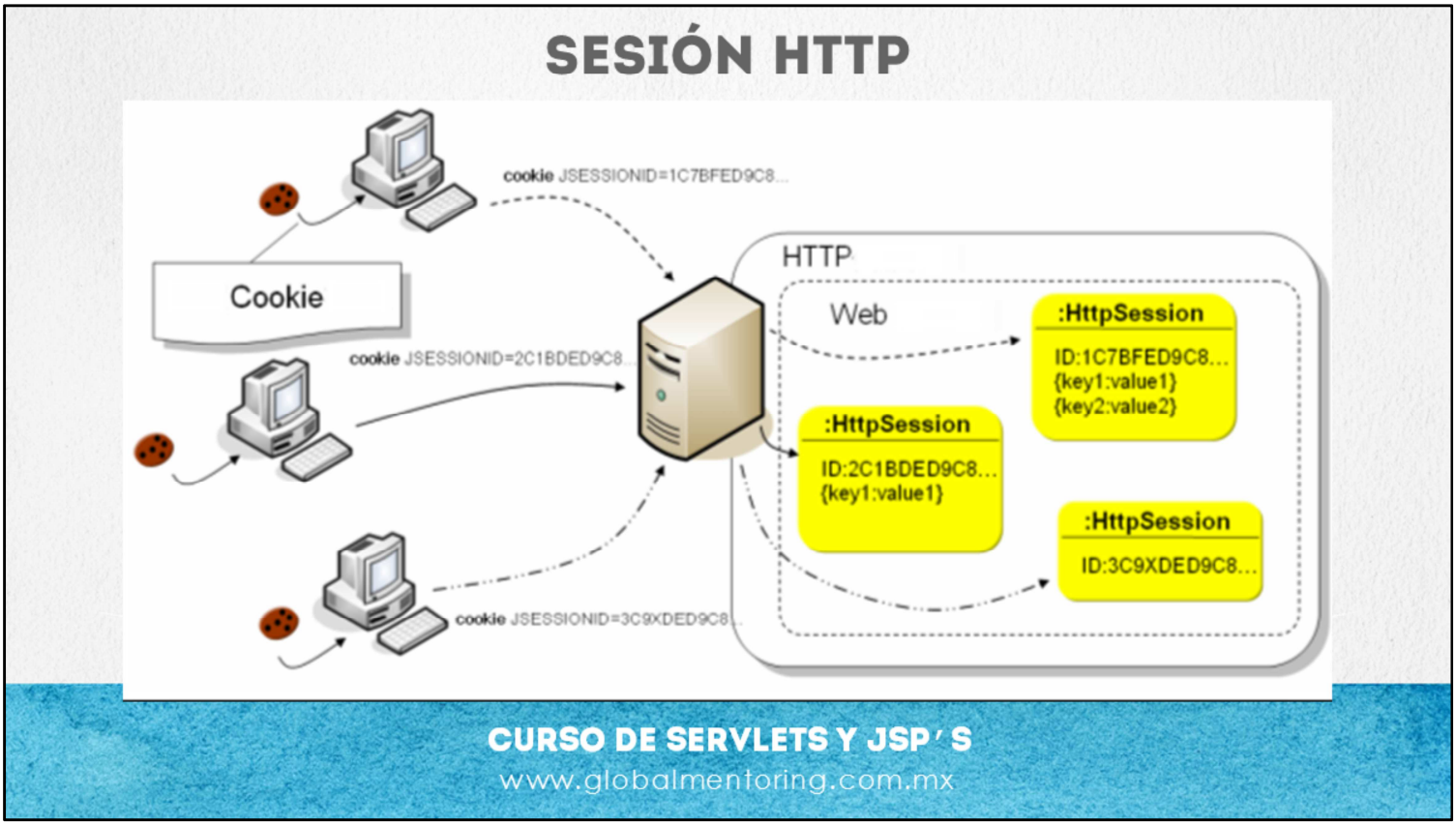
Hola, te saluda nuevamente Ubaldo Acosta. Espero que estés listo para comenzar con esta lección..

Vamos a estudiar el tema de Manejo de Sesiones con el objeto HttpSession.

¿Estás listo? ¡Vamos!



