

Desarrollo de Aplicaciones Web

Servlets. Sesiones.

Cookies

Cookie.- fragmento de información que se almacena en el disco duro del cliente a petición del servidor de la página.

Características de las cookies:

- Estructura.- (nombre, valor).
- Protocolo.- servidor envía cookies a cliente, cliente retorna cookie cuando realiza otra solicitud.
- Mecanismo intercambio.- automático. Usuario no interviene.
- Vida.- durante el tiempo estipulado o durante sesión (*jsessionid*).

Cookies

<<interface>> javax.servlet.http.Cookie
Cookie(String, String)
String getDomain() int getMaxAge() String getName() String getPath() boolean getSecure() String getValue() void setDomain(String) void setMaxAge(int) void setPath(String) void setValue(String) <i>otros métodos ...</i>

Cookies. Otras aplicaciones

Creación cookie nueva:

```
Cookie cookie= new Cookie("nombre_cookie", valor);
```

Fijación tiempo de vida:

```
cookie.setMaxAge(30*60);
```

Envío al cliente:

```
response.addCookie(cookie);
```

Cookies. Otras aplicaciones

Ej.- Extracción cookie(s) del cliente:

```
Cookie[ ] cookies= request.getCookies();
for (int i= 0; i<cookies.length; i++){
    Cookie cookie= cookies[i];
    if (cookie.getName().equals("mi_cookie")){
        String miCookie= cookie.getValue();
        out.println("Hello " + miCookie);
        break;
    }
}
```

Servlets. Sesiones.

[http](#) es un protocolo sin estado → cada petición es tratada de forma independiente por el servidor → el servidor no puede saber si una petición proviene del mismo o diferentes casos y si están relacionados entre sí.

Solucion.- crear sesiones, que permiten identificar y guardar información individual de cada cliente.

Sesión:

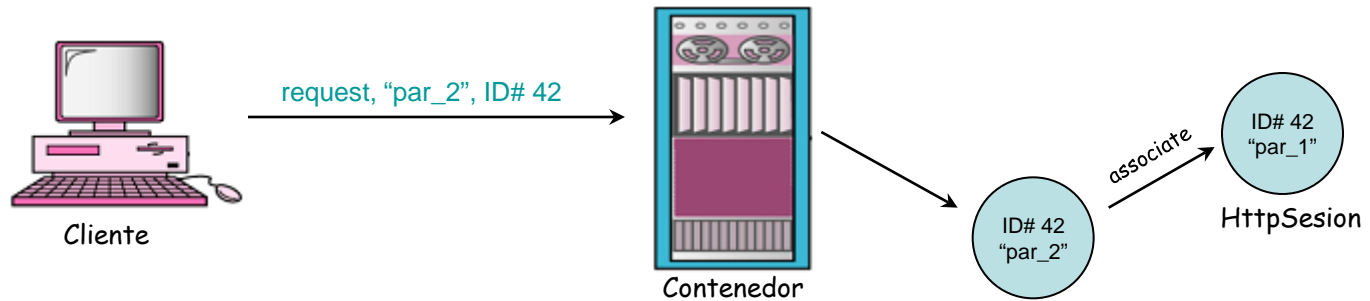
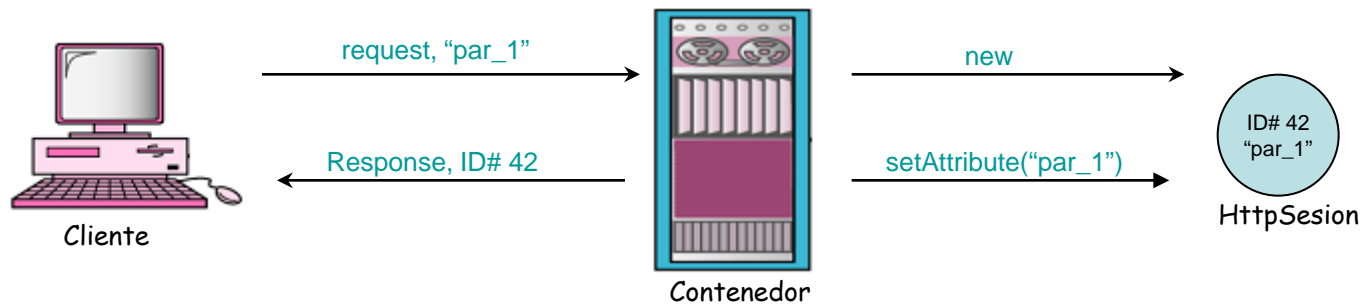
- Inicio.- al realizar una petición al servidor.
- Fin:
 - Al cabo de un tiempo de inactividad.
 - Al cerrarse la aplicación cliente.
 - Cierre por parte del cliente.

