



2014

Documentación externa.

Diseño de Software

El proyecto consiste en una aplicación Web para fotógrafos, en donde ellos puedan mostrar sus fotografías, ver productos, y tener un foro con consejos y opiniones de otros usuarios.

Estefany Quesada
Ricardo Pérez
Tatiana Zúñiga

Instituto Tecnológico de Costa Rica
Administración de Tecnologías de Información

27/05/2014

Índice

Resumen ejecutivo	2
Especificación de requerimientos	
Autenticar miembros.....	3
Agregar Fotos.....	5
Eliminar Fotos.....	6
Otorgar permisos.....	7
Crear publicación.....	7
Eliminar publicación.....	8
Comentar.....	8
Comentar Fotos.....	9
Crear álbum.....	10
Ver productos.....	11
Prioridades	12
Descripción de diseño de alto nivel	
Diagrama de arquitectura conceptual.....	13
Diagrama de Paquetes.....	14
Diagrama de Componentes.....	15
Diagrama de Despliegue.....	16
Diagrama de Clases.....	17
Justificación de patrones utilizados	18
Interacción con sistemas externos	23
Dominio y servicios activos	26
Conclusiones	26
Bibliografía	27

Resumen ejecutivo

Ser fotógrafo y querer enseñar su trabajo es todo uno. Tanto si se es profesional o aficionado, se desea que otros conozcan las obras que has creado, ya sea para vender sus servicios, o ya sea por el puro placer de compartir su trabajo. Esa es la razón por la que antes o después, todo fotógrafo se plantea tener su propia web o algún lugar donde pueda publicar su trabajo.

Sin embargo, no siempre es fácil contar con un lugar en el que se pueda enseñar sus fotografías y sirva para transmitir una imagen de un fotógrafo profesional y comprometido con su trabajo, o interactuar con personas de la industria fotográfica y recibir consejos o incluso conocer lo nuevo.

Con esas consideraciones en mente, se decide crear una aplicación web para fotógrafos con el fin de alojar y mostrar sus trabajos, tener acceso a fotografías de calidad de otros usuarios, sean estos profesionales o no, así como información sobre los aportes de los usuarios, esto con el fin de dar a conocer sus creaciones entre fotógrafos y público en general, y además brindar un lugar con la posibilidad de compartir información del campo, aconsejar a los menos experimentados, brindar tips, indicar tendencias, conocer las últimas opciones de cámaras que salen al mercado, entre otros.

Este documento muestra el análisis de diseño de la aplicación web uPhoto, el cual contiene diversas secciones que explican lo que el sistema puede hacer, y cómo puede éste puede ser construido.

Especificación de requerimientos

A la hora de realizar este proyecto se deben cumplir con los requerimientos que los clientes necesitan, en este caso deben evaluar las necesidades de los administradores de la página, y a su vez, las especificaciones por medio de fotografías por ejemplo. A continuación se detallarán por medio de diferentes casos de uso, los requerimientos específicos del sistema.

Nombre: Autenticar Miembros	
Actor: Usuario	
Descripción:	Determina como el actor con el perfil "Usuario general" accederá en el sistema, tras acceder al sistema el actor obtiene para sí mismo un nuevo rol con los permisos correspondientes para el manejo del sistema de acuerdo al rol obtenido, así como el respectivo cierre de sesión
Precondiciones:	<ul style="list-style-type: none">→ No existen precondiciones de inicio para inicio de sesión→ En caso de querer cerrar la sesión el actor debe tener una sesión iniciada y tener habilitado la opción "Cierre de sesión".
Flujo normal:	<ol style="list-style-type: none">1. El actor inicia la aplicación.2. La aplicación inicia mostrando a ventana de presentación.3. La aplicación muestra la ventana de inicio de sesión4. El actor ingresa los elementos de inicio de sesión<ul style="list-style-type: none">Elemento 1 Usuario de sistemaElemento 2 Contraseña asignada5. El actor selecciona la opción de "Ingresar al sistema".6. El sistema verifica la existencia de dicho actor en la base de datos del sistema.7. El sistema establece los permisos que posee el actor de acuerdo al nombre de usuario8. El sistema habilita los iconos de "Cierre de sesión" y de acciones que puede realizar el actor de acuerdo a los permisos que posee.9. El caso de uso finaliza.
Flujos alternativos:	<ol style="list-style-type: none">1. El actor selecciona la opción de "Ingresar al sistema" sin colocar todos los datos en el sistema de autenticación.<ol style="list-style-type: none">1) En el punto 4 de la sección "Autenticación en el sistema de control de activos" el actor omite colocar alguno o ambos elementos requeridos para el acceso (a Usuario del sistema, b Contraseña del sistema.)2) El actor en el punto 5 de la sección "Autenticación en el sistema de control de activos" selecciona la opción "Ingresar al sistema"3) El sistema verifica que hace falta un dato de autenticación.4) El sistema muestra un mensaje de advertencia5) Los campos de inicio de sesión (a Usuario del sistema, b Contraseña del sistema.) pasan a estar en blanco (se borran todos los caracteres que posean)6) El caso de uso termina.

2) El actor selecciona la opción de “Cancelar” en el sistema de autenticación.

- 1) El actor en el punto 5 de la sección “Autenticación en el sistema de control de activos” selecciona la opción “Cancelar”
- 2) El formulario de acceso al sistema se cierra.
- 3) El sistema habilita la opción de inicio de sesión y de envío de errores del sistema pero mantiene deshabilitado todas las opciones adicionales de gestión de la aplicación.
- 4) El caso de uso termina.

3) El actor selecciona la opción “Ingresar al sistema” introduciendo datos erróneos en el sistema de autenticación.

- 1) En el punto 4 de la sección “Autenticación en el sistema de control de activos” el actor coloca alguno o ambos elementos requeridos para el acceso (a Usuario del sistema, b Contraseña del sistema.) con información errónea o incorrecta.
- 2) El actor en el punto 5 de la sección “Autenticación en el sistema de control de activos” selecciona la opción “Ingresar al sistema”
- 3) El sistema verifica que un dato de autenticación es erróneo.
- 4) El sistema muestra un mensaje de error
- 5) Los campos de inicio de sesión (a Usuario del sistema, b Contraseña del sistema.) pasan a estar en blanco (se borran todos los caracteres que posean)
- 6) El caso de uso termina.

4) El actor intenta acceder al sistema con una sesión “ya iniciada”.

- 1) En el punto 4 de la sección “Autenticación en el sistema de control de activos” el actor coloca alguno o ambos elementos requeridos para el acceso (a Usuario del sistema, b Contraseña del sistema.) con información correcta.
- 2) El actor en el punto 5 de la sección “Autenticación en el sistema de control de activos” selecciona la opción “Ingresar al sistema”
- 3) El sistema verifica que el actor existe en el sistema.
- 4) El sistema determina que dicho actor ya se encuentra con una sesión abierta.
- 5) El sistema muestra un mensaje de advertencia
- 6) Los campos de inicio de sesión (a Usuario del sistema, b Contraseña del sistema.) pasan a estar en blanco (se borran todos los caracteres que posean)
- 7) El caso de uso termina.

