Marcos de Desarrollo

Sitio Web de Comentarios sobre Eventos Deportivos

Curso académico 2013-2014

1. Introducción

La práctica de Marcos de Desarrollo consiste en el diseño e implementación una aplicación Web basada en la utilización de las funcionalidades ofrecidas por la plataforma .NET. Esta aplicación consistirá en una sencilla web que dé soporte a la gestión de eventos deportivos y a la realización de comentarios sobre los mismos.

La práctica se evaluará de 0 a 10 puntos.

2. Funcionalidad básica

Este apartado especifica la funcionalidad básica correspondiente a la parte obligatoria de la práctica.

2.1 Modelo de información

El sitio Web permite visualizar en cada momento el conjunto de eventos deportivos sobre los que se pueden realizar consultas y comentarios.

De cada evento deportivo interesa almacenar su nombre (eg. Deportivo – Real Madrid), la fecha en la que tendrá lugar (día, mes, año, hora y minuto), la categoría a la que pertenece (eg. Fútbol, Baloncesto, Motor, etc.) y una pequeña reseña del mismo. En aras de la simplificación se asumirá un sistema de categorías plano, es decir, sin subcategorías.

La búsqueda de eventos, y la consulta de información asociada a los mismos, estarán accesibles para todos los usuarios sin necesidad de registro previo. Si un usuario desea realizar comentarios acerca de algún evento deportivo, deberá registrarse en el sitio Web especificando cierta información de registro.

La aplicación Web que se propone desarrollar consiste en un sitio que permite a múltiples usuarios intercambiar información acerca de eventos deportivos, que podría ser relevante de cara a las posibles apuestas sobre dichos eventos. Este sitio Web permitirá a los usuarios comentar aquellos eventos deportivos que todavía no hayan comenzado. Sólo los usuarios autenticados podrán añadir comentarios a los eventos deportivos.

De cara a facilitar el intercambio de información de interés, la aplicación permitirá la creación de grupos de usuarios con intereses comunes. Cualquier usuario autenticado podrá crear un nuevo grupo. Un usuario podrá formar parte de ninguno, uno o varios grupos y podrá recomendar eventos a cualquiera de los grupos a los que pertenece.

Un usuario se registra en la aplicación de comentarios sobre eventos deportivos especificando cierta **información de registro**, a saber, su seudónimo (login), contraseña, nombre, apellidos y dirección de correo electrónico. Opcionalmente podrá requerirle información relativa a su idioma y país para gestionar la internacionalización en base a las preferencias de usuario y no en base a la configuración del navegador.

2.2 Interacción con el usuario

A continuación, se detalla la funcionalidad que deberá ofrecer la aplicación al usuario:

1. **Registro de usuarios.** Debe permitir registrar nuevos usuarios, así como permitir actualizar la información de registro de los usuarios ya registrados.
2. **Autenticación y salida.** Un usuario se autenticará indicando su seudónimo y contraseña, con la posibilidad de recordar la contraseña para no tener que introducirla la siguiente vez. El usuario podrá salir explícitamente de la aplicación, lo que provoca que ya no se recuerde su contraseña, en caso de haber seleccionado dicha opción con anterioridad.
3. **Gestión de grupos de usuarios.** Un usuario autenticado podrá crear nuevos grupos. Un grupo consta de un nombre y una descripción. La aplicación dispondrá de un enlace “*Ver grupos*” que mostrará todos los grupos existentes en la aplicación, proporcionando para cada uno de ellos el número de miembros del grupo, el número de recomendaciones dirigidas al grupo y un enlace para que el usuario pueda darse de alta en el grupo. Si el usuario ya pertenece al grupo, no se mostrará el enlace “*Darse de alta*”. Cualquier usuario podrá ver los grupos existentes, pero únicamente los usuarios autenticados podrán darse de alta en un grupo. La aplicación dispondrá también de un enlace “*Mis grupos*” que mostrará los grupos a los que pertenece un usuario, proporcionando un enlace “*Darse de baja*” por cada grupo, que permitirá eliminar la suscripción del usuario a ese grupo.
4. **Búsqueda de eventos.** El usuario podrá buscar eventos deportivos por palabras clave del nombre del evento, especificando opcionalmente la categoría mediante un desplegable. Las palabras clave especificadas en la búsqueda tienen que estar todas contenidas en el nombre del evento como palabras o parte de palabras, sin distinguir entre mayúsculas y minúsculas, y en cualquier orden. Si el usuario especifica la categoría, la búsqueda se restringirá a los eventos de dicha categoría. El resultado de la búsqueda mostrará para cada evento: nombre, categoría, fecha, un enlace para añadir un nuevo comentario y un enlace para ver los comentarios existentes. Por último se incluirá un enlace para recomendar el evento a un grupo.
5. **Añadir comentario.** Un usuario puede añadir comentarios relativos a un evento. Si el usuario no estaba autenticado, cuando selecciona el enlace para añadir un comentario, se le redirige al formulario de autenticación, y tras autenticarse correctamente, se le muestra el formulario para añadir comentario.