Seguimiento sesión:

- Cookies
- Reescritura de URL
- Campos ocultos en formularios

Sesiones

Creación de una sesión

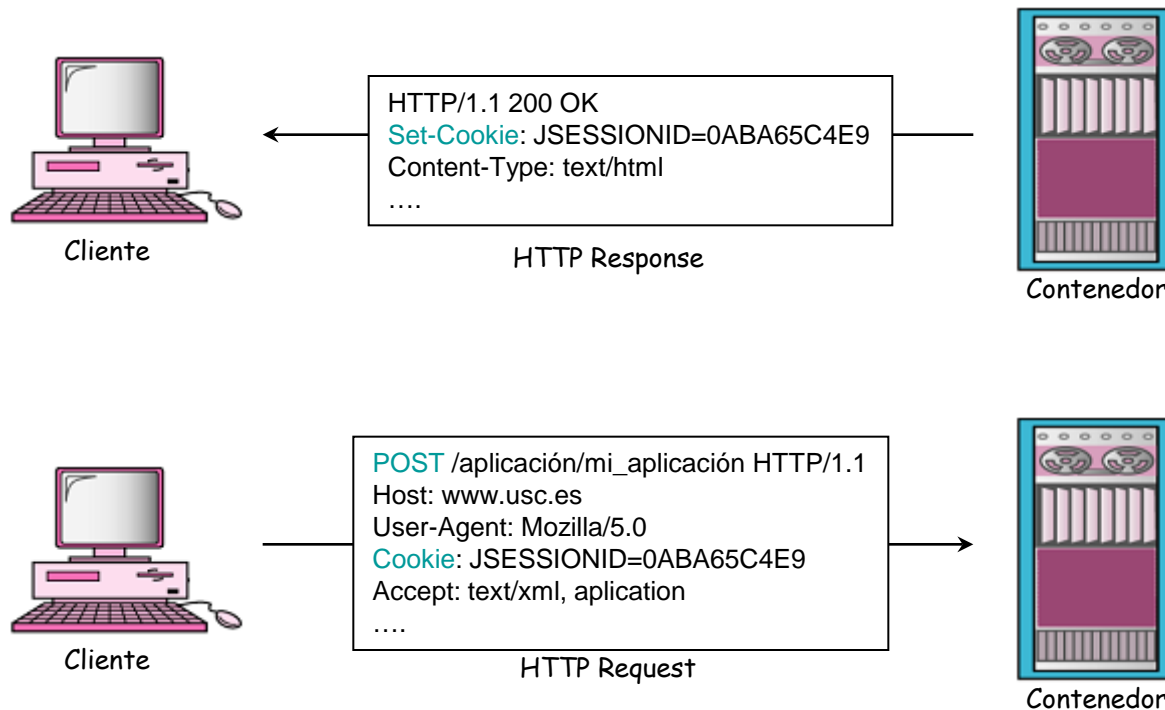


Sesiones. HttpSession Interface.

<<interface>> javax.servlet.http.HttpSession
Object getAttribute(String) long getCreationTime() String getId() long getLastAccessedTime() int getMaxInactiveInterval() ServletContext getServletContext() void invalidate() boolean isNew() void removeAttribute(String) void setAttribute(String, Object) void setMaxInactiveInterval(int) <i>otros métodos ...</i>

Sesiones. Cookies.

Cookie.- pequeña cantidad de información que se almacena en la máquina Cliente.



Sesiones. Cookies.

Envío de un cookie de sesión en la respuesta:

```
HttpSession session= request.getSession();
```

Tomar el ID de sesión en la solicitud:

```
HttpSession session= request.getSession();
```

Consultar si la sesión creada es nueva:

```
HttpSession session= request.getSession();
```

```
if (session.isNew())
```

```
else
```

Sesiones. Cookies.

Consultar si existe una sesión previa:

```
HttpSession session= request.getSession(false);
```

```
if (session==null)
```

```
    _____
```

```
else
```

```
    _____
```

Sesiones. Cookies.

Ej.-

```
public void doGet(HttpServletRequest request, HttpServletResponse response)  
    throws IOException, ServletException{
```

```
    response.setContentType("text/html");  
    PrintWriter out= response.getWriter();  
    out.println("Testeo de sesiones");
```

```
    HttpSession session= request.getSession(false);
```

```
    if (session==null){  
        out.println("No hay sesiones disponibles");  
        out.println("Abriendo una sesión nueva...");  
        session= request.getSession( );
```