Nombre: Agregar fotos Actor: Usuario	
Descripción:	Se permitirá a los miembros con los permisos adecuados (fotógrafos), agregar nuevas fotos
Precondiciones:	→ El usuario debe estar registrado como fotógrafo
Flujo normal:	<ol style="list-style-type: none"> 1. El actor selecciona la opción de “Agregar una foto” 2. El sistema muestra un formulario con la información necesaria para ingresar una foto a la aplicación 3. El actor llena los espacios de información del activo que desea ingresar en el sistema. 4. El actor selecciona la opción de “Agregar foto” para el envío del formulario al sistema para su validación. 5. El sistema verifica que la información de todo el formulario esté completa. 6. El sistema ingresa una nueva foto dentro de su sistema de almacenamiento. 7. El sistema envía un mensaje al actor confirmando la transacción. <ol style="list-style-type: none"> 1. Título: “Ingreso de información al sistema de activos.” 2. Enunciado: “Se ha realizado exitosamente el ingreso de un activo en el sistema.” 8. El actor cierra la ventana de mensaje. 9. El actor cierra el formulario. 10. El caso de uso finaliza.
Flujos alternativos:	<p>1.1 El actor selecciona la opción “Agregar foto” mientras el formulario está incompleto.</p> <ol style="list-style-type: none"> 1) En el punto 3 del caso de uso en la sección “agregar una foto” el actor no introduce información que es requerida por el sistema para el ingreso de la misma 2) El sistema verifica el formulario y descubre que dicho formulario requiere información adicional. 3) El sistema muestra un mensaje de advertencia 4) Los espacios de información del formulario se limpian de la información que contienen. 5) El caso de uso finaliza. <p>1.2 El actor selecciona la opción “Agregar foto” mientras el formulario posee información errónea.</p> <ol style="list-style-type: none"> 1) En el punto 3 del caso de uso en la sección “agregar una foto” el actor introduce información que es errónea para el ingreso de la misma. 2) El sistema verifica que la información de todo el formulario está completa. 3) El actor descubre un error en la información que se piensa introducir. 4) El actor selecciona la opción “Cancelar” en la ventana de verificación. 5) Los espacios de información del formulario se limpian de la información que contienen. 6) El caso de uso finaliza.

Nombre: Eliminar fotos Actor: Usuario	
Descripción:	Se permitirá a los miembros con los permisos adecuados (fotógrafos), puedan eliminar las fotos publicadas por ellos.
Precondiciones:	→ El usuario debe estar registrado como fotógrafo
Flujo normal:	<ol style="list-style-type: none"> 1. El actor ingresa al sistema. 2. El actor ingresa a las publicaciones de fotografías en su perfil 3. El actor selecciona la foto que desea eliminar. 4. El actor selecciona la opción de "Eliminar foto". 5. El sistema despliega un cuadro consultando si está seguro que desea eliminar la fotografía. 6. El actor selecciona Ok. 7. El sistema elimina la foto de su sistema de almacenamiento. 8. El sistema envía un mensaje al actor confirmando la eliminación. 9. El caso de uso finaliza.
Flujos alternativos:	<p>6.1 En el cuadro de comprobación el actor selecciona la opción "Cancelar".</p> <ol style="list-style-type: none"> 1) El usuario selecciona la opción cancelar, la foto no será eliminada 2) El caso regresa al punto 3 y continua con el flujo normal.

Nombre: Otorgar permisos	
Actor: Usuario	
Descripción:	Se permitirá que los actores con los permisos adecuados puedan otorgar permisos a actores del sistema.
Precondiciones:	→ El usuario debe estar registrado como administrador
Flujo normal:	<ol style="list-style-type: none"> 1. El actor selecciona la opción "Otorgar permisos". 2. El sistema presenta un formulario con los datos que se deben llenar. 3. El actor llena el formulario con la información pertinente. 4. El actor selecciona la opción "Enviar autorización". 5. El sistema recibe la autorización. 6. El sistema procesa la autorización y la almacena en el sistema de permisos. 7. El actor cierra el formulario. 8. El caso de uso finaliza.
Flujos alternativos:	<p>1.1 El actor selecciona la opción "Cancelar" en el formulario de envío del reporte.</p> <ol style="list-style-type: none"> 1) En el punto 4 del flujo básico del caso de uso en la sección "Otorgar permisos a usuarios" el actor selecciona la opción "Cancelar". 2) El formulario con la información que se le solicita al actor se limpia. 3) El caso de uso finaliza.

Nombre: Crear publicación	
Actor: Usuario	
Descripción:	Se permitirá a los miembros con los permisos adecuados, agregar publicaciones a la página
Precondiciones:	→ El actor debe estar registrado como usuario en la aplicación
Flujo normal:	<ol style="list-style-type: none"> 1. El actor ingresa a la página principal de la aplicación. 2. El actor selecciona la opción referente a crear una nueva publicación 3. El sistema habilita la opción de "Publicar" 4. El actor escribe la publicación 5. El actor selecciona la opción de publicar 6. La publicación se realiza 7. El caso de uso finaliza.
Flujos alternativos:	<p>6.1. Si el actor selecciona la opción cancelar.</p> <ol style="list-style-type: none"> 1) El sistema no realiza la publicación 2) Los espacios quedarán en blanco

Nombre: Eliminar publicación Actor: Usuario	
Descripción:	Se permitirá a los miembros con los permisos adecuados, eliminar publicaciones a la página
Precondiciones:	→ El actor debe estar registrado como usuario en la aplicación
Flujo normal:	<ol style="list-style-type: none"> 1. El actor ingresa a la página principal de la aplicación. 2. El actor selecciona la opción referente a eliminar una nueva publicación 3. El sistema habilita la opción de “Eliminar” 4. La página envía un mensaje para verificar que realmente se desea eliminar 5. La publicación se elimina 6. El caso de uso finaliza.
Flujos alternativos:	4.1. Si el actor selecciona la opción cancelar. <ol style="list-style-type: none"> 1) El sistema no elimina la publicación 2) El sistema vuelve a presentar las publicaciones y la opción “Eliminar”

Nombre: Comentar Actor: Usuario	
Descripción:	El usuario podrá publicar un comentario en una publicación establecida
Precondiciones:	→ El actor debe estar registrado como usuario en la aplicación
Flujo normal:	<ol style="list-style-type: none"> 1. El actor ingresa a la página principal de la aplicación. 2. El actor selecciona la opción referente a publicaciones 3. El actor busca la publicación deseada 4. El actor elige la publicación deseada 5. El actor elige la opción responder 6. El actor escribe el comentario 7. El actor elige la opción publicar 8. El caso de uso finaliza.
Flujos alternativos:	7.1. Si el actor selecciona la opción cancelar. <ol style="list-style-type: none"> 1) El sistema no realiza la publicación 2) El sistema vuelve a presentar las publicaciones

Nombre: Comentar fotos Actor: Usuario	
Descripción:	Se permitirá comentar fotos a todos los miembros
Precondiciones:	→ El usuario debe estar registrado como usuario de la aplicación
Flujo normal:	<ol style="list-style-type: none"> 1. El actor ingresa al sistema 2. El actor selecciona la foto que desea comentar de la página principal. 3. El sistema despliega en grande la fotografía y debajo de ella un espacio para ingresar el comentario. 4. El actor ingresa el comentario y le da clic al botón “comentar”. 5. El sistema realiza el comentario y lo deja visible a los usuarios. 6. El caso de uso finaliza.
Flujos alternativos:	<p>2.1 El actor busca un perfil en particular para comentar sus fotos.</p> <ol style="list-style-type: none"> 1. El sistema tiene un espacio para realizar búsquedas 2. El actor ingresa el nombre que desea ver el perfil 3. El sistema despliega los álbumes del fotógrafo buscado 4. El actor selecciona un álbum 5. El sistema despliega las fotos de dicho álbum 6. El actor selecciona la foto que desea comentar 7. El caso regresa al punto 3 del flujo normal. <p>4.1 El actor no le da clic al botón “comentar” y sale de la foto</p> <ol style="list-style-type: none"> 1. El actor cierra la fotografía que iba a comentar sin terminar la publicación del comentario 2. El sistema no guarda el comentario y lo devuelve a la página de fotografías 3. El caso regresa al punto 2 del flujo normal.

