

1. Introducción

Cada año, millones de personas viajan a distintas partes del mundo para hacer turismo y así recorrer nuevos horizontes. Ir a un lugar desconocido es desafiante si no se posee información sobre la ciudad deseada. Con la herramienta YourCity se busca facilitar al turista la visita a una ciudad desconocida, viajando a través de las rutas fijas que ofrece la aplicación u ofreciéndole la oportunidad de diseñar su propia ruta dinámica. En cada ruta, el turista puede hacer paradas en hitos utilizando las vías que acceden al mismo con el transporte que se ofrezca.

Un turista desea conocer una ciudad nueva y evitar experiencias desagradables, como no encontrar entrada en un hito, perderse en la ciudad o quedarse sin efectivo; por ello YourCity ofrece paquetes que incluyen las entradas a los hitos pertenecientes a una ruta fija así como beneficios como el transporte, comida, visita guiada, entre otros.

En un enfoque técnico, para modelar la base de datos se utiliza el diagrama OMT (Object Modeling Technique) que utiliza metodologías de análisis y diseño orientadas a objetos; dichas características proporcionan más flexibilidad y facilidad de uso lo cual es esencial para que la base de datos perdure en el tiempo. En dicho diagrama se muestran clases que representan a los entes en el universo de discurso, las mismas se asocian entre ellas para reflejar su comportamiento.

2. Descripción de las clases:

- Hito: Es una atracción turística ubicada en un punto geográfico dentro de una ciudad que puede ser visitado por el público general.
 - Lista de atributos:
 - Nombre: Atributo que identifica al Hito.
 - Longitud.
 - Latitud.
 - Altitud.
 - Descripción: información breve sobre un hito.
 - Estado: Atributo que indica si un hito está en funcionamiento o no. El dominio de estado es: disponible, en reparación y clausurado temporalmente.
 - Categoría[0..*]: En este atributo multivaluado se especifica el conjunto de géneros del Hito. Su dominio es: entretenimiento, deporte, cultura, infantil, ciencia, historia, música.
 - Edad apropiada: Edad mínima para poder acceder al hito.
 - Página Web: En caso de existir, este atributo es un link a la página web del hito.
 - Teléfono[0..*]: teléfonos de contacto del hito.
 - Email: correo electrónico asociado al hito.
 - CostoMonedaLocal: precio de la entrada en la moneda del país. El precio puede ser cero.
 - Horarios: Atributo de texto que incluye la información del horario de apertura del hito para cada día. Además incluye información de los días que no trabajan o de aquellos días que tienen un horario diferente.
 - Foto: Contiene la dirección completa del archivo en el que se encuentra la foto en la computadora que funciona como servidor de la aplicación. Es de tipo PATH.
 - Lista de operaciones:
 - calcularCostoEnUSD(relacionLocalUSD): se encarga de calcular el costo del ticket de un hito en dólares americanos; para ello necesita la relación entre la moneda local y el dólar americano.
 - listarEventos(): se encarga de buscar todos los eventos asociados al hito.
 - hitosCategoría(categoría): se encarga de mostrar todos los hitos que pertenezcan a la categoría dada.
 - hitosCategoríaCiudad(categoría, ciudad): filtra todos los hitos según la

categoría y la ciudad dada.

- `hitosEstadoCiudad(estado, ciudad)`: filtra todos los hitos según el estado y la ciudad dada.
 - `hitosPorMenorCosto(ciudad, costo)`: filtra todos los hitos de una ciudad cuyo costo de su ticket sea menor al costo especificado.
 - `hitosPorMayorCosto(ciudad, costo)`: filtra todos los hitos de una ciudad cuyo costo de su ticket sea mayor al costo especificado.
 - `hitosRangoPrecio(ciudad, costoMenor, costoMayor)`: filtra todos los hitos de una ciudad cuyo costo de su ticket se encuentre en el rango entre `costoMenor` y `costoMayor`.
- Vía: Es un camino que conduce de un punto a otro en la ciudad, se utiliza para acceder a los hitos.