6. **Ver comentarios de un evento.** A través del enlace para ver los comentarios de un evento se mostrarán, en una nueva página, los comentarios ordenados por la fecha de realización, apareciendo el más reciente primero (orden cronológico descendente). Por cada comentario se mostrará el seudónimo del usuario que lo realizó, la fecha en la que se insertó y el texto del comentario.
7. **Recomendar un evento.** Se mostrarán los grupos a los que pertenece el usuario y se le permitirá seleccionar aquellos a los que desea recomendar el evento. Cada recomendación puede llevar un texto asociado, justificando la recomendación.
8. **Mostrar recomendaciones.** La aplicación dispondrá de un enlace “*Mostrar recomendaciones*” que permitirá consultar las recomendaciones de eventos recibidas, en función de los grupos a los que pertenezca el usuario. El nombre del evento será un enlace para visualizar los comentarios asociados al mismo. Las recomendaciones se mostrarán ordenadas, de más reciente a más antigua.

Algunas opciones (como añadir un comentario) requieren que el usuario esté autenticado. Sin embargo, estas opciones deben aparecer disponibles al usuario no autenticado, con el objetivo de que conozca su existencia. La aplicación debe proporcionar una navegación fácil, permitiendo acceder de forma sencilla a las opciones disponibles y proporcionando *feedback* al usuario sobre si una operación ha sido realizada con éxito o no.

Para no alargar innecesariamente la práctica, **no** se implementará una aplicación de administración. En una situación real, la aplicación debería actualizar el conjunto de eventos o las categorías permitidas. Dado que no se implementará la aplicación de administración, la información sobre eventos y categorías se podrá introducir mediante sentencias INSERT INTO, que se incluirán como parte del script SQL de creación de tablas e inserción de datos.

2.3 Pruebas de integración de la capa modelo

Para verificar el correcto funcionamiento de la capa modelo, se utilizará el proyecto de Test que proporciona Visual Studio para implementar pruebas de integración automatizadas sobre los servicios de la capa modelo.

En una situación real, sería necesario implementar varios casos de prueba por cada método (caso de uso) de un servicio, para comprobar la correcta implementación de cada caso de uso en distintos escenarios. Sin embargo, para simplificar la realización de la práctica, bastará implementar un sólo caso de prueba por cada caso de uso, que compruebe su funcionamiento en un escenario “normal/usual” (sin errores, sin casos especiales, etc.).

2.4 Internacionalización

Con respecto a la internacionalización, se deberá utilizar el soporte de internacionalización de mensajes, de manera que el lenguaje de la interfaz gráfica no esté dentro del código de las aplicaciones. Sin embargo, no se pedirá que la aplicación esté disponible en varios idiomas, dado que esto alargaría innecesariamente el desarrollo de la

práctica. Tampoco se pedirá que la aplicación trate otros aspectos de internacionalización (e.g. fechas, cantidades monetarias, etc.).

3. Funcionalidad adicional: Etiquetado de comentarios

En la línea de la Web 2.0, el sitio Web de comentarios sobre eventos deportivos permitirá etiquetar los comentarios realizados por los usuarios, con la intención de mejorar su recuperación.

Mediante la realización de este apartado opcional, en el momento de añadir un comentario, el usuario podrá añadir una serie de etiquetas (de forma similar al etiquetado de un *post* en un *blog*). El usuario podrá utilizar etiquetas ya existentes o crear etiquetas nuevas en el momento de realizar el comentario. Ejemplos de etiquetas podrían ser: “última hora”, “opinión personal”, “fichaje”, “lesión”, etc.

La aplicación deberá controlar la presencia de etiquetas duplicadas, impidiendo la creación de etiquetas que ya existen. No será necesario tener en cuenta la eliminación de las etiquetas existentes en la aplicación. Sin embargo, en el caso de los comentarios ya etiquetados, la aplicación permitirá modificar el etiquetado, es decir, quitar una o varias de sus etiquetas y también añadir nuevas etiquetas sobre ellos.

A partir de las etiquetas se creará una “nube de etiquetas”, que estará visible en todas las páginas del sitio Web. Cada etiqueta de la nube será un enlace para mostrar los comentarios asociados a ella. Se valorará positivamente que el tamaño de fuente de cada etiqueta en la nube sea dependiente del número de comentarios etiquetados por ella, de tal manera que las etiquetas que más se repiten se muestren en tamaños de fuente mayores. Todas las etiquetas se mostrarán en minúsculas.