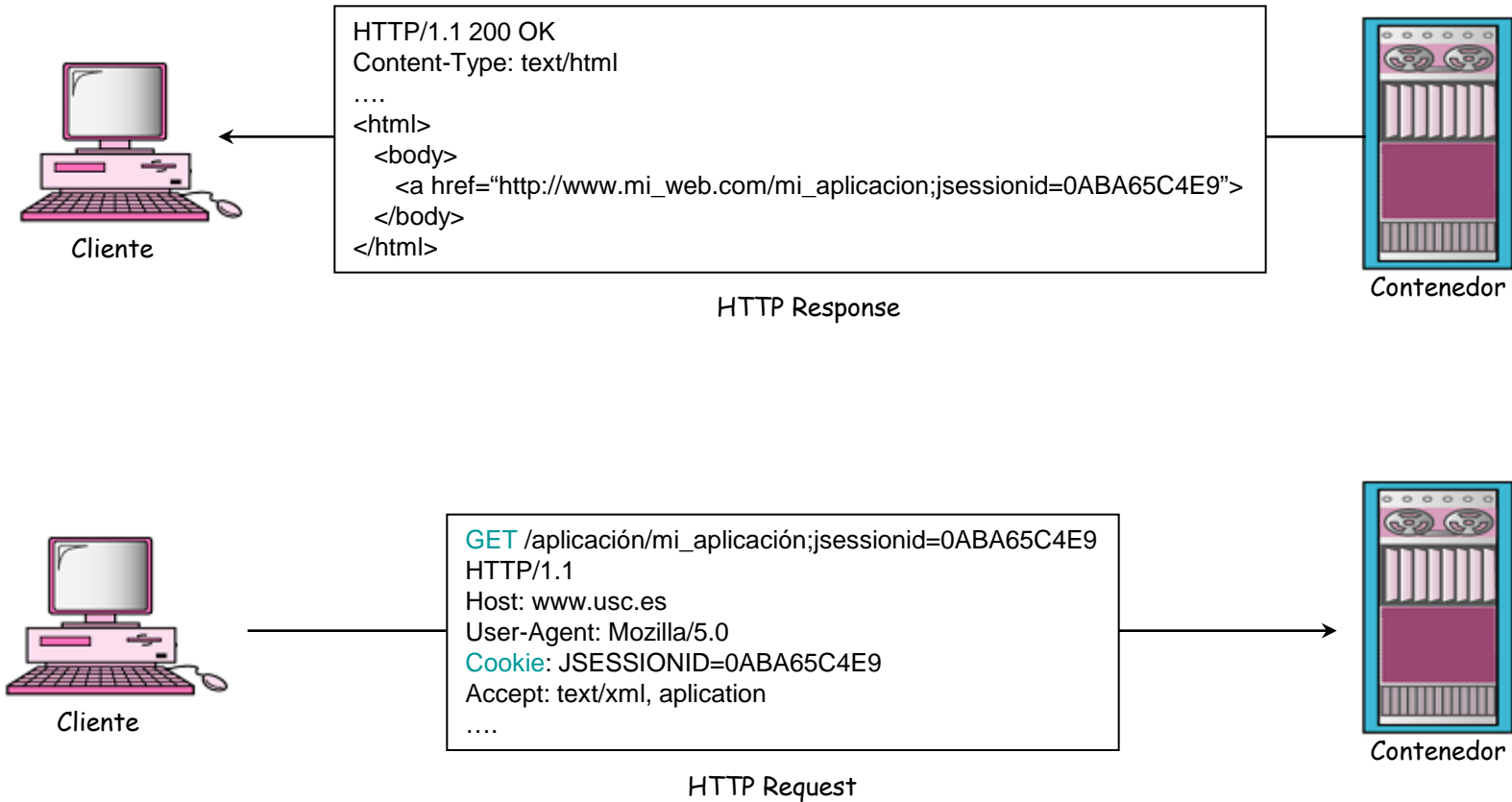
```
    }
```

```
    else
```

```
        out.println("Hay una sesión abierta");
```

```
}
```

Sesiones. Reescritura de URL.



Sesiones. Reescritura de URL.

Mediante los métodos: *encodeURL()* y *encodeRedirectURL()* del objeto *response*

Ej.-

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws IOException, ServletException{

    response.setContentType("text/html");
    PrintWriter out= response.getWriter();
    HttpSession session= request.getSession( );

    out.println("<html><body>");
    out.println("<a href=\""+response.encodeURL("miaplicación")+"\">púlsame</a>");
    out.println("<body><html>");
}
```

Sesiones. *Timeout.*

Existen tres métodos:

i) Mediante configuración en xml de despliegue:

Ej.-

```
<web-app>
    <servlet>
        . . .
    </servlet>
    <session-config>
        <session-timeout>15</session-timeout>
    </session-config>
</web-app>
```

Sesiones. *Timeout.*

ii) Mediante especificación de tiempo máximo:

Ej.-

```
session.setMaxInactiveInterval(15*60);
```

iii) Mediante métodos de finalización de sesión:

Ej.-

```
session.invalidate();
```