

Documentación externa: TuristearCR

INSTITUTO TECNOLÓGICO DE COSTA RICA

ESCUELA DE INGENIERÍA EN COMPUTACIÓN

LICENCIATURA EN ADMINISTRACIÓN DE TECNOLOGÍAS DE INFORMACIÓN

ESTUDIANTES:

MIGUEL ÁNGEL GONZÁLEZ GUTIÉRREZ - 201136785

MARIA FERNANDA FERNÁNDEZ OCAMPO -201062939

• Resumen ejecutivo

El auge turístico de hoy en día, está haciendo que muchas personas deseen conocer más sobre los atractivos turísticos que ofrece el país, pero lastimosamente no hay información en conjunto que describa los lugares turísticos o de recreo que se pueden visitar en ciertas zonas, viéndonos en la necesidad de esto, se pretende ayudar a la divulgación de información de parques nacionales y además de otros centros turísticos que se puedan ver beneficiados, promoviendo el desarrollo económico y social del área designada, para esto nos enfocaremos en la zona de Cartago, la cual posee muchos lugares, pero desconocidos para muchos.

Bajo la idea de que “en Cartago no hay nada”, se planea hacer una aplicación web en la cual cualquier usuario de la aplicación pueda acceder a información de lugares, horarios, precios, ubicaciones, para que así al usuario se le haga más fácil tomar una decisión sobre a donde puede ir, habiendo comparado entre las opciones que ofrece la aplicación y luego de visitar el lugar también pueda dejar algún comentario sobre el mismo.

Y también promoviendo la conservación del medio ambiente de nuestros parques nacionales, se les ofrece una forma en la que el usuario pueda registrarse, para acceder a información de los lugares turísticos de Cartago y si el lugar lo permite, el poder registrarse para realizar voluntariado en los mismos. Proporcionando así una herramienta a los guarda parques y otras entidades, para que tengan un mejor control sobre los voluntarios que pueden albergar, su forma de voluntariado y seleccionarlos dependiendo de las cualidades que puede ofrecer el usuario.

La aplicación web va a tener dos grandes funcionalidades, la primera es principalmente informativa para el usuario, mostrándole datos sobre parques, cafeterías, paradas de autobuses y tren, miradores, patrimonio arquitectónico, parques nacionales, museos, centros turísticos y botánicos, estos organizados en categorías para una búsqueda más eficaz. La segunda funcionalidad va más orientada para el uso de guarda parques y otros encargados, la cual consiste en brindarles información de voluntarios que estén dispuestos a colaborar de una forma gratuita, promoviendo así un mayor orden a la hora de seleccionar los voluntarios conforme a lo que el parque o centro turístico ocupe en ese momento.

- **Especificación de requerimientos**

- Requerimientos Funcionales**

- El sistema debe contener información del nombre, horario, ubicación, precios o tarifas; si es permitido por la administración del lugar, y otros datos sobre el lugar que se eligió.

- El sistema debe crear comentarios de un determinado lugar, generando mantenibilidad para que otros usuarios los observen.

- El sistema debe de brindar la opción de publicar en la red social twitter, el destino que el usuario ubique de su agrado.

- El sistema debe de mostrar las formas de voluntariado, las cuales deben contar con los horarios, una descripción más detallada, el encargado, área de registro, conforme al destino elegido.

- El sistema debe estar completamente documentado.

- El sistema debe de mostrar en el registro del voluntariado todos los campos de información básica de contacto del voluntario, así como una descripción del por qué desea hacer el voluntariado y que otras cualidades que puede aportar.

- El sistema debe ofrecer una sección de ayuda, un manual de usuario.

- Requerimientos No Funcionales**

- Mantener alta disponibilidad: El sistema deberá estar disponible el 90% del tiempo en general (24/7).

- Responder inicio de sesión pronto: La respuesta cuando un usuario envía la solicitud de iniciar sesión, deberá ser procesada por el sistema y entregada al usuario en menos de 1 minuto.

La mantenibilidad del sitio queda a cargo del o los administradores de la aplicación web, para lo cual se capacitarán a estas personas.

La aplicación web deberá funcionar en los siguientes sistemas operativos: Windows, Linux y Mac OS. Siempre y cuando el aparato cuente con un explorador web, como por ejemplo: Google Chrome, Mozilla Firefox, Safari e Internet Explorer.

