27-5-2014

Diseño de software

Luis Coto Mata- Erick Vargas Víctor

PROYECTO: TOURIST APP

PROFESOR: ANDREI FUENTES LEIVA

Tabla de contenido

Resumen Ejecutivo

Este proyecto consiste en crear una aplicación web que permita trazar rutas en un mapa, entre la ubicación actual y un destino definido por el usuario. Una vez trazada esa ruta, se deberá mostrar imágenes acerca del destino elegido.

Además de las características antes mencionadas, la aplicación deberá gestionar un catálogo de usuarios registrados, que podrán subir fotografías a sus cuentas. Cuando esto suceda, se deberá poner un pin en el mapa, en la ubicación en la que la fotografía fue subida.

El sistema tendrá una amigable e intuitiva interfaz gráfica con la que el usuario podrá interactuar con mucha familiaridad y no tendrá problemas para sacarle provecho a todas las funcionalidades que le ofrece la aplicación web.

Como mejora a futuro para el sistema, se tiene pensado que para cada trazo de ruta que se realice entre dos puntos, se muestren imágenes que hayan subido otros usuarios en las cercanías del destino.

Especificación de requerimientos

El presente apartado contiene la especificación de los requisitos para el desarrollo del proyecto correspondiente al curso: diseño de software. Estos requisitos se detallan de tal manera, que pueden servir como guía para las siguientes etapas del ciclo de desarrollo de software. Cada uno de ellos se asocia con un código único que permitirá identificarlos y referenciarlos en artefactos posteriores a este.

Además de la descripción de los requerimientos, se ofrece un esbozo acerca de las funcionalidades que debe presentar el sistema para ser aceptado.

Acrónimos

Tourist App	Nombre del proyecto.
RF	Requisito funcional.
RNF	Requisito no funcional.

Funciones del sistema

- Gestión de usuarios

El sistema deberá permitir crear usuarios. Estos tendrán una dirección de correo electrónica y una contraseña que le permitan autenticarse. Estos usuarios podrán almacenar imágenes que se etiquetarán según el lugar donde sean subidas. Las imágenes y la información proporcionada por los usuarios deberán ser almacenadas en una base de datos.

- Trazar rutas

El sistema será capaz de crear una ruta desde la ubicación actual del dispositivo conectado, hasta un lugar indicado.

- Mostrar imágenes

El sistema deberá mostrar imágenes acerca del lugar que haya buscado el usuario.

Requisitos

En este apartado se presentan los requisitos funcionales y no funcionales que deberán ser satisfechos por el sistema.

Requisitos Funcionales

- Gestión de usuarios

RF001. El usuario se podrá registrar en el sistema

Para ello se solicitará que digite una dirección de correo electrónico y una contraseña. Esta información se almacenará en una base de datos.

RF002. El usuario podrá ingresar al sistema

Se debe permitir al usuario ingresar a su cuenta usando alguna dirección de correo electrónico, si acierta la contraseña respectiva.

RF003. El usuario podrá recuperar su contraseña olvidada

Se debe permitir al usuario recuperar su contraseña, enviándole un correo a su dirección, con la contraseña que tiene asociada actualmente.

RF004. El usuario podrá cambiar su contraseña

Una vez autenticado, el usuario tendrá la posibilidad de cambiar su contraseña.

RF005. El usuario podrá almacenar las fotos deseadas en la base de datos del sistema

Los usuarios que ya han sido autenticados, podrán subir fotografías. Se deberá poner un pin en el lugar del mapa en donde ha sido subida la foto.

RF006. El usuario podrá visualizar las fotos que hayan sido subidas desde su cuenta.

Se debe permitir al usuario ver todas las fotos que haya subido anteriormente.

- Trazar rutas

RF007. El sistema podrá trazar rutas a destinos solicitados por el usuario

Se deberá de trazar rutas entre la posición actual y un destino indicado por el usuario del sistema. Para esto no se necesita tener un usuario autenticado.

- Mostrar imágenes

RF008. El sistema mostrará imágenes relacionadas con el destino de una ruta trazada

Se deberá buscar imágenes acerca de un lugar (destino) y deberán ser mostradas al usuario. Estas imágenes deben obtenerse mediante el uso de Google Custom Search API

Requisitos No Funcionales

- APIS externas

RFN1. El sistema debe utilizar la API de Google Maps

Para la ubicación del usuario y el trazado del camino se utilizará la API de Google Maps.

RFN2. El sistema debe utilizar la API de Google Custom Search

Para la extracción de las imágenes se utilizará la API de Google Custom Search.

