



AUT04_04 Ampliaciones empleados

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 Empleados para que siga el patrón MVC

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

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

2.2 Añadir paginación a la lista de empleados

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 siguientes

Nº	Nombre	Fecha Nac.	Fecha Cont.
10001	Georgi,Facello	1953-09-02	1986-06-26
10002	Bezalel,Simmel	1964-06-02	1985-11-21
10003	Parto,Bamford	1959-12-03	1986-08-28
10004	Chirstian,Koblick	1954-05-01	1986-12-01
10005	Kyoichi,Maliniak	1955-01-21	1989-09-12
10006	Anneke,Preusig	1953-04-20	1989-06-02
-	-	-	-
10016	Kazuhito,Cappelletti	1961-05-02	1995-01-27
10017	Cristinel,Bouloucos	1958-07-06	1993-08-03
10018	Kazuhide,Peha	1954-06-19	1987-04-03
10019	Lillian,Haddadi	1953-01-23	1999-04-30
10020	Mayuko,Warwick	1952-12-24	1991-01-26



...

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.

No bootstrap.

2.3 Permitir cambiar el tamaño de página

Permitir modificar el tamaño de página mostrada en la lista de empleados. **Buscador de empleados**

Para ello:

1. Se creará un formulario para cambiar el tamaño (puede ir al POST del servlet)
2. Se añadirá un método al servlet para crear la cookie en base al valor suministrado.

Configurar servlet

Tamaño página

Datos empleado

Nombre

Apellidos

El código para crear la cookie sería 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;

    }
```