TIC News

Programación en internet

Memoria 2: Programación en servidor

Guillermo Barco Muñoz

Abril de 2016

Table of Contents

Introducción

El portal TIC News es un agregador de noticias relacionadas con el mundo de la tecnología.

- Un usuario puede darse de alta en la aplicación o acceder si ya está registrado.
- Envíar una noticia, ver las noticias que ha enviado y editarlas.
- Ver noticias de otros usuarios, ver las más valoradas.
- Añadir un comentario en una noticia de otro usuario o decir si le gusta o no le gusta.
- Ver su perfil de usuario o el de otros usuarios.

Base de Datos y DAO

La base de datos utilizada es la recomendada por el profesor de la asignatura, HSQLDB. Aunque es muy básica, es suficiente para el desarrollo de la práctica y se puede utilizar incluyendo un .jar.

Hay tres interfaces DAO, la de comentarios, noticias y usuarios.

Atención de las peticiones GET y POST

Siguiendo el patrón de diseño post redirect get, todos las peticiones por post, hacen una redirección a otra petición get para evitar multiples accesos al DAO con los mismos datos.

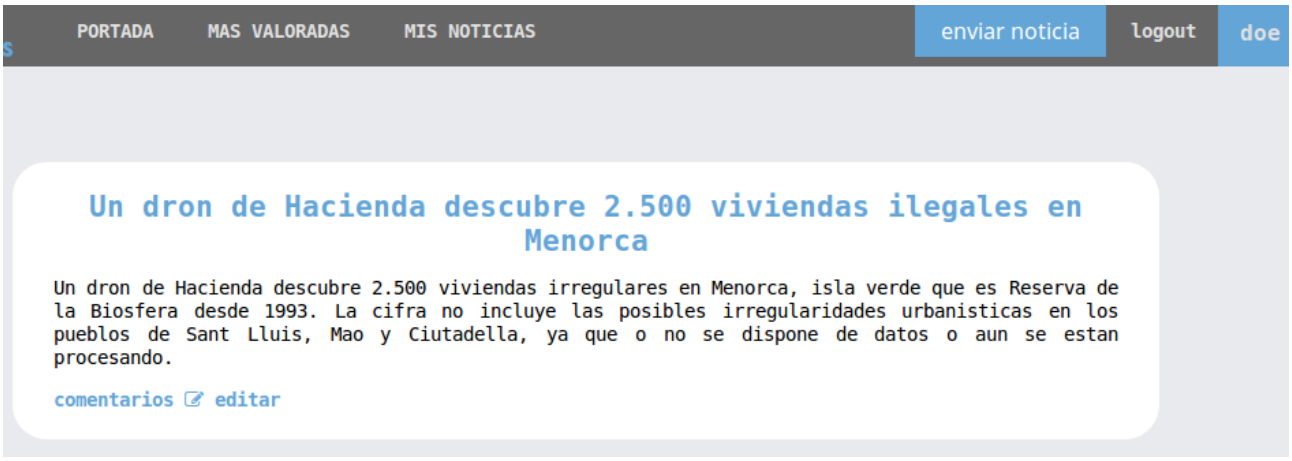
```
if (checkUserAndPassword(registredUser)) {

    createUserSession(registredUser);
    MyLogger.logMessage("User log in correct");
    response.sendRedirect(request.getContextPath() + "/Index");

} else {

    MyLogger.logMessage("User log in error");
    request.setAttribute("messages", "Usuario o contrasena incorrecta");
    RequestDispatcher view = request.getRequestDispatcher("/WEB-INF/login.jsp");
    view.forward(request, response);
}
```

Como podemos ver en esta petición POST del LoginServlet, una vez autenticado el usuario, el método manda una redirección a otro servlet donde se hace una petición GET.

Como caso excepcional, si el usuario es erróneo, se vuelve otra vez a enviar el recurso del login.jsp, es seguro hacerlo porque con un usuario erróneo no se modifica nada.

Gestión de los usuarios

La sitio web permite a diferentes usuarios acceder a la apliación, estos pueden ver sus noticias y editarlas. Además se controla que un usuario no pueda modificar las noticias de otro.




```
private boolean storyBelongsToUser(Long newsOwnerID) {

    User user = (User) session.getAttribute("registredUser");
    return user.getId() == newsOwnerID;
}
```

Detalles de implementación

Doble filtro. No solo existe un filtro que comprueba que existe sesión para acceder a la parte privada de un usuario, también hay un filtro que comprueba que un usuario que ya ha iniciado sesión, no accede al servlet de login ni al de registro.

Loger customizado. En lugar de crear en cada servlet un atributo de tipo logger, he optado por crear una clase estática MyLogger que contiene un logger y un método para enviar un mensaje. A esta clase se puede acceder estaticamente desde cualquier lugar.

Página de error. En caso de acceder a una página que no existe, he creado una página 404 customizada para que se muestre, en lugar del error 404 por defecto de Tomcat.

Limitaciones de la propuesta

No se pueden votar las noticias por ahora. Es imposible controlar que un usuario no registrado vote muchas veces una noticia. Posteriormente se podrá hacer con el uso de cookies de navegación.

Por ahora todos los elementos se muestran en orden aleatorio. Antes de hacerlo quiero consultar al profesor de la asginatura sobre dónde es el sitio correcto para ordenar.

No se controla que las noticias tengan campos vacíos. Aunque la comprobación en el servidor pordría estar ya implementada en esta entrega, considero preferible implementarlo junto a las comprobaciones en javascript. De esta manera las comprobaciones en el servidor se podrán gestionar como excepciones generadas por casos anómalos de comportamiento.

Al pasar la suite de test, ocurre un error al ejecutar la segunda clase de tests en la línea

```
conn = DriverManager.getConnection("jdbc:hsqldb:mem:/localhost/news1","sa","");
```

Observando el logger, creo que ocurre porque no se destruye correctamente conn en la anterior ejecución.

Pruebas

Por ahora, aparte de las pruebas de navegación realizadas a mano, solo existe testing del DAO. Queda pendiente preguntar al profesor de la asignatura sobre como testear los servlets.

Autoevaluación

Tiempo

He intentado dejar el código lo más limpio posible para facilitar la lectura y que me sea sencillo seguir añadiendo cosas. La refactorización es lo que más tiempo me ha llevado.

También he empleado bastante tiempo al principio hasta que me ha quedado claro como funcionaba todo.

Puntos débiles

Creo que como único punto débil puedo decir que la aplicación es bastante básica. Hay cosas (pocas) que aún no funcionan y pueden dar errores (detalladas en Limitaciones de la propuesta) que serán arregladas en sucesivas entregas.

Puntos fuertes

La aplicación es muy sólida (es solo una opinión hasta que haya tests que lo demuesten).

El código esta muy bien modulado y es muy legible.