- Lista de atributos:

- Nombre: Atributo que identifica a la vía.
- Tipo: Este atributo corresponde a la clasificación de la vía, es decir, su dominio puede ser: calle, avenida, caminería, autopista, elevado, camino, puente.
- TiposTransporte: Este es un atributo multivaluado en el cual se tienen los distintos tipos de vehículos que podrían transitar dicha vía. El dominio del transporte es: a pie, automóvil, autobús, taxi, bicicleta, metro y tren.
- InicioLatitud: Latitud del punto de inicio de una vía.
- InicioLongitud: Longitud del punto de inicio de una vía.
- finLatitud: Latitud del punto final de una vía.
- finLongitud: Longitud del punto final de una vía.

- Lista de operaciones:

- `viasHitosEnCiudad(ciudad)`: retorna todas las vías en una ciudad que son hitos.

- Ruta: Es un recorrido entre dos puntos de una ciudad que incluye hitos y permite recorrer un conjunto de hitos de una manera específica.

- Lista de atributos:

- Nombre: Atributo que identifica a una ruta.
- Descripción: Breve descripción opcional que describe características de la ruta en cuestión.
- Latitud del punto de inicio: atributo numérico que indica la latitud del punto

inicial de una ruta.

- Longitud del punto de inicio: atributo numérico que indica la longitud del punto inicial de una ruta.
- Latitud del punto final: atributo numérico que indica la latitud del punto final de una ruta.
- Longitud del punto final: atributo numérico que indica la longitud del punto final de una ruta.
- Número de visitas: representa el número de personas que han realizado la ruta.
- Fecha de Ingreso: Atributo de fecha que indica cuándo fue creada una ruta en la base de datos.

○ Lista de operaciones:

- hallarDuracionAproximada(): Mediante esta operación se puede hallar la duración aproximada de recorrido de la ruta calculando un promedio que se obtiene de los atributos “horaInicio” y “horaFin” de la asociación “toma” entre usuario y ruta.
- agregarRutaConVias(vias): agrega una ruta con todas las vías en ella.
- rutasCiudadServicio(ciudad, categoriaServicio): filtra las rutas en un ciudad cuyos servicios tengan la categoría especificada.
- rutasCiudadSeguridad(nivelSeguridad): retorna todas las rutas en una ciudad que poseen en nivel de seguridad especificado. El nivel seguridad especificado puede tomar los valores enteros entre uno y cinco.

- Valoración: Es una representación de la opinión de un usuario sobre un hito, ruta o ciudad. El usuario debe colocar de forma obligatoria una puntuación numérica del 1 al 5.

○ Lista de atributos:

- Puntuación: Opinión expresada numéricamente con respecto a la apreciación general del hito, ruta o ciudad visitada por el usuario.
- Seguridad: Opinión expresada numéricamente con respecto a la apreciación de la seguridad en un hito, ruta o ciudad visitada por el usuario.
- Comentario: Espacio de texto para especificar la experiencia del usuario en el hito.
- Relación Precio-valor: Opinión expresada numéricamente con respecto a la relación entre el precio y el valor (calidad de experiencia) que puede tener un hito, ruta o ciudad visitada por el usuario.

- Usuario: Un usuario es la persona a quien se dirige el portal, la cual puede utilizarlo para consultar la información acerca de rutas turísticas urbanas.

- Lista de atributos:

- Alias: Es el identificador del usuario dentro del portal el cual debe ser único.
- Nombres: Es un String que contiene los nombres completos del usuario.
- Apellidos: Es un String que contiene los apellidos completos del usuario.
- Email: Es la dirección de correo electrónico con la que el usuario se registra al portal y debe ser única.
- Género: Indica el sexo del usuario.
- Contraseña: Es un String necesario para hacer login en el portal. La contraseña y el nombre de usuario deben estar relacionados para poder ingresar al sistema.
- EsEstudiante: Un booleano que representa si el usuario es estudiante.
- Fecha de Nacimiento: Se especifica el día, mes y año de nacimiento del usuario.
- Biografía: Es un espacio de texto que el usuario puede rellenar a su gusto.
- Interés: Es un atributo multivaluado que contiene los tipos de hitos que le interesan al usuario. El dominio de un interés puede ser: entretenimiento, deporte, cultura, infantil, ciencia, historia.