4. Funcionalidad adicional: AJAX

Con un enfoque “clásico”, cada vez que el usuario hace clic en un enlace o envía un formulario, la aplicación Web devuelve una página HTML completa. En ocasiones, este modelo puede resultar poco interactivo. Con el enfoque “AJAX” (“Asynchronous JavaScript + XML”) [1], la respuesta de la aplicación a una interacción no es una página completa, sino sólo el contenido correspondiente a la parte de la página que es necesario actualizar (o el nuevo fragmento de HTML correspondiente a esa parte).

Bajo el enfoque AJAX, un código JavaScript en el navegador envía una petición HTTP cuando el usuario realiza una interacción. La aplicación Web procesará la petición invocando el correspondiente caso de uso en la capa modelo, y a diferencia de una interacción normal, devolverá la información que es necesario actualizar en la página (o el nuevo fragmento de HTML). El código JavaScript en la página que visualiza el navegador recibirá la respuesta y modificará el árbol DOM de la página para actualizar la información necesaria.

En esta parte opcional se debe analizar la interfaz de la aplicación y elegir al menos un caso de uso en el que podría ser interesante aplicar el enfoque AJAX. Para la implementación podrá emplearse AJAX Control Toolkit [2] o cualquier otra herramienta.

5. Normativa y evaluación

5.1 Composición de los grupos

La práctica se realizará en grupos de 3 personas.

5.2 Estándar de codificación

Con la finalidad de escribir código de calidad y fácilmente legible, se seguirá un sistema de codificación común, que define reglas para nombrar clases, atributos y métodos, normas de indentación, etc. Esto permite que en un equipo de desarrollo el aspecto del código sea el mismo, independientemente de qué programador lo haya escrito, lo que facilita el mantenimiento. Para la práctica se utilizará el estándar de codificación C# [3][4]. Los ejemplos de la asignatura siguen estas sencillas convenciones de nombrado. Para no alargar la práctica, no será necesario documentar las clases. Sin embargo, sí será obligatorio documentar las excepciones que lanza cada método.

5.3 Formato de entrega de la versión final

La distribución fuente de la aplicación se entregará con el propio sistema de control de versiones (Subversion) que se utilizará para el desarrollo de la práctica. Cuando se aproxime la fecha tope de cada entrega, se darán instrucciones detalladas.

A continuación se detalla el formato de la memoria. **Como norma general, los diagramas emplearán la notación UML. Los diagramas deben ser de calidad (y no hechos por reingeniería inversa; ¡el diseño precede a la implementación!) y estar explicados de manera breve, pero clara. Es importante destacar que no se pretende que se haga un documento grande, sino un documento que explique breve, pero claramente, cada uno de los apartados que a continuación se describen. La calidad de la memoria será vital para la corrección de la práctica.**

1. Arquitectura global

Comentar la estructura global de paquetes de la aplicación (no es necesario emplear la notación UML para este aspecto particular). No se deben mostrar todos los paquetes, sino sólo los paquetes de más alto nivel que reflejen la arquitectura de la aplicación y sus principales subpaquetes. Por cada paquete se explicará brevemente qué contiene.

2. Modelo

2.1 Clases persistentes

Muestra todas las clases persistentes en uno o varios diagramas de clases y las relaciones relevantes entre ellas.

2.2 Interfaces de los servicios ofrecidos por el modelo

Explica el criterio que se ha seguido para agrupar los casos de uso en servicios (fachadas). Incluye un diagrama de clases para cada servicio, que muestra la

interfaz del servicio (con la declaración completa de sus operaciones) y los tipos de datos que utilizan las operaciones.

2.3 Diseño de un DAO

Se incluirá un diagrama de clases que ilustre la implementación de uno de los DAOs de la aplicación. No es necesario mostrar el diseño del resto de DAOs, dado que su arquitectura es similar.

2.4 Diseño de un servicio del modelo

Se incluirá un diagrama de clases que ilustre la implementación de un servicio del modelo. También se incluirá un diagrama de secuencia que muestre la ejecución de un caso de uso en la capa modelo.

No es necesario mostrar el diseño del resto de servicios, dado que su arquitectura es similar.

2.5 Otros aspectos

Si es necesario, se documentarán aspectos particulares que se consideren relevantes para la correcta comprensión de la capa modelo.