Nombre: Crear álbum Actor: Usuario	
Descripción:	Se permitirá crear álbumes a los miembros con los permisos adecuados (fotógrafos).
Precondiciones:	→ El usuario debe estar registrado como usuario fotógrafo. → El usuario debe tener una cuenta en Flickr
Flujo normal:	<ol style="list-style-type: none"> 1. El actor ingresa al sistema 2. El actor selecciona la opción crear álbum 3. El sistema despliega una ventana para ingresar el nombre del álbum 4. El actor ingresa el nombre del álbum a crear 5. El sistema en la ventana desplegada anteriormente muestra 2 opciones: <ul style="list-style-type: none"> • Agregar fotos desde Flickr • Subir fotos. 6. El actor selecciona la opción deseada. 7. El actor selecciona las fotos que desea subir. 8. El actor selecciona el botón "Subir" 9. El sistema muestra las fotos cargadas en la aplicación. 10. El caso de uso finaliza.
Flujos alternativos:	<p>4.1 El actor selecciona Agregar fotos desde Flickr.</p> <ol style="list-style-type: none"> 1. El sistema despliega una ventana de login en Flickr. 2. El usuario ingresa sus datos de login en Flickr. 3. El sistema despliega una ventana en donde se encuentran las fotos del usuario almacenadas en Flickr. 4. El caso regresa al punto 7 del flujo normal. <p>4.2 El actor selecciona Subir fotos.</p> <ol style="list-style-type: none"> 1. El sistema despliega una ventana de examinar, para buscar la ubicación de la foto a subir. 2. El caso regresa al punto 7 del flujo normal. <p>6.1 El actor no selecciona el botón "Subir"</p> <ol style="list-style-type: none"> 1. El actor cierra la ventana en donde tiene las fotos seleccionadas para subir sin dar clic en "Subir". 2. El sistema vuelve a la página principal 3. El caso regresa al punto 2 del flujo normal, o finaliza.

Nombre: Ver productos	
Actor: Usuario	
Descripción:	Todos los usuarios podrán ver productos de cámaras publicadas en Amazon
Precondiciones:	→ El actor debe estar registrado como usuario en la aplicación
Flujo normal:	<ol style="list-style-type: none"> 1) El actor ingresa a la página principal de la aplicación. 2) El actor selecciona la opción ver productos. 3) El sistema conecta con la página de Amazon y despliega una ventana con los productos a la venta relacionados con cámaras. 4) El actor selecciona el producto que quiere ver. 5) El sistema despliega la información del producto seleccionado. 6) El actor decide comprar (dicha compra es con Amazon no con nuestra aplicación). 7) El caso de uso finaliza.
Flujos alternativos:	No hay

Prioridades

Para el proceso de priorización, se toman en cuenta dos campos, la funcionalidad y los factores de calidad, los cuales se indican a continuación:

Funcionalidad

Dado que la función primordial de uPhoto es brindar un espacio donde los fotógrafos puedan mostrar su trabajo, la función de poder cargar o importar las fotografías, y poder mostrarlas en la aplicación, es el sentido de real de la aplicación, y por lo tanto es en orden de prioridad la función principal a desarrollar y depurar.

En segundo lugar se posiciona el hecho de brindar la opción de compartir información con otros, o en otras palabras, la interacción con otros usuarios, por lo que resulta muy importante que la funcionalidad de las publicaciones cumpla con lo esperado por los usuarios.

Por último, se considera que la opción de conocer lo último del mercado con respecto a cámaras no es tan importante como las mencionadas anteriormente y representa un extra, una opción informativa que se desea brindar para que la experiencia en la aplicación sea más completa.

Factores de calidad

Los factores de calidad de software se definen como el grado en que un sistema, componente o proceso cumple con un grupo específico de requisitos, expectativas y necesidades de los usuarios.

Para el caso de uPhoto, la usabilidad representa el factor de calidad más importante, ya que es necesario que el servicio sea comprensible y que aprender a utilizar la herramienta sea sencillo. En segundo lugar se encuentra la eficiencia, ya que ésta es fundamental al tratarse de un servicio web, por lo que los tiempos de respuesta y el uso adecuado de los recursos son esenciales para que los usuarios sientan comodidad y satisfacción en el uso del servicio.

En el tercer puesto se sitúa a la confiabilidad, ya que los usuarios requieren que sus datos se encuentren seguros, dentro de un sistema robusto y confiable, y que en caso de fallos, éste pueda recuperarse rápidamente. Por último se coloca a la mantenibilidad, ya que la facilidad para realizar cambios o el hecho de realizar pruebas se pueden llevar a cabo sin impactar de gran manera el servicio brindado.

Descripción de diseño de alto nivel

Diagrama de arquitectura conceptual

Este diagrama nos permite ver la arquitectura conceptual de la aplicación a gran escala. Tomando en cuenta los elementos que interactúan dentro de la aplicación como las exteriores con las que se relaciona para efectos de funcionalidad de la misma.

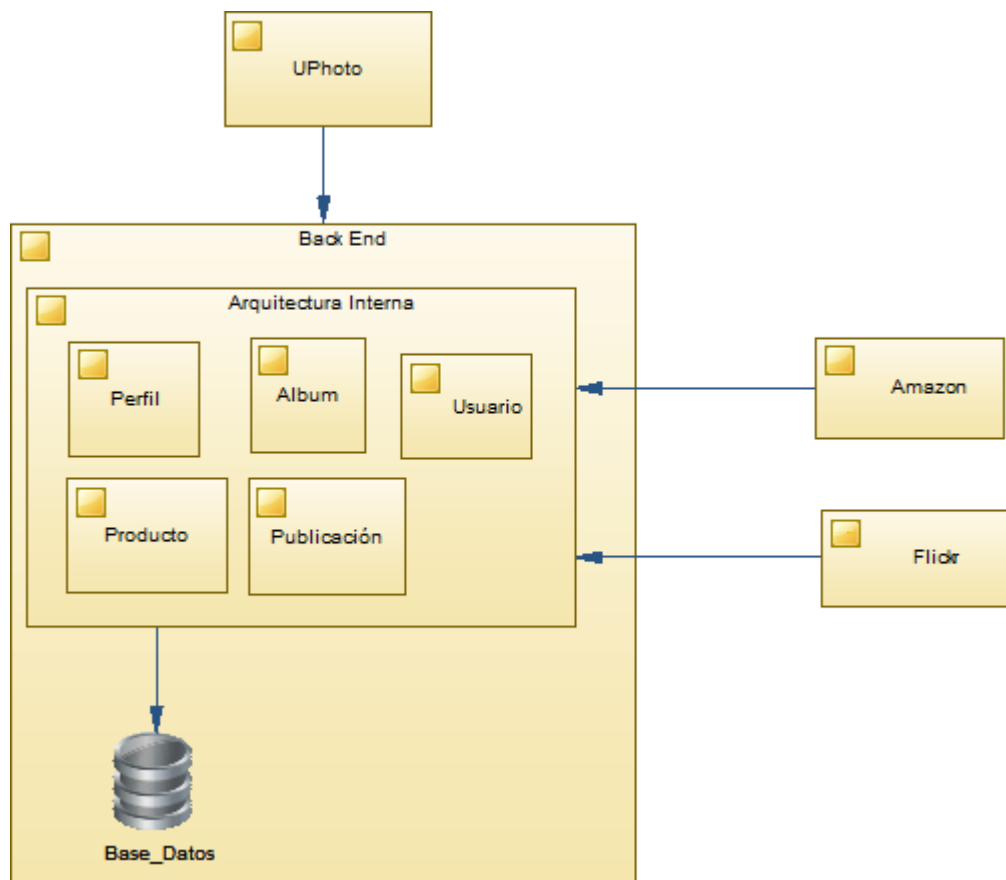


Diagrama de paquetes

Este diagrama muestra la organización de elementos a gran escala, con sus subsistemas y sus dependencias en la aplicación.