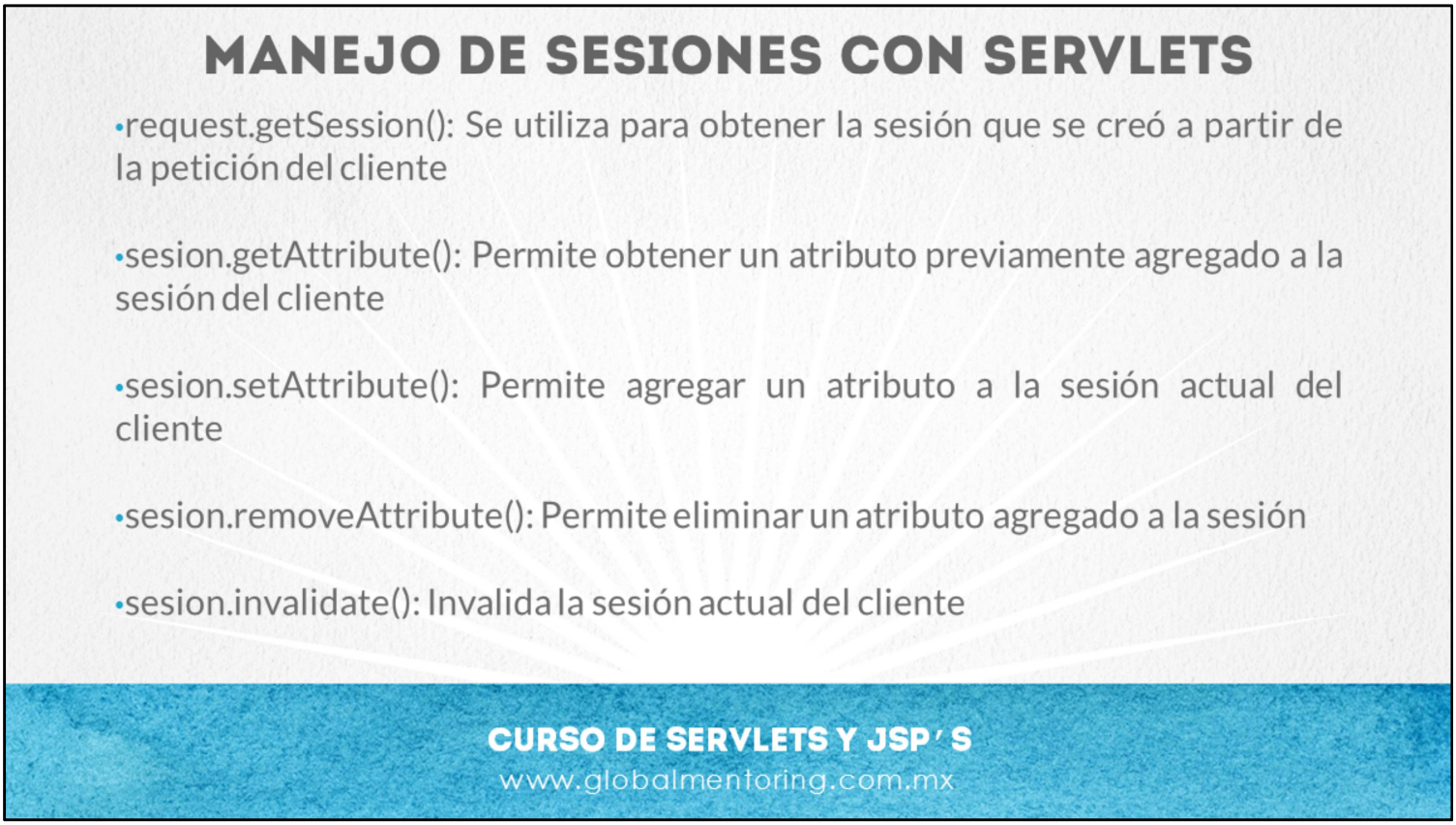
Vamos a revisar ahora el concepto de HttpSession.

El API de los Servlets nos permite administrar las sesiones de los clientes por medio de un objeto llamado HttpSession y cabe resaltar que esto ocurre de manera automática, cada petición que realiza un cliente hacia nuestro recurso web de manera automática crea una nueva sesión por ejemplo si tenemos dos navegadores distintos en la misma PC, uno utilizando Internet Explorer y en otro utilizando Firefox se consideran distintos clientes debido a que es diferente navegador web y cada cliente crea una nueva sesión en el servidor web, pero si tenemos una nueva ventana del mismo navegador web por ejemplo si abrimos una nueva ventana de Firefox esto no se considera un nuevo cliente y por lo tanto no va a crear una nueva sesión del lado del servidor.

Ahora una sesión nos va a permitir administrar las peticiones que realiza un usuario, hay que recordar que el protocolo HTTP se considera que es un protocolo que no tiene estado, por lo que NO va a recordar información que el usuario haya enviado previamente.