- Lista de operaciones:

- calcularEdad(): toma la fecha actual y la resta con la fecha de nacimiento del usuario y retorna la edad.
- esTerceraEdad(): retorna si el usuario ya ha llegado a la tercera edad.
- usuarioContratos(): retorna la cantidad de paquetes contratados por el usuario mediante el portal.

- Restricciones explícitas:

- La fecha actual menos la fecha de nacimiento no puede ser menor de 15 años. Es decir, que un usuario debe tener por lo mínimo 16 años para ser un usuario.

- Ciudad: Una ciudad es una entidad urbana, definida en un área geográfica.

- Lista de atributos:

- Nombre: Atributo que sirve como identificador principal.
- País: Nombre del país al que pertenece una la ciudad.
- NombreMonedaLocal: Este atributo contiene el nombre de la moneda local.

- ValorMonedaLocal: Este atributo tiene el valor de la moneda local en relación al dólar.
- Servicio: Es un conjunto de lugares o actividades destinados a incrementar la satisfacción de un usuario al momento de visitar un hito o tomar una ruta por determinadas vías.
 - Lista de atributos:
 - Nombre: Identifica al servicio, sin embargo, pueden haber varios servicios con un mismo nombre como es el caso de un Mc Donalds.
 - Categoría [0..*]: Con este atributo se puede especificar la finalidad de dicho servicio. El dominio en este caso es el siguiente: salud, entretenimiento, comida, transporte, estación de gasolina, entre otros.
 - Número de contacto [0..*]: En caso de tener, un número telefónico que permita contactar a dicho servicio.
 - Email: En caso de tenerlo, correo electrónico de dicho servicio.
 - PáginaWeb: En caso de tenerlo, página web de dicho servicio.
 - Dirección: dirección del servicio (puede ser no específica).
 - EsGratis: Es un atributo booleano que indica si el servicio es gratuito o no.
 - Descripción: Un campo de texto en el cual se explica brevemente las características de dicho servicio. Como en el caso de un restaurante, mediante este atributo puede explicar brevemente el tipo de comidas que ofrece y el público hacia el cual está dirigido.
- Paquete: es una promoción (mayormente paga) ofrecida en el portal para dar la facilidad al usuario de obtener, mediante una sola compra, entradas en los distintos hitos a visitar, servicio de transporte, comida y guía turístico, entre otros servicios disponibles.
 - Lista de atributos:
 - Nombre: Atributo que identifica un nombre de un paquete. El nombre de un paquete va a ser creativo para atraer la atención de un usuario
 - Precio: Es el monto que paga un usuario estándar. Un usuario estándar es aquel que no es estudiante, no pertenece a la tercera edad y no es un niño.
 - CostoEstudiante: Es precio del paquete que paga un usuario o un acompañante que sea estudiante
 - CostoTerceraEdad: Es el precio que paga un usuario o acompañante que pertenezca a la tercera edad.
 - CostoNiño: Es el precio que paga un acompañante al ser un niño.
 - Descripción: En este atributo se explica brevemente las características que el

paquete incluye.

- Lista de operaciones:
 - `calcularTarifaMinima()`: Esta operación permite calcular el monto básico del paquete para una persona estándar haciendo uso de los precios en dólares de los hitos que se encuentran en este paquete.
- Eventos: Sucesos que se llevan a cabo en un hito determinado.
 - Lista de atributos:
 - **Nombre:** Atributo que identifica al evento.
 - **Categoría:** Este atributo clasifica al evento dentro de un dominio específico. Dicho dominio es el siguiente: musical, entretenimiento, cultural, histórico, social, festivo.
 - **CostoMonedaLocal:** Indica el precio de dicho evento en la moneda local
 - **Descripción:** atributo que explica brevemente las características de un evento.
 - Lista de operaciones:
 - `calcularCostoEnUSD(relacionLocalUSD)`: se encarga de calcular el costo de entrada del evento en dólares americanos; para ello necesita la relación entre la moneda local y el dólar americano.