Entre los paquetes identificados está la capa de presentación la cual va a depender de la capa lógica con la que se desenvuelva la aplicación, la capa lógica a su vez va depender del paquete de varios y de la capa de datos la cual maneja una base de datos interna; externamente la capa de soporte servidor y la capa de servicios. Todos estos paquetes internamente agrupan un conjunto de componentes que llevan a cabo la funcionalidad de la aplicación.

Se muestra el siguiente diagrama presentando los paquetes de la aplicación:

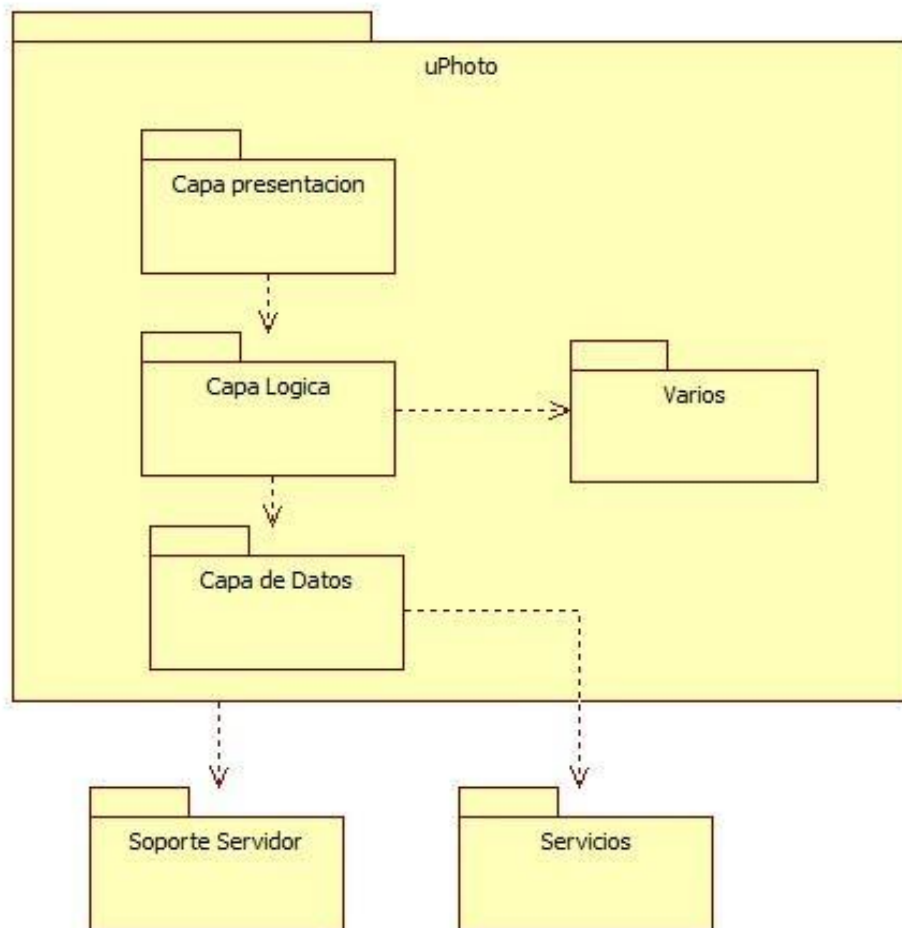


Diagrama de componentes

Describe relación y estructura de componentes, los cuales pueden ser de alto o bajo nivel; la aplicación está compuesta por los siguientes componentes:

- Capa de presentación contiene lo relacionado con la interfaz y la autenticación de los usuarios.
- Capa lógica contiene toda la lógica interna de la aplicación.
- El componente de varios está comprendido por seguridad, privacidad y ciertos detalles de la aplicación.
- Capa de datos contiene una base de datos local de la aplicación.
- Externamente están los componentes de soporte el cual tiene la base de datos servidor con todos los datos de la aplicación, y el componente de servicios que está compuesto por los Api's necesarias para llevar a cabo la aplicación.

