

Marcos de Desarrollo

Photogram: red Social de compartición de fotografías

Curso académico 2020.21

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 una red social para la compartición de fotografías y realización de comentarios sobre las mismas.

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 permitirá al usuario compartir sus imágenes, sobre las que el resto de usuarios podrá realizar consultas y comentario.

De cada imagen interesará almacenar un título, una breve descripción, la fecha de subida, la información exif de la misma si se conoce (Apertura de diafragma (f), tiempo de exposición (t), ISO, balance de blancos (wb)) y la categoría a la que pertenece (e.g. Retrato, Paisaje nocturno, Paisaje nocturno, etc.). En aras de la simplificación se asumirá un sistema de categorías plano, es decir, sin subcategorías y que cada imagen sólo pueda pertenecer a una de estas categorías.

La búsqueda de imágenes, y la consulta de información asociada a las mismas, estarán accesibles para todos los usuarios sin necesidad de registro previo. Si un usuario desea realizar comentarios acerca de alguna imagen, o indicar un “me gusta” a alguna de ellas, deberá registrarse en el sitio Web especificando cierta información de registro.

Este sitio Web permitirá a los usuarios comentar todas aquellas imágenes subidas a la red social. Asimismo permitirá que los usuarios seleccionen las cuentas de otros usuarios que deseen seguir para estar informados de las imágenes que suban a la red social.

Un usuario se registrará en la aplicación 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. La página principal a la que se redirija tras la autenticación permitirá acceder a las opciones de búsqueda de imágenes y mostrará el perfil del usuario actual. Posibilitará también un enlace para la subida de nuevas imágenes.
3. **Visualización de usuarios.** La aplicación facilitará un perfil para cada usuario. En ese perfil se mostrarán sus últimas imágenes subidas (e.g. 3), un enlace “*Ver Seguidos*” que mostrará todos los usuarios a los que ese usuario sigue en un momento dado y un enlace de “*Ver Seguidores*” que mostrará el listado de cuentas que están siguiendo al usuario. Tanto los nombres de los usuarios seguidos como de los seguidores mostrados serán un enlace a su perfil de usuario. Para acceder de manera directa al perfil del usuario con la sesión iniciada, se proporcionará un enlace de “*Mi Perfil*”.
4. **Seguimiento de usuarios.** Un usuario autenticado podrá seguir a otros usuarios con el objetivo de estar al tanto de las imágenes que estos suban a la aplicación. Para ello, el perfil de cada usuario tendrá una opción de “*Seguir Usuario*” para que el usuario con la sesión iniciada pueda convertirse en su seguidor.
5. **Subida de imágenes.** La aplicación facilitará la subida de imágenes junto con sus datos descriptivos (título, descripción...). La imagen podrá almacenarse internamente en la base de datos o bien guardarse en una estructura de directorios del sistema (en cuyo caso se almacenará en la base de datos únicamente el nombre del fichero al que deberá hacerse referencia posteriormente para facilitar su visualización).
6. **Búsqueda de imágenes.** El usuario podrá buscar imágenes por palabras clave del título de la imagen y/o de su descripción, restringiendo opcionalmente la categoría de la búsqueda mediante un desplegable. Las palabras clave especificadas en la búsqueda tienen que estar todas contenidas en el título o en la descripción de la imagen como palabras o parte de palabras, sin distinguir entre mayúsculas y minúsculas, y en cualquier orden. Opcionalmente, el usuario podrá especificar una categoría, en cuyo caso, la búsqueda se restringirá a las imágenes de dicha categoría. El resultado de la búsqueda mostrará para cada imagen: la imagen propiamente dicha (o una miniatura que enlace a la imagen original), el título, los datos exif, la categoría, fecha, el autor (que será un enlace a su perfil de usuario), un enlace para añadir un nuevo comentario, un enlace para ver los comentarios existentes (si los hubiese) y un enlace para indicar “*Me gusta*” sobre la imagen.

Se indicará también el número de “Me gusta” de la imagen, pero no será necesario poder visualizar de quién es la autoría de cada uno de los “Me gusta” de la imagen.

7. **Indicar “Me gusta”.** Un usuario puede indicar que le gusta una imagen. 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 su “Me gusta”. No será posible indicar “Me gusta” más de una vez sobre la misma imagen.
8. **Añadir comentario.** Un usuario puede añadir comentarios relativos a una imagen. 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. Un usuario también podrá modificar o eliminar los comentarios realizados por él mismo.
9. **Ver comentarios de una imagen.** A través del enlace para ver los comentarios de una imagen 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 (que nuevamente será un enlace a su perfil), la fecha en la que se insertó y el texto del comentario. Esta opción aparecerá disponible únicamente cuando una imagen tenga comentarios asociados.