• Prioridades (funcionalidades y factores de calidad más importantes)

Funcionalidades:

- Publicar en la red social twitter, el Twitt lo realiza cada usuario.
- Creación de cuentas de usuario para visitantes y administradores.
- Brindar la información de los destinos turísticos.
- Generar una base de datos de Voluntarios Inscritos al sistema de voluntariado de parques nacionales.
- Facilidad de modificaciones y evaluaciones del sistema.
- Documentación interna y externa

Factores de Calidad:

- Mantenibilidad con las consultas ejecutadas por los usuarios, tanto visitantes como administradores.
- Integridad en el ingreso, modificación y borrado de comentarios.
- Usabilidad para que el usuario, cuente con una excelente UserExperience.
- Naturalidad en las redacciones de información.
- Confiabilidad en la funcionalidad principal, la cual es brindar información y opción de registro para voluntario para las reservas que cumplan con el mismo.
- Interfaz ordenada y navegabilidad coherente.

- Descripción de diseño de alto nivel

Diagrama de arquitectura conceptual

Visual Paradigm for UML Standard Edition (Universidad del País Vasco)

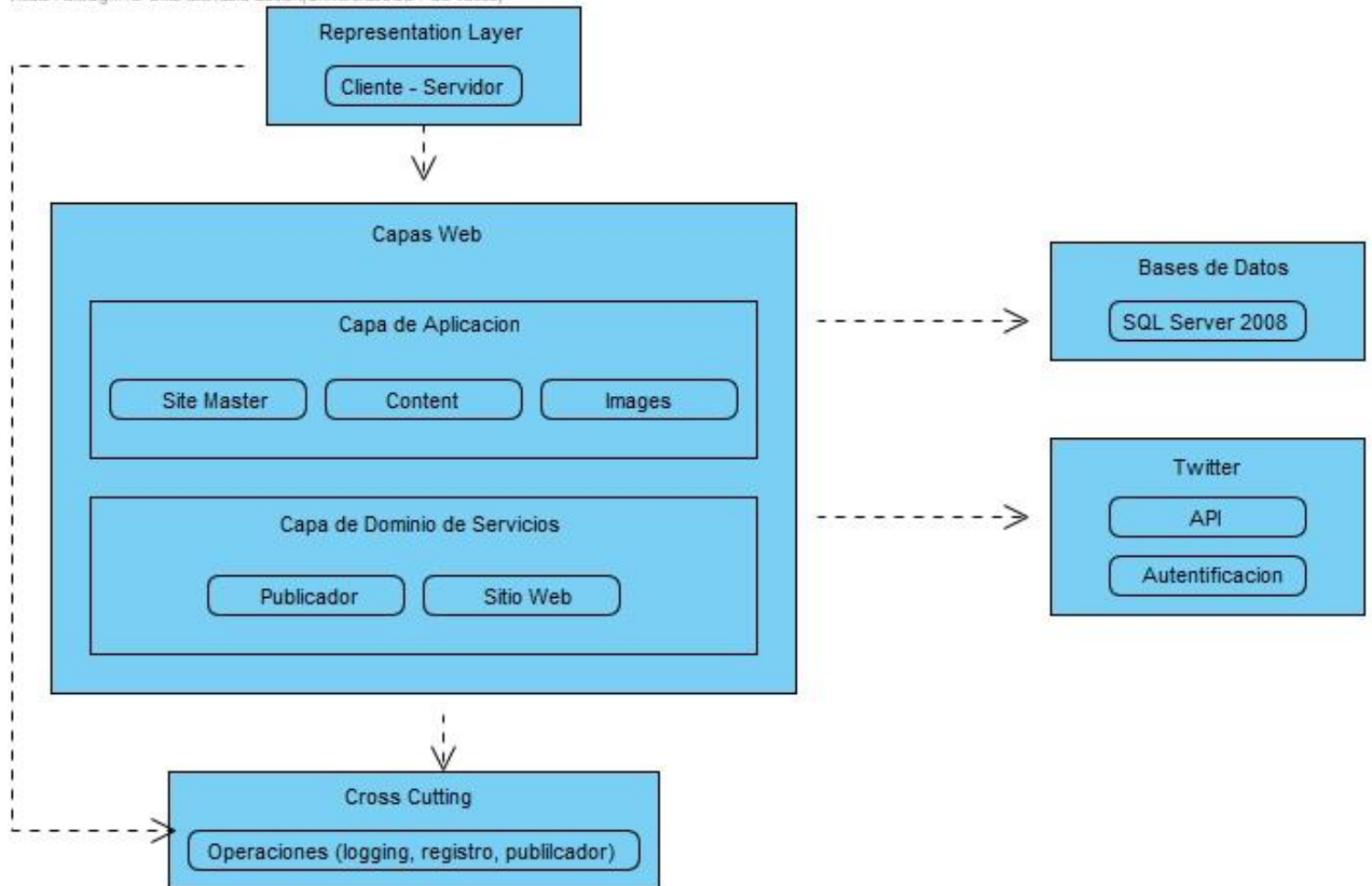


Diagrama de paquetes

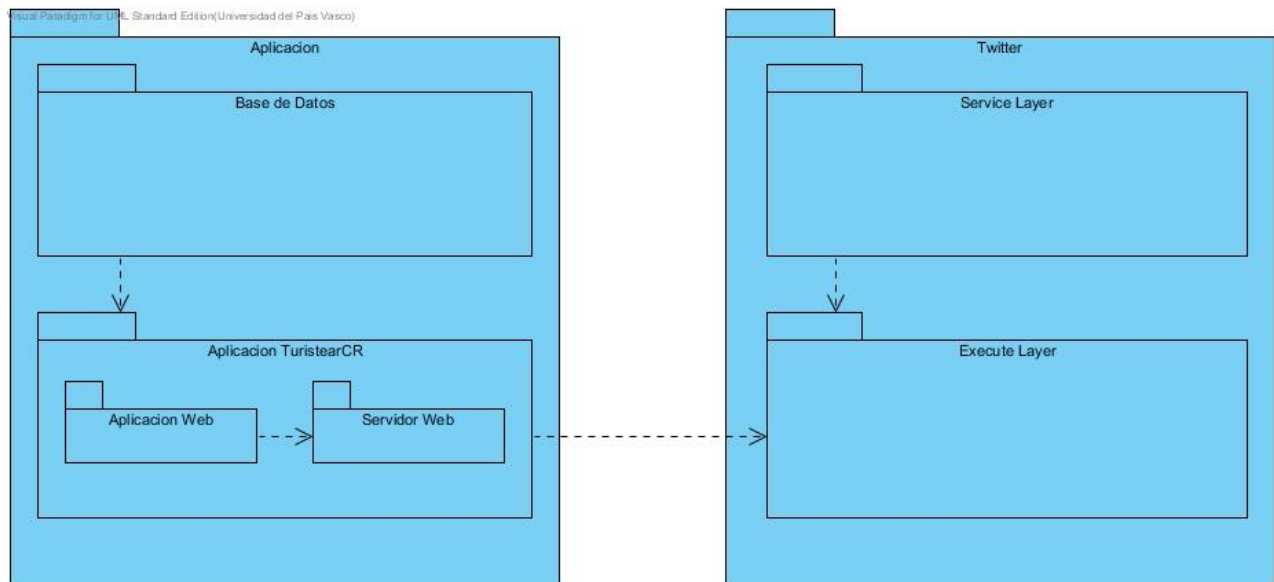


Diagrama de componentes

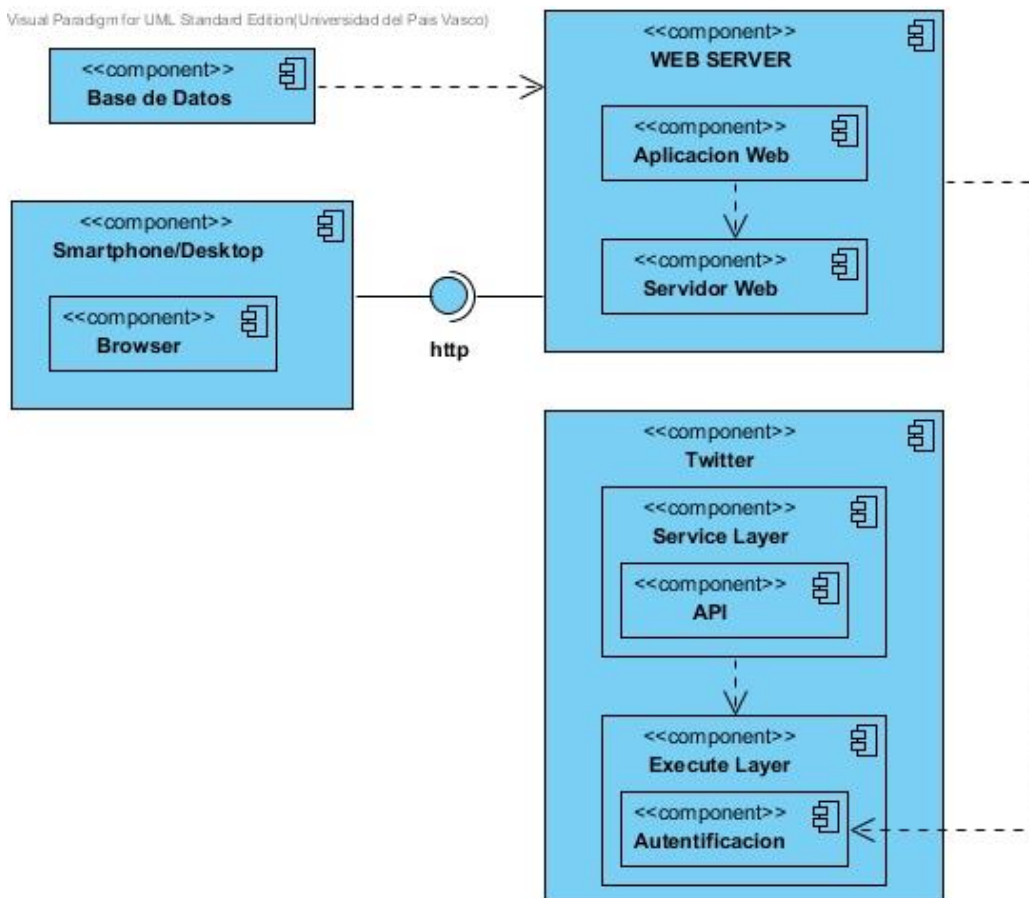
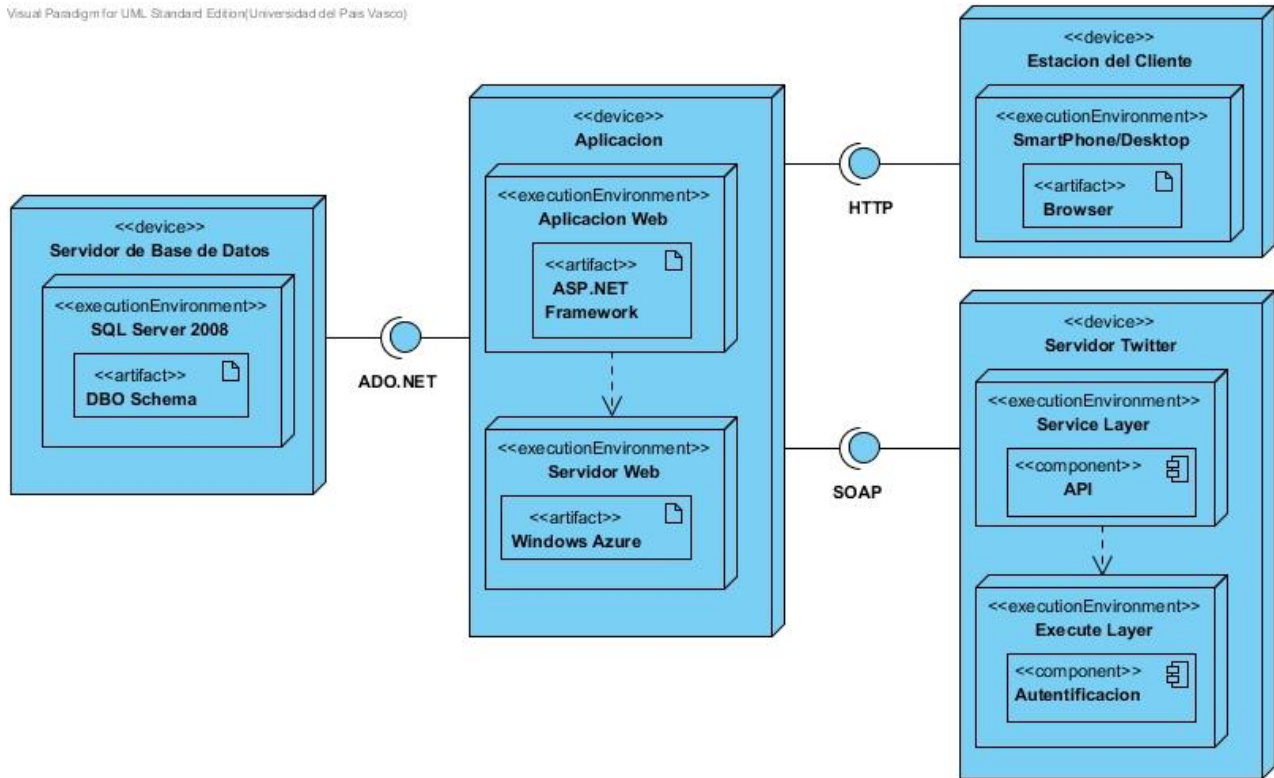
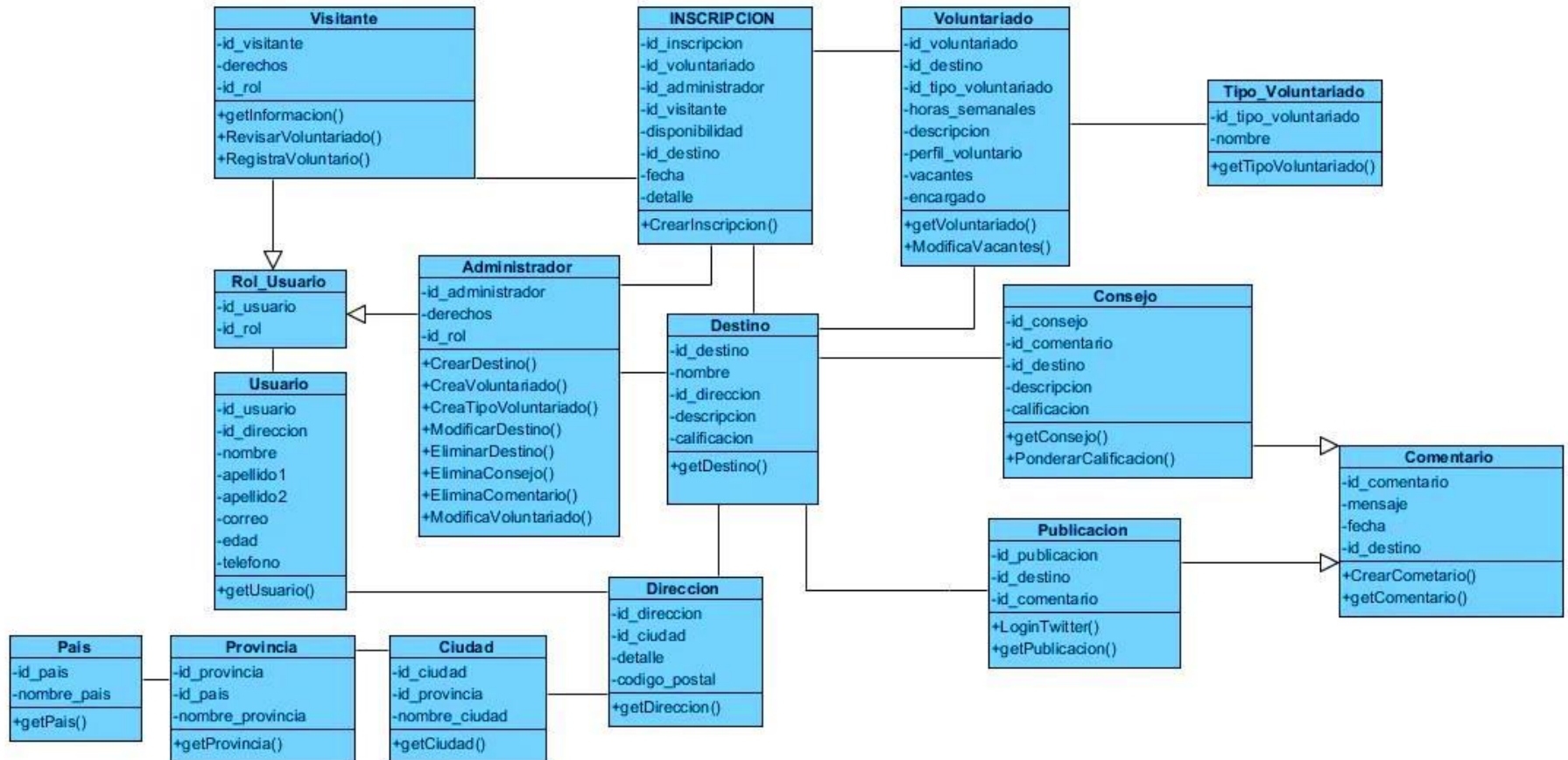