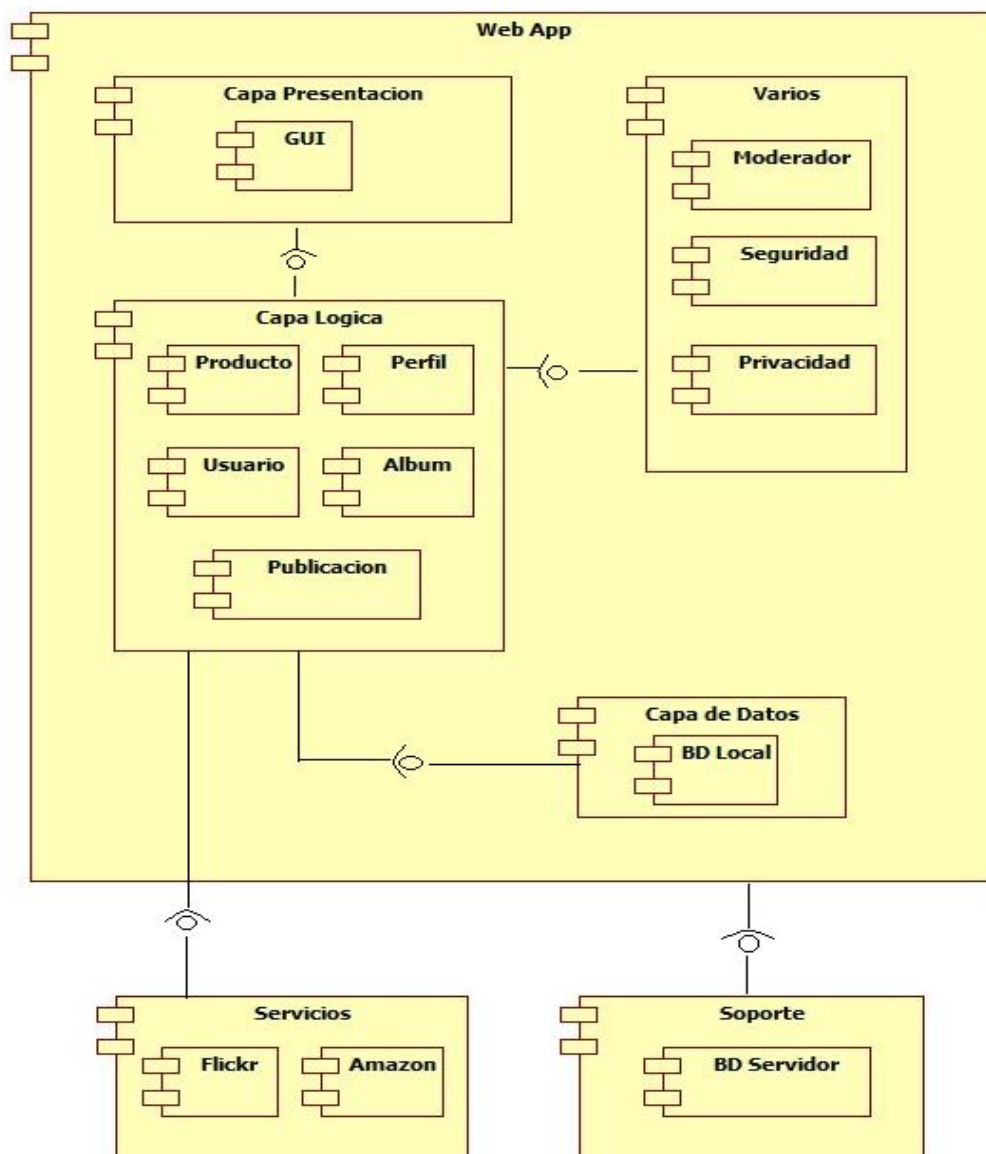


Diagrama de despliegue

Este diagrama muestra la organización en nodos o capas físicas, al igual que las conexiones con dispositivos o Api's externas como es en este caso Amazon y Flickr.

Entre los nodos que se presentan están los dispositivos físicos como el Device App Service y los nodos de ambiente de ejecución como el Device support.

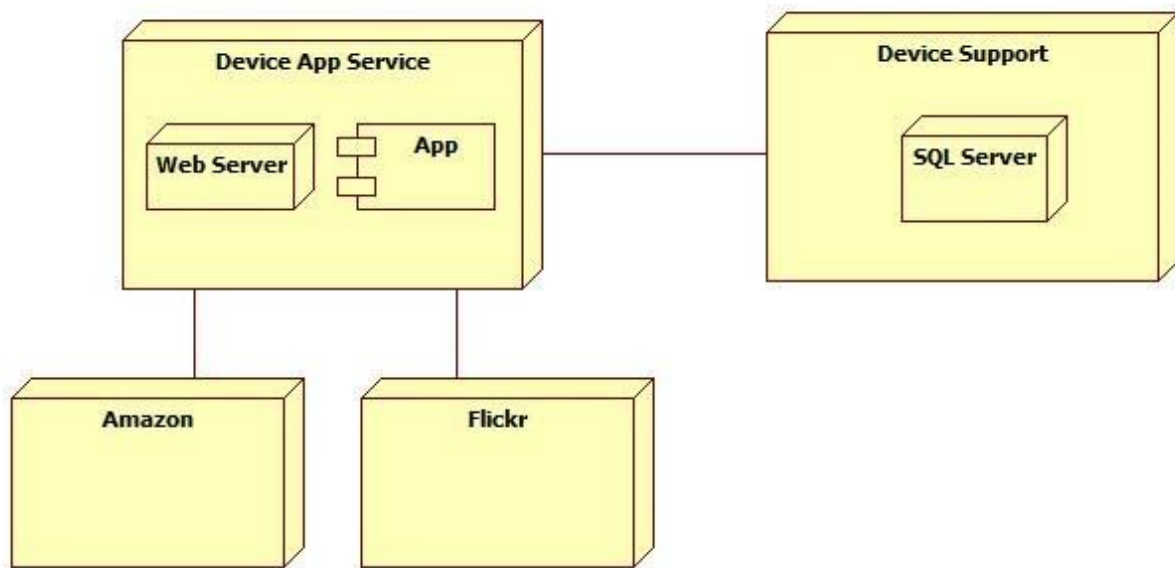
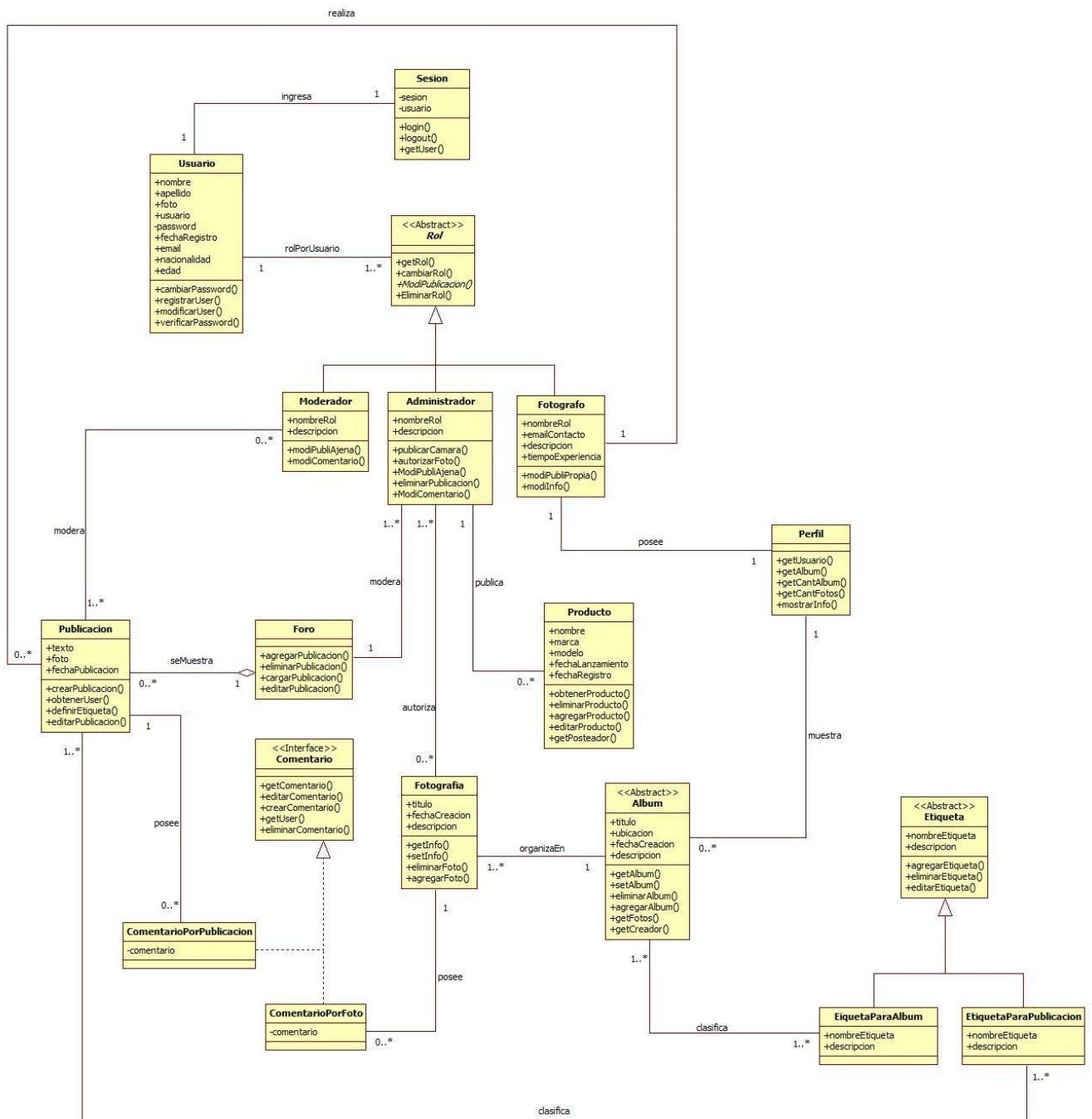


Diagrama de clases

Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones (incluyendo herencia, agregación, asociación, entre otros). Además, son el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer (análisis), como para mostrar cómo puede ser construido (diseño).