3. Interfaz gráfica

Se mostrará un diagrama de secuencia que ilustre el procesamiento de un caso de uso representativo. El diagrama mostrará las principales clases o artefactos que intervienen en la ejecución del caso de uso. En lo que respecta al modelo, no es necesario detallar el procesamiento de la petición en el interior del servicio (dado que ya se hizo en el apartado 2.4).

Si es necesario, también se documentarán aspectos particulares que se consideren relevantes para la correcta comprensión de la interfaz gráfica.

4. Un apartado para la parte adicional

*Se explicará **claramente** cómo se ha abordado el diseño de esta parte.*

5. Compilación e instalación de la aplicación

Se explicará claramente cómo compilar e instalar la aplicación, asumiendo un entorno correctamente configurado (e.g. base de datos instalada, etc.), y que en consecuencia, no es preciso documentar. Las instrucciones de instalación deberían ser lo más sencillas posibles.

6. Problemas conocidos

Lista los errores que se conoce que tiene el código.

5.4 Iteraciones y entregas

Para la realización de la aplicación se seguirá un enfoque basado en iteraciones, de manera que cada iteración incorpora más funcionalidad sobre la anterior, hasta que en la última iteración se termina con un software que implementa toda la funcionalidad. En particular, el desarrollo de la aplicación se hará en dos iteraciones:

- **Primera iteración.** Se implementará la capa modelo y las pruebas de integración especificadas en el apartado 2.3. Además, si se desea hacer las partes adicionales especificadas en los apartados 3 y 4, se deberá abordar en esta iteración la parte correspondiente al modelo. Se incluirán pruebas de integración para los servicios ofrecidos por la capa modelo.
- **Segunda iteración.** Se corregirán los errores detectados en la primera iteración y, si es necesario, se actualizarán las pruebas de integración de la capa modelo. Se implementará la capa Web. Además se redactará la memoria. Si se ha decidido hacer alguna de las partes adicionales, deberá implementarse en esta iteración la parte correspondiente a la capa Web.
- **Se informará en las clases de la materia y a través de la página web de la materia (<http://sabia.tic.udc.es/docencia/mad>) con suficiente antelación de la fecha concreta de entrega de cada iteración. (Fecha tentativa entrega primera iteración principios/mediados Noviembre)**

A efectos de planificación debe tenerse en cuenta que cada iteración debe estar lista para el primer día posible de su plazo de entrega.

La entrega de cada iteración es obligatoria y deben estar presentes todos los miembros del grupo. En la entrega se realizará una demo y se harán preguntas individualizadas sobre el diseño y la implementación. **La corrección de la primera iteración no llevará una nota asociada ni será necesario entregar una memoria.** Bastará con mostrar los diagramas correspondientes al apartado 2 de la memoria mediante una herramienta para la elaboración de diagramas UML. También se considerarán válidos los diagramas generados con Visual Studio, completándolos adecuadamente cuando sea necesario. El objetivo de la corrección de esta primera iteración es intentar detectar errores importantes, y en ese caso, orientar al alumno hacia su resolución. **En la corrección de la segunda iteración se pondrá una nota y se deberá entregar la memoria impresa.**

Para la realización de la práctica puede reusarse todo el código (en formato binario o fuente) que se desee de los ejemplos distribuidos en la asignatura, y de hecho, se recomienda hacerlo, dado que facilitará notablemente el comienzo de la práctica. También es importante resaltar que el código utilizado debe comprenderse y no usarlo “ciegamente”.

5.5 Evaluación

La nota de la práctica se calculará como sigue:

- Parte básica/obligatoria. Puntuación: **7 puntos**.
- Parte opcional “Etiquetado de comentarios”. Puntuación: **1.75 puntos**.
- Parte opcional “AJAX” (opcional). Puntuación: **1.25 puntos**.

En todos los casos, para la puntuación de cada parte se tendrá en cuenta:

- Su correcto funcionamiento.
- La calidad del diseño.
- La calidad del código.
- La calidad de la memoria.

Una práctica copiada significará un suspenso para el grupo que ha dejado copiar y el que ha copiado; a todos los efectos, no se hará ninguna distinción. Los suspensos por práctica copiada tendrán que realizar una práctica distinta, que además deberán proponer (y ser aceptada).

6. Referencias

[1] Ajax: A New Approach to Web Applications, <http://www.adaptivepath.com/ideas/ajaxnew-approach-web-applications>

[2] Ajax Control Toolkit. <http://www.asp.net/ajaxlibrary/act.ashx>

[3] Scott Bellware, *C# Code Style Guide*, <http://www.sourceformat.com/pdf/cs-coding-standard-bellware.pdf>

[4] C# Coding Conventions (C# Programming Guide) [http://msdn.microsoft.com/en-us/library/vstudio/ff926074\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/vstudio/ff926074(v=vs.100).aspx)