        Cookie[] cookies = request.getCookies();
        PrintWriter print = response.getWriter();
        print.println("<html><body>");
        print.println("<h2><i>Cookie recuperada
correctamente:</i></h2><br>");
        for(int i = 0; i < cookies.length; i++) {
            print.println("Id de la cookie: <strong>"+
cookies[i].getValue() + "</strong>");
            print.println("<br>");
            print.println("Valor de la cookie:
<strong>" + cookies[i].getName()+"</strong>");
            print.println("<br>");
        }
        print.println("</body></html>");
    }
}
```

Genera un objeto de la clase Cookie.

```
Cookie[] cookies = request.getCookies();
```

Accedemos a los valores mediante su índice.

```
[i].getValue();  
[i].getName();
```

- Para probar el funcionamiento del proyecto, ejecutar primero el servlet ManejoCookies.java mediante la direccion URL:
 - <http://localhost:8080/Manejo Cookie/generaCookie>



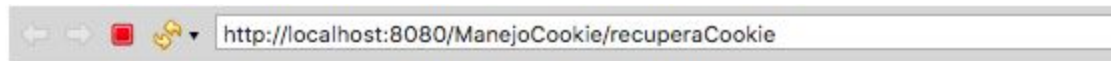
Cookie guardada correctamente:

Valor de la cookie: **viernes2019**

Tiempo de duracion de cookie: **9000**

Comentario: **Cookie guardada con propositos educativos**

- Para recuperarlas es necesario ejecutar el segundo servlet de nombre RecuperaCookie.java mediante la url:
 - <http://localhost:8080/ManejoCookie/recuperaCookie>



Cookie recuperada correctamente:

Id de la cookie: **viernes2019**

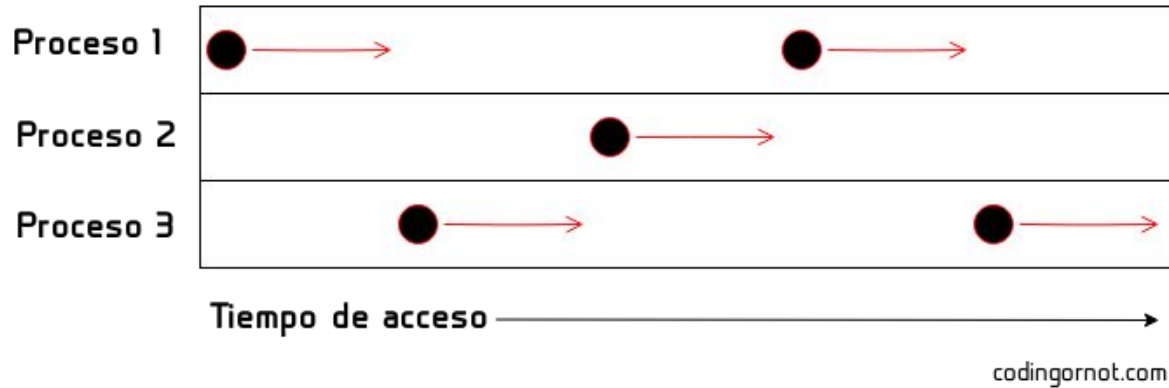
Valor de la cookie: **userjorge**

/* Concurrencia */

¿Qué es la concurrencia?

- Cuando ocurren varios sucesos de manera contemporánea.
- Dos procesos son concurrentes, cuando son procesados al mismo tiempo.

Funcionamiento de la concurrencia.



¿Qué es un hilo?



Que es la programación MultiHilos

- Pueden ejecutar varios hilos de ejecución al mismo tiempo.
- Nuestro windows-linux-mac es por definición multitarea, puedes estar reproduciendo una película en tu reproductor además de un navegador web y por otro lado un archivo word.

Trabajo multihilos con servlets

Los servlets pueden tener la característica de thread safe, puede trabajar con múltiples hilos de ejecución sin morir en el intento.