A continuación se muestra el diagrama de clases de la aplicación uPhoto:



Justificación de patrones utilizados

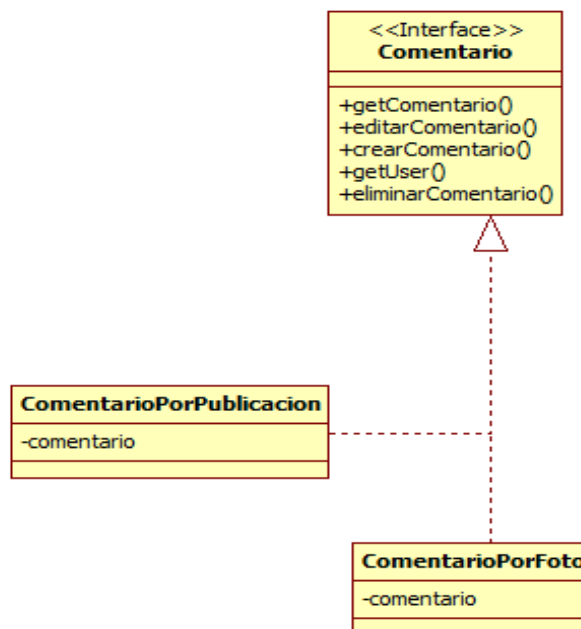
El objetivo del diseño de software es llegar a un producto de calidad. El diseño debe ser adecuado para la aplicación en particular, pero dicha solución también debe ser general para poder enfrentar distintos escenarios y evitar el rediseño.

Este es el objetivo de los patrones de diseño, donde cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada múltiples veces sin hacerlo ni siquiera dos veces de la misma forma. A continuación se mencionan patrones utilizados en la aplicación web uPhoto.

Interface solo lectura

Este representa un patrón en el que una interfaz es utilizada para restringir qué clases tienen privilegios para llamar métodos de actualización de una clase.

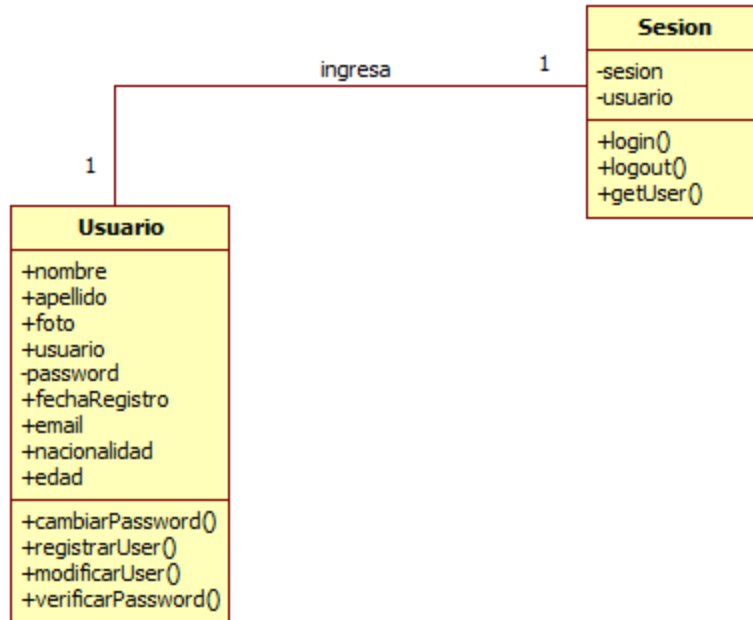
- Problema: se debe utilizar el patrón para heredar los métodos que se utilizarán sobre los comentarios, y que éstos no sean públicos y no cualquiera pueda editarlos, sino sólo el usuario actualmente autenticado.
- Solución: crear una Interface donde los métodos que la implementen heredan la definición de sus métodos, pero no su implementación, así se puede definir la implementación según el caso.
- Estructura:



Singleton

El patrón de diseño Singleton (instancia única) está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto.

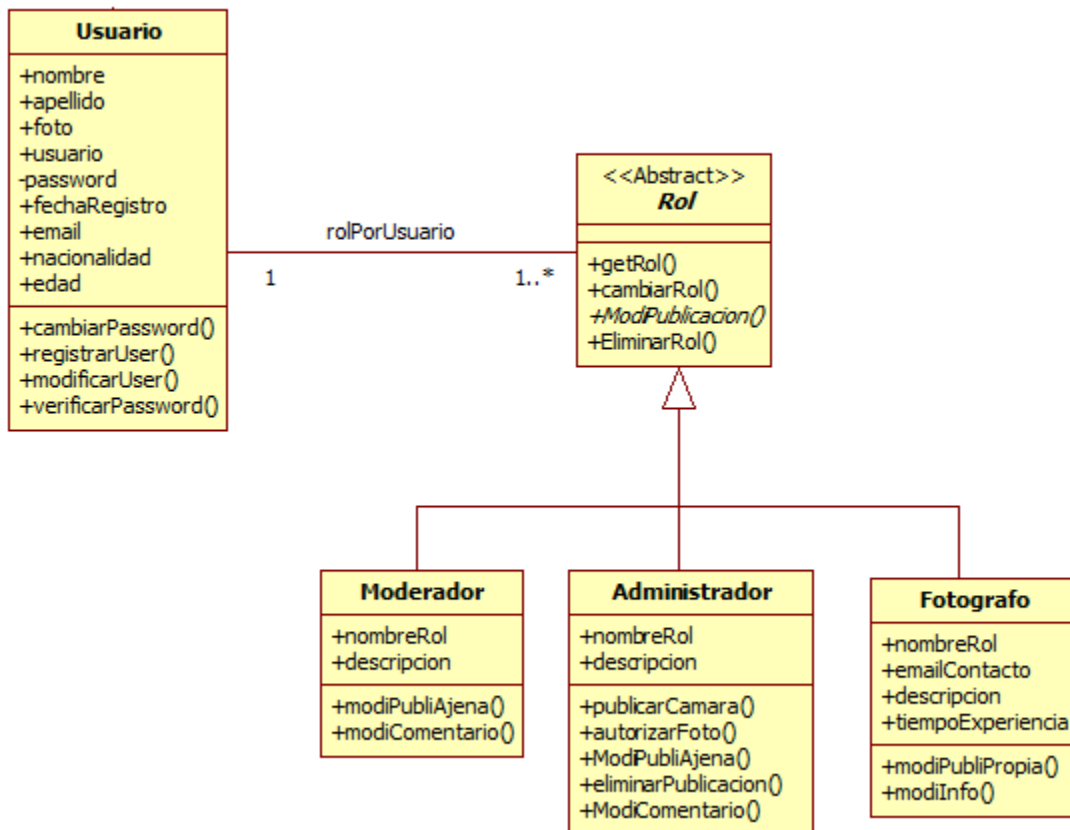
- Problema: tener diversas sesiones de usuario activas o usuarios autenticados al mismo tiempo con los mismos datos, o que para realizar acciones sea necesario autenticarse todo el tiempo.
- Solución: se crea la clase “Sesión” donde se usa el patrón Singleton, que permite que se mantenga una sola instancia de la sesión activa.
- Estructura:



Rol-jugador

Este patrón de diseño se utiliza en casos donde un objeto puede tener distintos roles en diferentes contextos, y cada rol representa un conjunto particular de propiedades asociadas a dicho objeto.