Prioridades

A continuación se muestran las funcionalidades y factores que, entre otras funciones, permitirán guiar el proceso de desarrollo y además, usarse como métricas para los procesos de control y aseguramiento de la calidad.

Las funcionalidades, tomadas de la especificación de requerimientos, están ordenadas de la siguiente manera:

1º. Trazar rutas

2º. Mostrar imágenes

3º. Gestión de usuarios

Los factores de calidad esperados, presentan el siguiente orden:

1º. Usabilidad

Presentar una apariencia atractiva, además una amigable interacción con el usuario y una forma intuitiva de acceder a todas sus funcionalidades, serán factores críticos para la aprobación del sistema.

2º. Eficiencia

Al ser una aplicación web, el uso óptimo de los recursos se convierte en el segundo factor más importante para determinar la calidad del producto. La aplicación debe evitar sesiones de carga abandonadas debido a tiempos de espera demasiado lentos, cuellos de botella y cualquier tiempo de espera excesivo ante las solicitudes del usuario.

3º. Confiabilidad

Nadie utilizaría una escoba que no barra bien. Basado en esa idea, la confiablidad de nuestro sistema, esa capacidad de garantizar que hace lo que se espera que haga, es el tercer factor más importante para determinar su calidad.

4º. Mantenimiento

Para facilitar la solución de cualquier error y ahorrarles a los ingenieros, muchos dolores de cabeza, el sistema debe crearse pensado en poder aplicarle un fácil mantenimiento.

Descripción de diseño de alto nivel

Diagrama de arquitectura conceptual

Ver Anexo: Diagrama de arquitectura conceptual
Diagramas\Arquitectura Conceptual.pdf

Diagrama de paquetes

Ver Anexo: Diagrama de paquetes
Diagramas\Paquetes.pdf

Diagrama de componentes

Ver Anexo: Diagrama de componentes
Diagramas\Componentes.pdf

Diagrama de despliegue

Ver Anexo: Diagrama de despliegue
Diagramas\Despliegue.pdf

Descripción detallada

Diagrama de clases

Ver Anexo: Diagrama de clases
Diagramas\Clases.pdf

El diagrama de clases anexado, presenta la estructura que tendrá el sistema. En este se puede apreciar la interrelación que habrá entre el usuario externo del sistema, que utilizará una interfaz web para acceder a las funcionalidades que ofrece la aplicación.

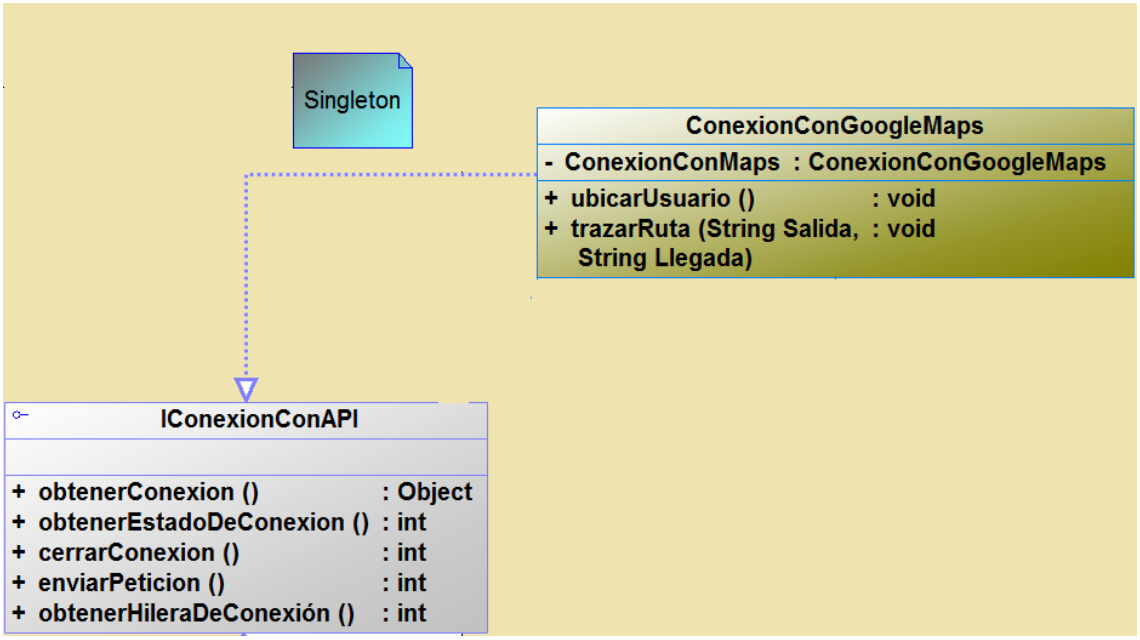
Esta interfaz será la encargada de permitir la comunicación con sistemas externos, como los servicios proveídos por Google, además accederá las bases de datos del sistema, para autenticar clientes y almacenar fotografías.