Algunas opciones (como subir una imagen; añadir un comentario; seguir a un usuario...) 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 búsqueda de imágenes deberá ser de acceso público. 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, por ejemplo, gestionar las categorías permitidas. Dado que no se implementará la aplicación de administración, la información 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 imágenes

En la línea de la Web 2.0, el sitio Web de compartición de fotografías permitirá etiquetar las imágenes subidas por los usuarios, con la intención de mejorar su recuperación.

Mediante la realización de este apartado opcional, en el momento de añadir una imagen a la galería, el usuario podrá añadir una serie de etiquetas. El usuario podrá utilizar etiquetas ya existentes o crear etiquetas nuevas en el momento de subir la imagen. Ejemplos de etiquetas podrían ser: “B&W”, “Fine Art”, “Larga Exposición”, “A Coruña”, “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 las imágenes ya etiquetadas, la aplicación permitirá a su autor modificar el etiquetado, es decir, quitar una o varias de sus etiquetas y también añadir nuevas etiquetas sobre ellas.

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 las imágenes asociadas 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: Cacheado de búsquedas

Habitualmente las búsquedas que realizan los usuarios de una aplicación suelen ser repetitivas. Una manera de reducir el tiempo de respuesta se basa en el empleo de *cachés* que almacenen los resultados a las consultas más habituales de manera que no sea necesario acudir a la base de datos ante cada petición (o al menos reducir las llamadas).

En este parte opcional se propone el empleo de las clases incluidas en el espacio de nombres `System.Runtime.Caching`, aunque puede emplearse cualquier otro paquete o librería (`Caching Application Block` incluido en la `Enterprise Library 5.0`, etc.). En aras de una mayor simplicidad, no se requerirá un estudio dinámico de las consultas más habituales sino que se mantendrán en caché las 5 últimas consultas realizadas por los usuarios de la aplicación. Cuando un usuario realice una búsqueda incluida en la caché, será ésta la que proporcione los resultados de las imágenes que cumplan los criterios especificados.

Aunque en un caso real no sería así, no será necesario comprobar si se han insertado nuevas imágenes en la base de datos desde la creación de la entrada en la caché. Sin

embargo, cuando se soliciten los comentarios de alguna de las imágenes incluidas en el resultado de la búsqueda, sí será necesario obtener la lista actualizada de los mismos.

5. Funcionalidad adicional: OAuth 2.0

Cada vez son más las aplicaciones que permiten al usuario autenticarse en ellas mediante una delegación en algún servicio externo, por ejemplo, Google o Facebook. Este permite que el usuario tenga que memorizar un número menor de contraseñas a la vez que dota a la aplicación de una mayor *confiabilidad* al delegar el procesamiento de las credenciales en una entidad externa por lo general de reconocido prestigio.

Esta autenticación se realiza tras el intercambio de una serie de *tokens* (*Access_token*, *refresh_token*, ...) que impiden la usurpación de la misma.

Bajo este enfoque, se plantea el uso de la API de Auth2.0 [2] de servicios como Google, Facebook, Amazon, Live, etc. para su integración en la aplicación Web Desarrollada.

En esta parte opcional se debe desarrollar el acceso a la aplicación web mediante el uso del API OAuth 2 de alguno de los proveedores mencionados y detallar en la memoria el flujo de intercambio de información que posibilita la autenticación.

6. Normativa y evaluación

6.1 Composición de los grupos

La práctica se realizará en grupos de 3 personas. Será necesario consensuar con los profesores de la materia cualquier cambio a este respecto.

6.2 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.

6.3 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 de la memoria).

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.

6.4 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 alguna de las partes adicionales especificadas, 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 plataforma Moodle con suficiente antelación de la fecha concreta de entrega de cada iteración. (Fecha tentativa entrega primera iteración: primeros/mediados Abril; Fecha tentativa de entrega segunda iteración: Final cuatrimestre)**

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 deberá entregar la memoria impresa. Esta corrección será la única que llevará asociada una calificación.**

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 ser usado “ciegamente”.

6.5 Evaluación

La nota de la práctica se calculará como sigue:

- Parte básica/obligatoria. Puntuación: **7 puntos.**
- Parte opcional “Etiquetado de Imágenes”: **1,5 puntos**
- Parte opcional “Cacheado de búsquedas”: **0,75 puntos**
- Parte opcional “OAuth”: **0,75 puntos**

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

- Su correcto funcionamiento y la claridad de las explicaciones ofrecidas durante defensa de la práctica.
- La calidad del diseño.
- La calidad del código.
- La calidad de la memoria.

Las notas de las aplicaciones aprobadas se conservan hasta la convocatoria de segunda oportunidad (o convocatoria Extraordinaria de Diciembre)

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).

7. Referencias

- [1] Ajax: A New Approach to Web Applications, <http://www.adaptivepath.com/ideas/ajaxnew-approach-web-applications>
- [2] OAuth 2.0. <https://oauth.net/2/>
- [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)