Reglas específicas de thread safe:

- El método `service()` del servlet no debe acceder a ninguna variable miembro a menos que estas variables miembro sean seguras para subprocesos.
- No debe reasignar las variables miembro, ya que esto puede afectar a otros subprocesos que se ejecutan dentro del método `service()`.



Quiz



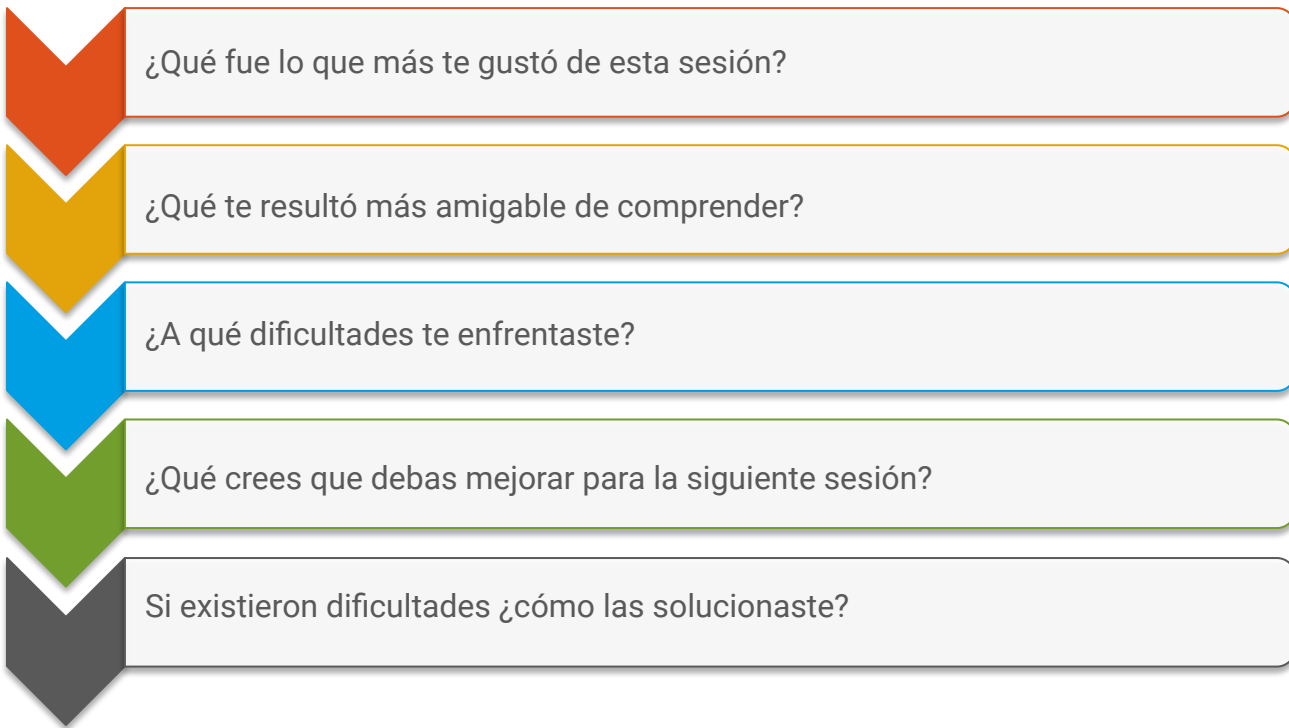


Cierre

{desafío}
latam_



15 minutos



¿Qué fue lo que más te gustó de esta sesión?

¿Qué te resultó más amigable de comprender?

¿A qué dificultades te enfrentaste?

¿Qué crees que debas mejorar para la siguiente sesión?

Si existieron dificultades ¿cómo las solucionaste?



*Academia de
talentos digitales*

www.desafiolatam.com



/DesafioLatam



/DesafioLatam



/DesafioLatam



/DesafioLatam