Justificación de patrones usados

A continuación se detallan los problemas encontrados y su solución mediante el uso de patrones:

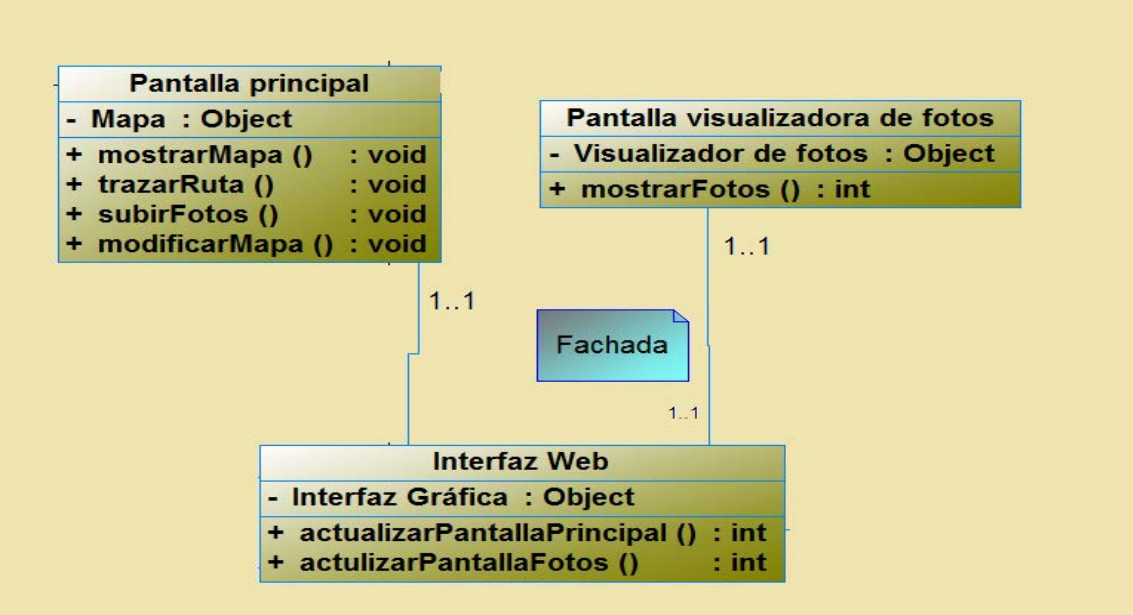
- Limitar la cantidad de conexiones por cada usuario, a sistemas externos.

  Esto para controlar la concurrencia de usuarios y evitar abusos en la cantidad de solicitudes tramitadas a través de las APIs que podrían desencadenar en mal funcionamiento de la aplicación. Para poder solucionar este problema, se aplicó el patrón Singleton, como se nota en la siguiente imagen, cada usuario podrá realizar una única conexión con el API de Google Maps y sucede análogamente con la conexión a Google Custom Search.