El objeto HttpSession se obtiene a partir del objeto HttpServletRequest (request). Por lo que una sesión se utiliza para administrar las distintas peticiones (request) del usuario.

Una sesión entonces tiene una duración de vida más larga que una petición o lo que es lo mismo el objeto request. Por lo tanto una sesión se destruye hasta que transcurra el tiempo que le asignamos a la sesión de inactividad o también podemos destruirla manualmente por medio del método invalidate. Más adelante veremos un ejemplo de este tema.



Vamos a estudiar ahora el API de los Servlets para manejar el concepto de sesiones.

En primera instancia para poder obtener una sesión vamos a utilizar el objeto HTTPServletRequest y por medio de este objeto vamos a crear una sesión con el siguiente método: reques.getSession().

La sesión creada se asociada con cliente que generó la petición. Vamos a revisar varios métodos más. Una vez que tenemos el objeto de sesión, de los métodos que más utilizaremos son:

getAttribute(): Este método nos permite obtener un atributo previamente agregado a la sesión del cliente.

setAttribute(): Este método de igual manera está alojado dentro del objeto HttSession. Este método setAttribute() permite agregar nueva información del cliente en la sesión web y así como podemos recuperar y agregar nuevos atributos también el API de los Servlets nos permite remover los atributos que ya no estemos utilizando en una sesión HTTP.

sesion.invalidate(): Este método nos permite eliminar la sesión actual del cliente y también va a eliminar cualquier atributo que hemos agregado previamente por medio del método serAttribute().

Otros métodos importantes por ejemplo el método sesion.isNew(): este método nos permite conocer si la sesión ha sido recientemente creada esto es en la primera petición del cliente si el cliente hace una segunda petición al recurso web esta sesión ya ha sido creada previamente y por lo tanto este método isNew() regresaría un falso, en cambio si es la primera petición este método regresa verdadero.

También tenemos el método sesion.get.CreationAccesedTime() y este método nos permite saber cuál es el tiempo en que se creó esta sesión HTTP

También tenemos el método sesion.get.LastAccesedTime() este método nos permite conocer de una vez en que esta sesión fue accedida por parte del cliente.

Posteriormente tenemos el método sesion.getMaxInactiveInterval() este método nos permite conocer el tiempo de inactividad con que se ha definido esta sesión y esto se mide en segundos, entonces, si este método nos regresa por ejemplo un valor de 3600 segundos, significa que esta sesión va a esperar una hora hasta que se destruya si es que no recibe ninguna petición por parte del cliente, entonces si pasa una hora de inactividad por parte del usuario entonces la sesión actual se va a eliminar y el usuario va a perder toda la información que se haya almacenado en esta sesión HTTP.

Cabe resaltar que este valor lo podemos modificar por medio del método setMaxInactiveInterval() y nos va a permitir especificar el tiempo de inactividad que vamos a permitir que transcurra si es que el usuario no realiza una petición HTTP. Ahora este valor de inactividad también lo podemos configurar desde el archivo de web.xml que vamos a ver posteriormente también si lo configuramos dentro del archivo web.xml se dice que es una configuración de manera declarativa debido a que no estamos agregando código Java para modificar ese valor sino simplemente estamos declarando el valor en el archivo web.xml. En cambio, si lo hacemos por medio del método setMaxInactiveInterval() se dice que estamos modificando el valor de manera programática debido a que estamos utilizando código Java para modificar el valor de ese atributo, vamos a revisar a continuación algunos ejercicios para revisar el

API de sesiones de los Servlets.

EJERCICIOS CURSO PROGRAMACIÓN CON JAVA

- **ABRIR LOS ARCHIVOS DE EJERCICIOS EN PDF.**
- **EJERCICIO**: Ejercicio Manejo de Sesiones con HttpSesion.

CURSO DE SERVLETS Y JSP'S
www.globalmentoring.com.mx

En Global Mentoring promovemos la Pasión por la Tecnología Java. Te invitamos a visitar nuestro sitio Web donde encontrarás cursos Java Online desde Niveles Básicos, Intermedios y Avanzados, y así te conviertas en un experto programador Java.

A continuación te presentamos nuestro listado de cursos:

- Lógica de Programación
- Fundamentos de Java
- Programación con Java
- Java con JDBC
- HTML, CSS y JavaScript
- Servlets y JSP's
- Struts Framework
- Hibernate Framework
- Spring Framework
- JavaServer Faces
- Java EE (EJB, JPA y Web Services)
- JBoss Administration
- Android con Java
- HTML5 y CSS3

Datos de Contacto:

Sitio Web: www.globalmentoring.com.mx
Email: informes@globalmentoring.com.mx