3. Descripción de asociaciones

- Contiene (Hito - contiene - Hito): Asociación recursiva que indica la contención de un hito dentro de otro hito.
 - Roles: En este caso se poseen dos roles: Externo e interno. El primero corresponde a un Hito padre que contiene otros Hitos internos.
 - Cardinalidades: Para cada Hito con rol externo se pueden tener 0..* Hitos internos. Y para cada Hito interno se puede tener 1 Hito externo, es decir, puede tener solo un padre.
 - Ejemplo de instancia:
 - `Contiene = {(USB, BibliotecaUSB), (USB, Cromovegetal)}`
 - Restricciones explícitas:
 - Un Hito no se contiene a sí mismo, es decir, los nombres tanto de Hito interno como del externo no deben coincidir.

- Contrata (Usuario - compra - Paquete): Esta asociación indica que un usuario puede comprar algún paquete.
 - Lista de atributos: Tiene un atributo multivaluado llamado “acompañante” en el cual se especifica el número de acompañantes y el tipo de acompañante. Por ejemplo: 3 niños, 2 normales, 4 tercera edad, 1 estudiante.
 - Acompañante:
 - Tipo: Atributo que indica qué usuario es, es decir, puede tomar valores del siguiente dominio: niño, tercera edad, estudiante, normal.
 - Número: Atributo numérico que indica la cantidad de personas por tipo.
 - Lista de operaciones:
 - CalcularPrecioTotal(): Esta operación se calcula a partir del número de acompañantes y su tipo y el precio para cada tipo de acompañante.
 - Cardinalidades: Un usuario puede contratar 0..* paquetes, y un paquete puede ser contratado por 0..* usuarios.
 - Ejemplo de instancia:
 - Contrata = {(Luis, “Tour de los museos de arte Parisinos”, {(niño,3), (estudiante,1), (tercera edad,2), (normal,3)})}

- Crea (Usuario - crea - Dinámica): Esta asociación indica que un usuario puede crear una ruta dinámica.
 - Cardinalidades: Un usuario puede crear entre 0..* rutas dinámicas, y una ruta dinámica es creada únicamente por 1 usuario.
 - Ejemplo de instancia:
 - Crea = {(Luis, “Paseo por mis museos favoritos de Berlín”)}

- Es (Vía - Es - Hito): Esta asociación indica si una vía es un Hito o no.
 - Cardinalidades: Una vía puede ser o no ser un Hito es decir una vía es 0..1 veces un Hito, y un Hito es 0..1 veces una vía, es decir, un Hito puede ser o no ser una vía.
 - Ejemplo de instancia:
 - Es = {(Puente de San Francisco, Puente de San Francisco)}
 - Restricciones explícitas:
 - Para que una Vía sea un Hito, el nombre de ambas clases debe coincidir.

- EstaEn (Ruta - estaEn - Ciudad): Esta asociación indica si una ruta esta en una ciudad.
 - Cardinalidades: Una Ruta puede estar en 1 Ciudad y en una Ciudad pueden estar

1..* Rutas.