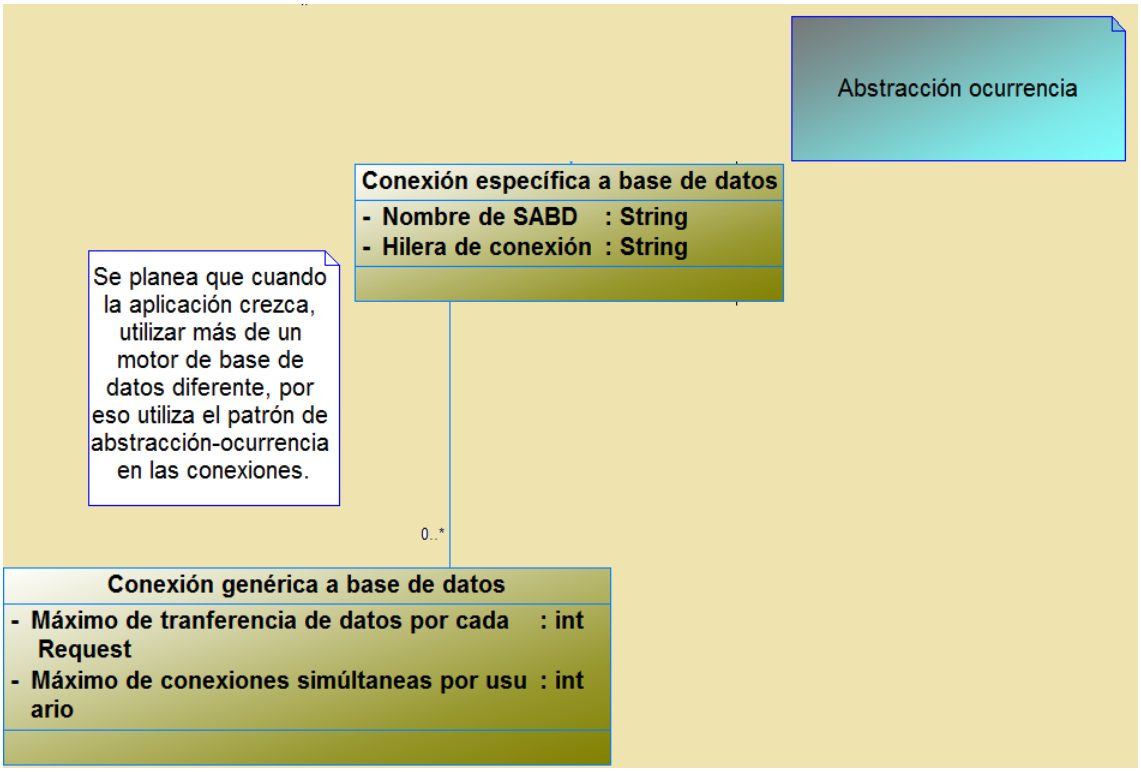
- Integrar varias clases en una sola, para facilitar manejo del sistema

Esto para simplificar el manejo de la interfaz de usuario, para esto se utilizó el patrón Fachada como se muestra a continuación:



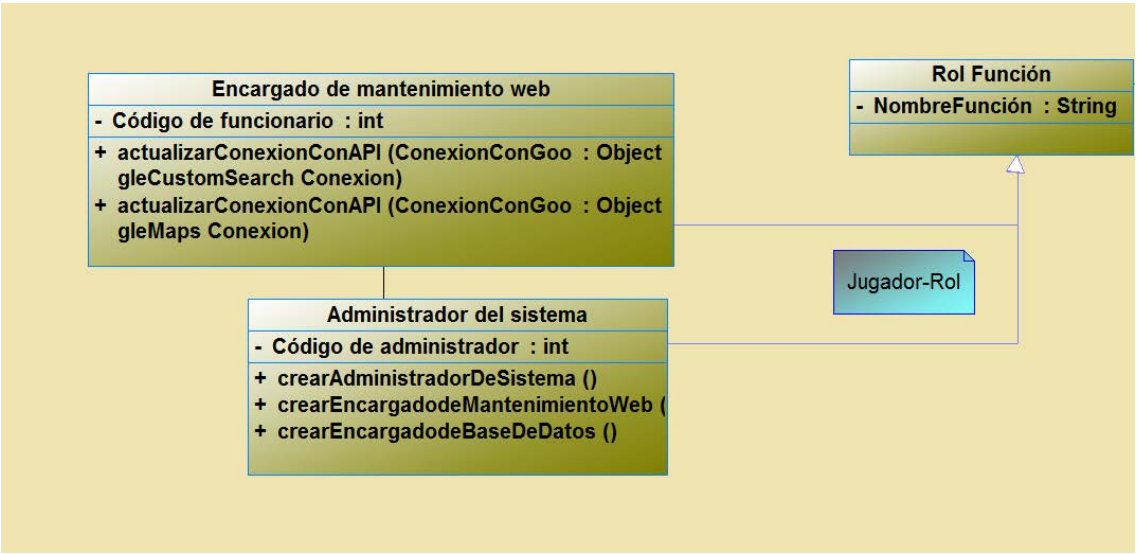
- Evitar la redundancia innecesaria de datos

Para mantener una alta eficiencia en el uso de recursos, según lo indicado por los factores de calidad esperados, se utilizó el patrón de abstracción-ocurrencia como se puede apreciar en la siguiente figura:



