

# Desarrollo de Aplicaciones Web

**Parámetros, Atributos y Cookies.**

# Parámetros de inicialización

**Parámetro.-** dato de tipo *string* , que permite almacenar información utilizada en la aplicación correspondiente.

Resuelve el inconveniente de tener que introducir dicha información como parte del código a compilar.

Ej.- 

```
PrintWriter out= response.getWriter();  
out.println("mrpablo@usc.es");
```

# Parámetros de inicialización

**Solución** mediante uso de parámetros del propio servlet

```
<servlet>
  <servlet-name> ..... <servlet-name>
  <servlet-class> ..... </servlet-class>

  <init-param>
    <param-name> miEmail </param-name>
    <param-value> miEmail@usc.es </param-value>
  </init-param>
  ....
  ....
</servlet>
```

```
out.println(getServletConfig().getInitParameter("miEmail"));
```

# Parámetros de contexto

Parámetros análogos a los de inicialización, pero disponibles para la aplicación completa.

```
<web-app>
  <servlet>
    <servlet-name> ..... <servlet-name>
    <servlet-class> ..... </servlet-class>
  </servlet>

  <context-param>
    <param-name> miEmail </param-name>
    <param-value> miEmail@usc.es </param-value>
  </context-param>
</web-app>
```

```
ServletContext context= getServletContext();
out.println(context.getInitParameter("miEmail"));
```

# Atributos

**Atributo.**- objeto que consta de nombre y valor. Además tiene un periodo de validez (vida) determinado.

Pueden manejarse en términos de contexto, solicitud o sesión, existiendo métodos adecuados para las interfaces de cada caso respectivo:

*ServletContext*, *ServletRequest* y *HttpSession*. Estos métodos son:

- *getAttributeNames();*
- *getAttribute(String);*
- *setAttribute(String);*
- *removeAttribute(String);*

# Atributos

Ej.-

```
public void doGet(HttpServletRequest, .....) ...{  
  
    response.setContentType("text/html");  
  
    PrintWriter out= response.getWriter();  
    out.println("Testeo de atributos de contexto<br>");  
  
    getServletContext().setAttribute("PBI", "6");  
    getServletContext().setAttribute("DAAI", "7,5");  
  
    out.println(getServletContext().getAttribute("PBI");  
    out.println(getServletContext().getAttribute("DAAI");  
}
```

# 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

<b>&lt;&lt;interface&gt;&gt;</b> <b>javax.servlet.http.Cookie</b>
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

## 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

**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;  
    }  
}
```