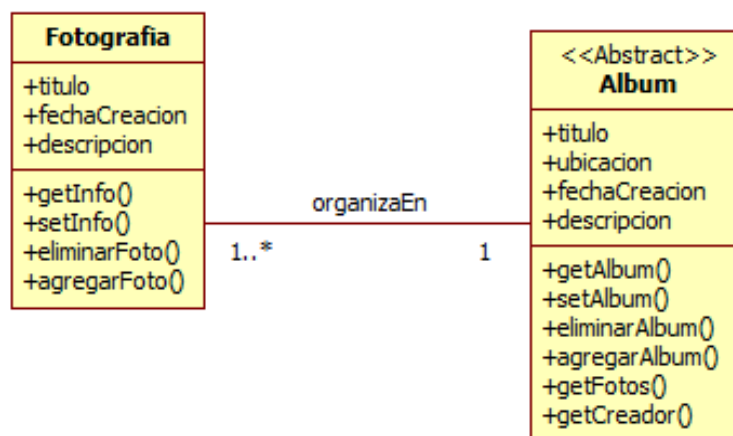
- Problema: dentro de la aplicación, un usuario registrado puede ser Moderador, Administrador o Fotógrafo, pero a la vez puede ser Moderador y Fotógrafo.
- Solución: se crea una clase abstracta "Rol", donde se encapsula información de los roles y se permite que un mismo usuario pueda tener diversos roles.
- Estructura:



Abstracción – ocurrencia

Este es un patrón en el cual dos clases están relacionados por una asociación, y una de las clases representa una abstracción de los demás.

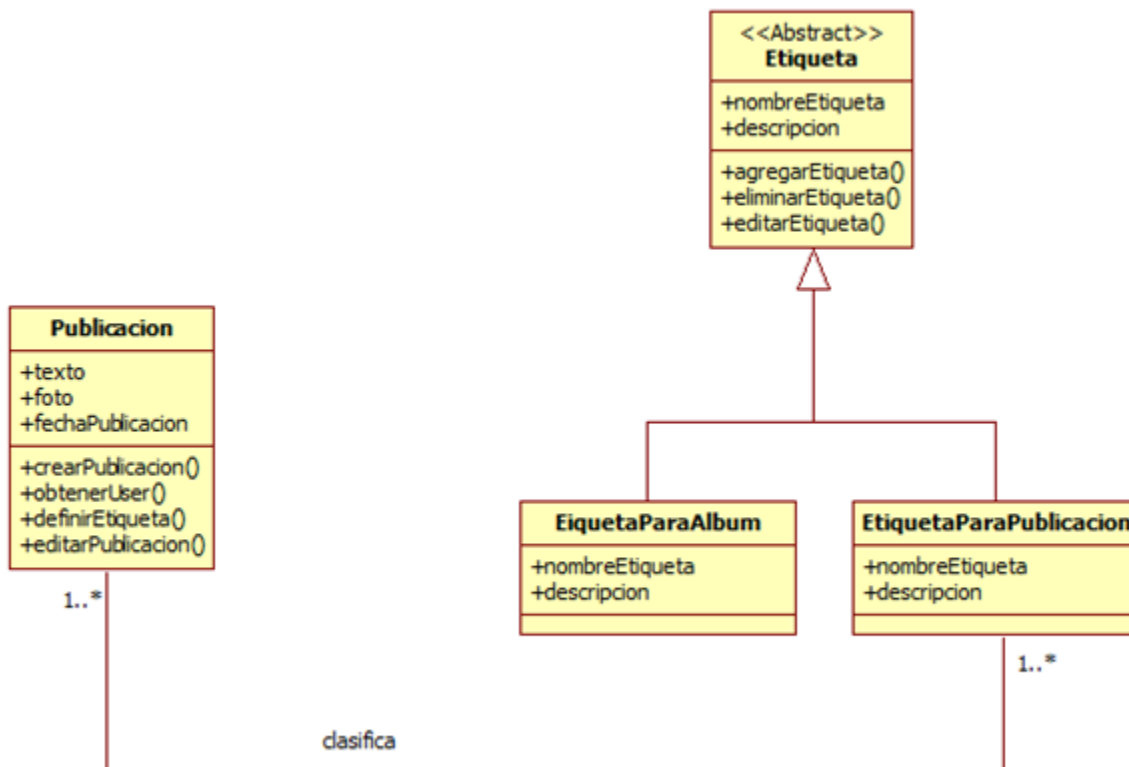
- Problema: hay fotografías que comparten información como el lugar donde fueron tomadas y la descripción o incluso el motivo de la obtención de dichas fotografías en ese lugar, pero además poseen información diferente por cada foto.
- Solución: crear una clase “Álbum” donde se tiene la relación de las fotografías, y el “Álbum” representa una abstracción del conjunto de fotografías.
- Estructura:



Delegado

Es un patrón de diseño que se utiliza cuando se diseña un método de una clase, y éste requiere de otro método de otra clase para implementarlo.

- Problema: se desea trabajar con métodos de la clase abstracta “Etiqueta”, sin tener que heredar.
- Solución: se define el método “definirEtiqueta()” que invoca la definición del método “agregarEtiqueta()” de la clase “EtiquetaParaPublicacion”, que a su vez hereda este método de la clase abstracta “Etiqueta”.
- Estructura:



Interacciones con sistemas externos

La aplicación desarrollada contará con 2 conexiones a sistemas externos, la primera será con Flickr, una aplicación creada para el almacenamiento de fotografías en la nube, donde cada persona registrada en esta aplicación cuenta con 1000 GB de almacenamiento gratuito, y además puede conectar la cuenta por medio de las nuevas aplicaciones para teléfonos inteligentes, para compartir toda clase de fotografías en tiempo real.

La segunda es la aplicación para conectarse a la tienda virtual Amazon, nuestra página se conectará a esta tienda con el fin de ofrecerle a los usuarios la posibilidad de acceder a accesorios y cámaras fotográficas especializadas.

Conexión con Flickr

La API de Flickr está compuesta por un grupo de métodos a los que se puede llamar.

Para realizar una acción usando la API de Flickr, se debe seleccionar una convención de llamada, enviar una solicitud a su extremo y especificar un método y algunos argumentos, y así se recibirá una respuesta con formato. Todos los formatos de solicitud, toman una lista de parámetros con nombre.

Hay un conjunto de formatos para llevar a cabo ciertas funciones:

- El parámetro OBLIGATORIO **method** se usa para especificar el método de llamada.
- El parámetro OBLIGATORIO **api_key** se usa para especificar tu clave API.
- El parámetro opcional **format** se usa para especificar un formato de respuesta.

Aspectos a considerar:

- Autenticación: El flujo de autenticación de Flickr está diseñado para garantizar que las cuentas de nuestros usuarios sean seguras. por lo que la forma para acceder desde la aplicación que se está creando hasta la API de Flickr, es por medio de que los usuarios se dirijan a Flickr para autenticar una aplicación de terceros
- Límites: El acceso a la API es por clave. Si la aplicación se mantiene debajo de las 3600 consultas por hora en toda la clave (lo que significa la

suma de todos los usuarios de la integración), no se tendrá problemas. Si se detectamos un abuso con la clave, se procederá a que ésta caduque o a desactivarla, para preservar la funcionalidad de la API de Flickr para los demás

- Almacenamiento en caché a corto plazo: La aplicación puede almacenar imágenes y resultados de la API en la memoria caché durante 24 horas como máximo para reducir la carga de la API.
- Para hacer Uso de la URL corta “flic.kr”: Si prefieres usar una URL corta para una foto, se puede usar el servicio para acortar URL nativo flic.kr. El formato es:flic.kr/p/[short-photo-id] donde el ID de la foto corto es una conversión base58 del ID de la foto..