Diagrama de despliegue

Visual Paradigm for UML, Standard Edition (Universidad del País Vasco)

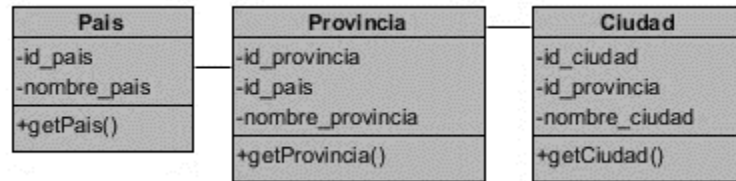


- Descripción detallada (diagrama de clases)



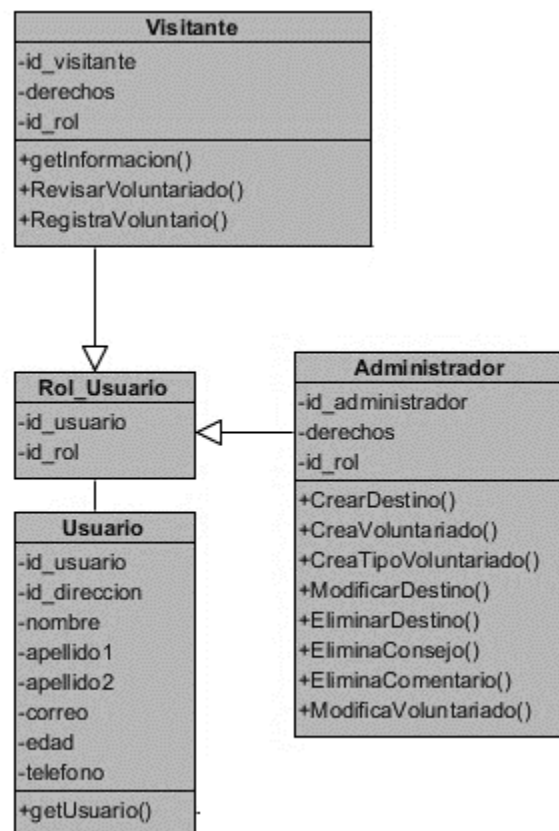
- Justificación de patrones usados

Abstracción/ocurrencia



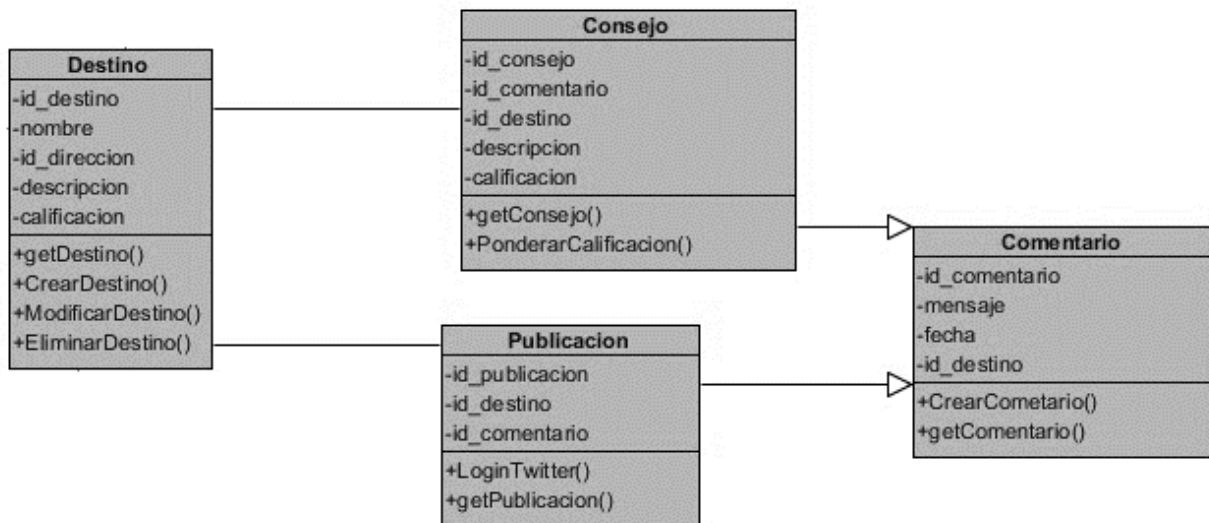
Se elige esta secuencia de entidades para cumplir con este patrón, ya que los objetos comparten información sin embargo a la vez cada uno tiene información diferente. En el diagrama anterior que ha una ciudad ocurre una provincia.

