



AUT04_04. Ampliación Consumo Eléctrico

1. Objetivos

Los objetivos de esta actividad son:

- Profundizar en el concepto de patrón MVC
- Repasar y utilizar nuevos métodos del API JDBC.
- Comunicación entre páginas a través de parámetros de request.
- Utilización de cookies para almacenar preferencias de usuario

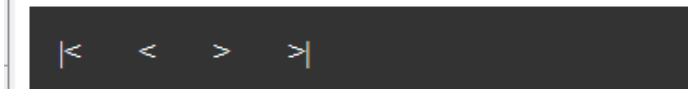
2. Ampliaciones

2.1 Modificar ConsumoClientes para que siga el patrón MVC. El proyecto se llamará ConsumoElectrico_nombre.

Modificar el proyecto *Consumo Electrico* para que el responsable de “pintar” o renderizar la salida sea una página JSP **misclientesList.jsp**. Para ello el servlet almacenará en la request la lista de clientes y pasará el control (forward) a la JSP.

La JSP recuperará de request la lista de empleados y la pintará, utilizando scriplets y expresiones JSP.

2.2 Añadir paginación a la lista de misclientes.



Se añadirá un parámetro en la request (por ejemplo ?page=5) que indicará a la JSP qué página tiene que mostrar.

Para ello es necesario definir el valor del tamaño de página (pageSize) que puede ser un valor constante o una propiedad de empleados.properties.

Para indicar a MySQL la paginación, hay que añadir al final de las consultas la cláusula LIMIT.

LIMIT *primerReg*, *NumReg*

Por ejemplo:

LIMIT 0,20 → Muestra los primeros 20 registros

LIMIT 20,20 → Muestra los 20 siguiente.....

Si el parámetro **page** es null se asumirá que tiene valor 0.

La página JSP generará unos link de navegación que permitirán avanzar y retroceder por páginas.



2.3 Permitir cambiar el tamaño de página

Permitir modificar el tamaño de página mostrada en la lista de misclientes.

Para ello:

1. Se creará un formulario para cambiar el tamaño (puede ir al POST del servlet).

Buscador de clientes

Número de salidas por página	
Salidas	<input type="text"/>
	<input type="button" value="Enviar"/>
Datos cliente	
Nombre	<input type="text"/>
	<input type="button" value="Enviar"/>

2. Se añadirá un método al servlet para crear la cookie en base al valor suministrado.

El código para crear la cookie sería algo así como:

```
Integer customPageSize = ParseUtils.  
    safeParseInt(request.getParameter("pageSize"));  
  
if (customPageSize != null) {  
    Cookie pageSizeCookie = new Cookie(PAGE_SIZE_COOKIE_NAME,  
        customPageSize.toString());  
  
    // 1 año de vida  
    pageSizeCookie.setMaxAge(24 * 60 * 60 * 365);  
    pageSizeCookie.setPath(request.getContextPath());  
    response.addCookie(pageSizeCookie);  
}
```

3. Para leer la cookie algo como:

Tengo una clase ServletUtils para facilitar la recuperación de cookies.

```
public class ServletUtils {  
  
    public static Cookie getCookie(HttpServletRequest request, String cookieName) {  
  
        if (request.getCookies() == null)  
            return null;  
  
    }  
}
```



```
for(Cookie cookie : request.getCookies()) {  
  
    if (cookie.getName().equalsIgnoreCase(cookieName))  
        return cookie;  
}  
return null;  
  
}
```

Utilizando esa clase recupero el valor de la cookie.

```
Cookie pageSizeCookie = ServletUtils.getCookie(request, PAGE_SIZE_COOKIE_NAME);  
  
if (pageSizeCookie != null) {  
    return ParseUtils.safeParseInt(pageSizeCookie.getValue());  
} else {  
    return null;  
}
```