Para este caso en específico la conexión entre Flickr y nuestra aplicación se hará utilizando FlickrNet API Library, la cual está escrita completamente en C# y puede accederse desde los siguiente frameworks:

- .Net Framework 2.0 and above.
- .Net Compact Framework 2.0 SP1 and above.
- Silverlight 3.0 and above.
- Windows 7 Phone
- Mono
- Monotouch for iPhone

Conexión con Amazon

Amazon cuenta con una API que permite a los desarrolladores acceder a su página y mostrar productos po este medio. Utilizando acciones como las búsquedas y los comentarios de los clientes de amazon.

Esta API lleva por nombre Product Advertising y proporciona acceso mediante programación a la selección de productos de Amazon y la funcionalidad de descubrimiento.

¿Cómo funciona?

Acceso Amazon para la selección de productos: le da acceso a la selección de millones de productos en categorías como libros, música, descargas digitales y aparatos electrónicos de Amazon.com, Amazon.ca, Amazon.co.uk, Amazon.de de Amazon, Amazon.fr y Amazon.co.jp.

Aproveche las capacidades de descubrimiento del producto que facilita Amazon: El API Publicidad Producto le permite aprovechar las características centradas en el cliente de Amazon, como la búsqueda de productos, opiniones de clientes, productos similares, Accesorios, listas de favoritos y mucho más.

Beneficios económicos de su sitio web: El API Publicidad Producto le permite incluir su etiqueta asociada en peticiones de API para formatear automáticamente los URL devueltos por el API para asegurarse de que usted gana comisiones por referencia cuando los usuarios que se refieren a sitios de Amazon compran productos que califican.

Para el caso de esta aplicación se utilizará ItemLookup para realizar peticiones y obtener títulos de productos, enlaces a Amazon, urls de imágenes y precios de los nuevos anuncios, usados y de colección en Amazon, Opiniones de los usuarios, accesorios, productos similares y mucho más.

Se tomarán en cuenta aspectos como:

- Acuerdo de licencia del producto: para más detalles sobre el uso de directrices, políticas y requisitos.
- La Guía de Aplicación de Mejores Prácticas para asegurarse de que la aplicación cumple, escalable y eficiente.
- Las normas de ahorro de API, que están diseñados para promover el uso eficiente de la API.

Dominio y servidor activos

La aplicación uPhoto se encuentra alojada en los servicios de Windows Azure, el cual es el encargado de proporcionar el alojamiento de la aplicación y el almacenamiento de la base de datos. La dirección web de la aplicación es <http://uphoto.azurewebsites.net/>.

Para la gestión del código fuente de la aplicación, cada uno de los miembros hará su aporte al desarrollo en un repositorio en Github, el cual tiene la dirección <https://github.com/Kazzak/uPhoto>, donde se aloja el código fuente de la aplicación y el servicio de Windows Azure lo exporta para “correrlo” en sus servidores.

Conclusiones

Después de realizar la primer parte del proyecto, se llegaron a las siguientes conclusiones:

- La aplicación uPhoto brinda un espacio a los fotógrafos de mostrar su trabajo de forma profesional, donde las personas pueden ver su trabajo y su perfil, dándoles la oportunidad de ser contactados
- El uso de los diferentes diagramas utilizados en este proyecto, permite observar con mayor claridad cuáles son las partes que componen la aplicación uPhoto, además de la interacción con las demás aplicaciones externas.
- Mientras se realizaban los diagramas, se podían observar los diferentes problemas de diseño que iban surgiendo, por lo que la utilización de los patrones de diseño, ayudó a mejorarlos y a solventarlos
- Se buscaron los diferentes requerimientos y se especificaron por medio de casos de uso, permitiendo de esta manera tener una clara imagen de lo que la aplicación uPhoto debe hacer y los parámetros que debe cumplir
- La aplicación uPhoto estará alojada en los servicios de Windows Azure
- EL código fuente se almacenará en un repositorio de Github

Bibliografía

Amazon Product Advertising API Best Practices. Tomado el 25 de mayo del 2014, recuperado de http://aws.amazon.com/articles/1057?_encoding=UTF8&jiveRedirect=1

Acuerdo de Licencia del Producto. Tomado el 25 de mayo del 2014, recuperado de https://affiliate-program.amazon.com/gp/advertising/api/detail/agreement.html?ie=UTF8&pf_rd_i=assoc-api-detail-5-v2&pf_rd_m=ATVPDKIKX0DER&pf_rd_p=&pf_rd_r=&pf_rd_s=assoc-center-1&pf_rd_t=501&ref=amb_link_83957651_1

La guía de aplicación de las mejores prácticas. Tomado el 25 de mayo del 2014, recuperado de http://aws.amazon.com/articles/1057?_encoding=UTF8&jiveRedirect=1

Las normas de ahorro de la API de Amazon. Tomado el 25 de mayo del 2014, recuperado de <http://docs.aws.amazon.com/AWSECommerceService/latest/DG/TroubleshootingApplications.html>

Flickr.Net API Library. Tomado el 23 de mayo del 2014, recuperado de <http://flickrnet.codeplex.com/>

Configurando nuestro sitio web base con Bootstrap | Tutos y Tips (2014, 30 de Marzo). Recuperado de <http://www.tutosytips.com/tutorial-configurando-nuestro-sitio-con-bootstrap-3/>

Garcia, M. (2013, 09 de Octubre). MSDN Blogs Recuperado de <http://blogs.msdn.com/b/en-un-lugar-de-la-nube/archive/2013/10/09/todo-lo-que-deber-237-as-saber-sobre-la-versi-243-n-gratuita-de-windows-azure-de-30-d-237-as.aspx>

Hanselman, S. (2012, 07 de Junio). Getting started on Windows Azure with .NET and Visual Studio. - YouTube Recuperado de <https://www.youtube.com/watch?v=i8MOJho5TDs>

Patrones de diseño (2011, 25 de Octubre). Recuperado de <http://www.slideshare.net/kaolong/patrones-de-diseo-i>

Salinas, J. (2011, 06 de Mayo). Aprendiendo a leer la documentación de un patrón de diseño : Método factoría - Inicio - ematiz Recuperado de <http://www.tekuento.net/inicio/-/blogs/aprendiendo-a-leer-la-documentacion-de-un-patron-de-diseno-metodo-factoria>

WINDOWS AZURE: CREACION Y ADMINISTRACION DE UNA BASE DE DATOS SQL SERVER - IT PRO Colombia - Site Home - TechNet Blogs (2013, 01 de Marzo). Recuperado de <http://blogs.technet.com/b/itprocol/archive/2013/03/01/windows-azure-creacion-y-administracion-de-una-base-de-datos-sql-server.aspx>

cs.sjsu.edu (s. f.). Actor-Role Pattern Recuperado de <http://www.cs.sjsu.edu/~pearce/oom/patterns2/patterns/analysis/Actor.htm>

prgassist.blogspot.com (2011, 30 de Abril). Programmer Assistant: Design Patterns(Singleton, Decorator, Immutable Class) Recuperado de <http://prgassist.blogspot.com/2011/04/design-patternssingleton-decorator.html>

site.uottawa.ca (s. f.). Fact Guru: Object Oriented Software Engineering Recuperado de <http://www.site.uottawa.ca:4321/oose/index.html#abstraction-occurrence>

site.uottawa.ca (s. f.). Fact Guru: Object Oriented Software Engineering Recuperado de <http://www.site.uottawa.ca:4321/oose/index.html#abstraction-occurrence>