- Ejemplo de instancia:
 - EstaEn = {(Ruta de museos de Guerra, Berlin), (Ruta de teatros de la ópera de Berlín, Berlín)}
- Incluye (Paquete - incluye - Fija): Esta asociación representa que un Paquete incluye Rutas fijas.
 - Cardinalidades: Un Paquete puede incluir 1 Ruta fija, y una Ruta fija puede estar incluidas en 0..* Paquetes.
 - Ejemplo de instancia:
 - Incluye = {(“Super recorrido por plazas con comida y bebida incluida”, “Tour por las plazas de Berlín”)}
- Ocurre (Evento - ocurre - Hito): Esta asociación indica que los eventos deben ocurrir en Hitos y que en los Hitos pueden o no haber Eventos.
 - Lista de atributos: Esta asociación tiene como atributos la fecha inicio del evento y la fecha fin.
 - FechaInicio: Fecha en la que inicia dicho evento.
 - FechaFin: Fecha en la que termina el evento. Esta fecha podría no estar especificada ya que podría no conocerse cuando será el fin de dicho evento.
 - Cardinalidades: Un Evento puede ocurrir únicamente en 1 Hito, sin embargo, en un Hito pueden ocurrir 0..* Eventos.
 - Ejemplo de instancia:
 - Ocurre = {(“Festival de verano”, “Plaza Bolívar”)}
 - Restricciones explícitas:
 - Un evento no puede ocurrir en un Hito cuyo estado no sea “Disponible”
 - La fecha inicio no puede ser mayor que la fecha fin.
- RutaEn (Ruta - rutaEn - Hito): Con esta asociación lo que se busca representar es que dentro de un Hito pueden haber a su vez Rutas ya que un Hito puede tener Hitos y pueden haber diversas maneras de recorrer dichos Hitos internos.
 - Cardinalidades: Un Hito puede tener 0..* Rutas dentro de sí, y una Ruta puede estar 0..1 veces en un Hito, esto es, una Ruta puede estar o no estar dentro de un Hito. Si la Ruta está dentro del Hito no puede estar en otros Hitos ya que consideraría como una Ruta interna particular de un Hito.
 - Ejemplo de instancia:
 - RutaEn = {(“Paseo por los jardines de la USB”, USB)}

- SeAccedePor (Hito - SeAccedePor - Vía): Esta asociación indica que un Hito puede ser accedido mediante ciertas vías.
 - Cardinalidades: Un Hito se puede acceder por 1..*Vías, y una vía puede servir de acceso a 0..* Hitos.
 - Ejemplo de instancia:
 - SeAccedePor = {(Plaza Francia, Avenida Luis Roche)}
- SeEncuentra (Vía - seEncuentra - Ciudad): Esta asociación indica las vías que se encuentran dentro de una ciudad.
 - Cardinalidades: Una ciudad puede tener 1..* Vías, mientras que una Vía puede encontrarse solo en una Ciudad.
 - Ejemplo de instancia:
 - SeEncuentra = {(“Avenida las Delicias”, “Maracay”)}
- Tiene (Hito - tiene - Servicio): Esta asociación sirve para explicar si un Hito tiene o no Servicios asociados a él. Como por ejemplo el servicio de restaurante dentro de un museo.
 - Cardinalidades: Un Hito puede tener 0..* Servicios asociados a él, mientras que un mismo servicio puede estar asociado también a un hito.
 - Ejemplo de instancia:
 - Tiene = {(Museo, Restaurante), (Museo, Baño)}
- TieneServicio (Vía - tiene - Servicio): Esta asociación representa aquellos servicios que se encuentran en medio de una vía o se acceden por medio de ella.
 - Cardinalidades: Una Vía tiene entre 0..* Servicios y un Servicio puede estar en una vía.
 - Ejemplo de instancia:
 - TieneServicio = {(“Avenida las Delicias”, Gasolinera PDV)}
- Toma (Usuario -toma- Ruta): Aquí van a haber unos atributos que indiquen hora de inicio y hora de fin del recorrido de la ruta.
 - Lista de atributos:
 - EsAprobada: Atributo que permite al usuario que toma la Ruta, sólo si es una ruta dinámica, aprobarla para que esta pueda ser promovida como una ruta fija.
 - Fecha: Indica la fecha en la que el usuario tomó la ruta.