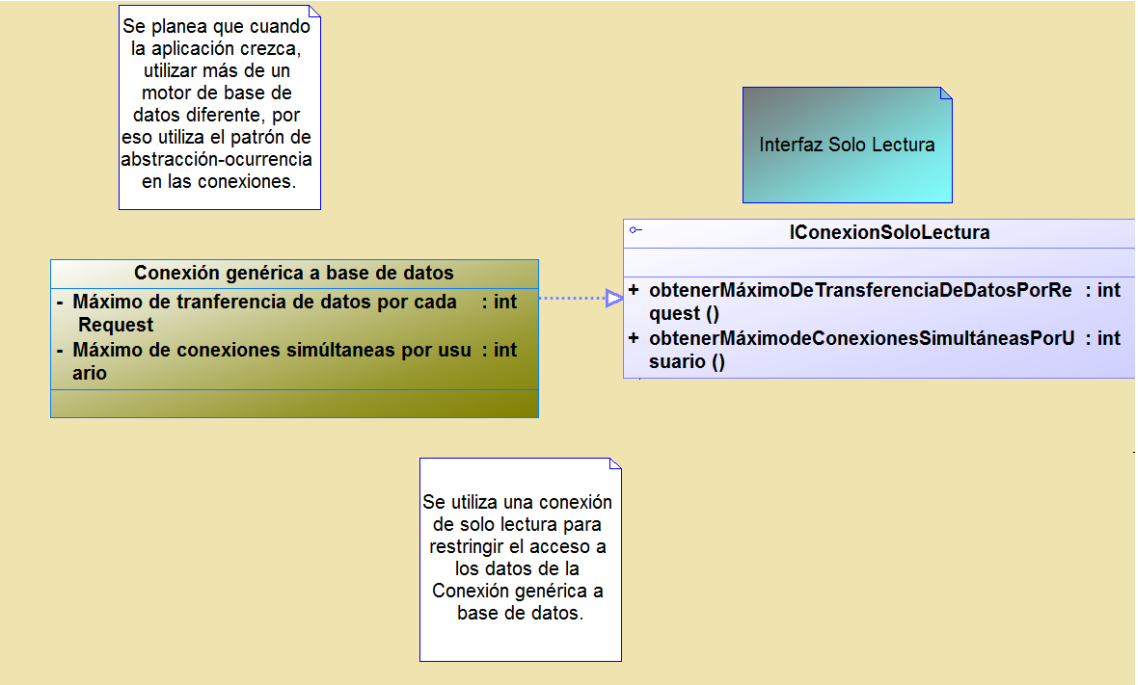
- Permitir a un objeto ejercer diversos roles (Rol-jugador)

Para permitir que una persona ejerza funciones de administración y mantenimiento de la aplicación. Como se muestra a continuación:



- Asegurar que algunos datos no sean cambiados por usuarios estándares

Para lograr esto, se aplicará el patrón de interfaz de solo lectura, permitiendo volver inmutables a los atributos de solo lectura de la clase de conexión genérica a la base de datos, de la siguiente forma:



Problemas de diseño

- Usuarios:

Como se mencionó en las funcionalidades, el sistema va administrar cuentas de usuario. Estas van a estar almacenadas en una base de datos, y cuando el usuario ingrese a su perfil le permitirá subir fotos y registrar los lugares donde estas fueron tomadas.

El sistema no amerita que sea muy seguro por lo que la aplicación no se va a centrar tanto en este enfoque. Cuando el usuario se registra en la aplicación, se realiza con el fin de que esta pueda llevar un control y además que le proporcione un tipo de seguridad para que no se dé un tipo de suplantación de identidad.

- Rutas:

Las rutas son otro aspecto mencionado en las funcionalidades. La aplicación va estar enfocada en que el usuario va ingresar su ubicación actual y un destino en específico y de esta manera en un mapa va a aparecer la ruta para llegar a este lugar. Con la utilización de uno de los APIs (Google Maps) es que se va a desarrollar a resolver este problema de diseño.

- Imágenes:

La imágenes que van a ser subidas en el sistema no van a tener ninguna privacidad, debido a que se suponen que las fotos de los lugares que van a subirse son sitios públicos y turísticos. Como se mencionó anteriormente, la idea es recopilar información de los sitios aledaños al destino y a la vez presentarle estos sitios que también podría visitar por su cercanía. Como una futura implementación, es que se va a recopilar información tanto el API de Google Custom Search, como de la base de datos de la aplicación para que ambas puedan ser mostradas a otros usuarios.

Y de esta manera se podrían resolver estos problemas de diseño anteriormente descritos.

Interacción con sistemas externos

El sistema sacará provecho de las funcionalidades provistas por los siguientes sistemas:

- **Google Maps**

  Es un servidor de aplicaciones de mapas en la web que pertenece a Google. Se pretende conectarse a su API, para mostrar mapas, obtener la localización del usuario y trazar rutas.

- **Google Custom Search**

  Es un servicio que permite realizar búsquedas personalizadas. A través de su API, se buscarán imágenes acerca del destino que elija un usuario al momento de trazar una ruta.