Jugador-rol



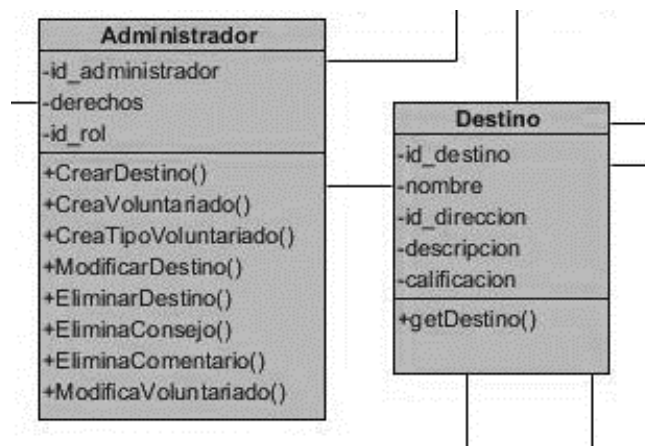
Se elige estas entidades ya que es de suma importancia encapsular información de roles en clases, porque el administrador y el visitante tienen funcionalidades diferentes, para diferentes contextos. Esto evita la herencia múltiple al generarlas en clases.

Observer



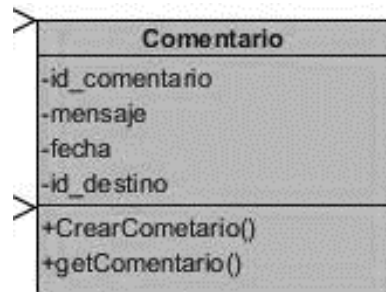
Se elige esta secuencia de entidades para cumplir con este patrón, ya que en este caso cuando se crea un nuevo destino, este debe de estar disponible para que cualquier persona lo vea, y pueda realizar publicaciones con el mismo, además de agregar comentarios y consejos.

Delegado



Se elige esta secuencia de entidades para cumplir con este patrón, ya que en este caso solo el administrador puede crear, modificar o eliminar datos de la otra clase, cumpliendo así la función de delegado la clase **Administrador** y la clase **Destino** como delegador.

Inmutable



Se elige esta secuencia de entidades para cumplir con este patrón, ya que en este caso no se puede modificar un comentario ya hecho.

• Problemas de diseño e implementación

Dentro de las dificultades adquiridas, durante la elaboración de esta documentación, fue la correcta generación de los diagramas, en especial el conceptual y el de clases, ya que requieren un poco más de detalle y tiempo de análisis.

Además tuvimos problemas con el software utilizado para la modelación de los diagramas, ya que era de pago, y no nos permitía utilizar algunas de sus funcionalidades, el software usado se llama, Visual Paradigm UML.

Se tuvo también problemas con la implementación del API REST de Twitter, ya que las versiones con las que estábamos trabajando ya estaban obsoletas, fue hasta que actualizamos las librerías que todo se normalizo.

Otro problema solucionado fue que al trabajar con Windows Azure, este pide una cuenta de evaluación de menos de un mes, y pues requiere ingresar una tarjeta de crédito que posea un mínimo de \$100 de respaldo en ella, a nosotros se nos venció y debimos buscar otra tarjeta para crear una nueva cuenta.

• Interacción con sistemas externos

El sistema podrá interactuar con las siguientes aplicaciones:

- Twitter: El API de Twitter nos ofrece una manera fácil y rápida de hacer crecer la base de usuarios de la aplicación móvil. Permittiéndonos que el usuario pueda generar twitts de los comentarios hechos, o de consejos dados en la aplicación.

- Páginas Web: La aplicación podrá tener acceso a determinadas páginas web de otros usuarios relacionadas con un tema en específico, en nuestro caso, con páginas web de hospedaje.

- **Otros Detalles**

La aplicación se montó sobre el sistema Windows Azure, el cual nos permite alojar nuestra aplicación, y además nos brinda el servicio básico de base de datos. También se utilizó el sistema GitHub, el cual nos sirvió como alojamiento para la documentación de la aplicación, así como para el proyecto.