- HoraInicio: Atributo que indica la hora en la que el Usuario tomó dicha Ruta.
 - HoraFin: Atributo que indica la hora en la que el Usuario finalizó dicha Ruta.
 - Cardinalidades: Un Usuario puede tomar entre 0..* Rutas, y una Ruta puede ser tomada por 0..* Usuarios.
 - Ejemplo de instancia:
 - Toma = {(Luis, “Los teatros favoritos de Teresa”, TRUE, 7:00 AM, 8:00 PM)}
 - Restricciones explícitas:
 - Un usuario puede aprobar una ruta sólo si esta ruta es dinámica, de lo contrario, el valor por defecto para el atributo “EsAprobada” sera FALSE en caso de que sea una ruta fija y no tendrá efecto alguno.
 - La HoraFin debe ser mayor que la HoraInicio.
- ValoraCiudad (Ternaria entre: Usuario, valoración, Ciudad): Un Usuario puede valorar una ciudad.
 - Cardinalidades:
 - Un Usuario con una Ciudad puede realizar 0..* valoraciones, ya que puede no valorarla pero también puede que la visite varias veces y dé una valoración nueva.
 - Una Valoración de una Ciudad puede haber sido realizada por 1 solo Usuario.
 - Un Usuario que da una Valoración puede corresponder sólo a 1 Ciudad.
 - Ejemplo de instancia:
 - ValoraCiudad = {(Luis, 5, 5 ,5, “Es la ciudad más hermosa que he conocido”, Venecia)} (Los valores numéricos corresponden a las valoraciones que el usuario dio sobre la ciudad)
 - ValoraHito (Ternaria entre: Usuario, valoración,Hito): Un Usuario puede valorar un Hito.
 - Cardinalidades:
 - Un Usuario con un Hito puede realizar 0..* valoraciones, ya que puede no valorar el Hito pero también puede valorarlo varias veces en distintos momentos (ya que el Hito podría mejorar o empeorar).
 - Una Valoración de un Hito puede haber sido realizada por 1 solo Usuario.
 - Un Usuario que da una Valoración puede corresponder sólo a 1 Hito.
 - Ejemplo de instancia:
 - ValoraCiudad = {(Luis, 3, 5 ,3, “Esta plaza realmente no es muy segura, pero al menos es gratuita, sin embargo esta muy descuidada”, Plaza

Francia))}

- ValoraRuta (Ternaria entre: Usuario, valoración, Ruta): Un Usuario puede valorar una Ruta.
 - Cardinalidades:
 - Un Usuario con una Ruta puede realizar 0..* valoraciones, ya que puede no valorar la Ruta, sin embargo, puede valorarlo varias veces en distintos momentos (ya que la Ruta podría mejorar o empeorar).
 - Una Valoración de una Ruta puede haber sido realizada por 1 solo Usuario.
 - Un Usuario que da una Valoración puede corresponder sólo a 1 Ruta.
 - Ejemplo de instancia:
 - ValoraCiudad = {(Luis, 5, 3 ,4, “Esta Ruta fue muy divertida, sin embargo, algunos de los museos fueron realmente costosos”, “Ruta de los museos de arte moderno”)}

4. Descripción de agregaciones:

- Agregación entre Ruta e Hito: Esta asociación de agregación indica que una ruta esta compuesta de un conjunto de hitos en un orden en específico. En particular este orden define el recorrido de una ruta. Aunque un hito es parte de una ruta, dicho Hito existe aunque no esté en una Ruta.
 - Cardinalidades: Una ruta puede tener entre 1..* hitos, y un hito puede pertenecer a 0..* rutas.
 - Ejemplo de instancia:
 - {(“Ruta Amigo Deportista”, “UCV”), (“Ruta Amigo Deportista”, “Los Próceres”)}
- Agregación entre Ruta y vía: Esta asociación de agregación indica que una ruta esta compuesta de un conjunto de vías.
 - Cardinalidades: Una ruta puede tener entre 1..* vías, y una vía puede pertenecer a 0..* rutas.
 - Ejemplo de instancia:
 - {(“Ruta Playera”, “Carretera de la Guaira”), (“Ruta Playera”, “Taguao”)}

5. Descripción de Generalizaciones:

- Ruta dinámica: es una especialización de ruta. Una ruta es dinámica cuando el usuario decide cómo recorrerla y que hitos visitar. Un usuario puede proponerla como fija y de esta forma otros usuarios pueden tomarla y aprobarla. Dependiendo del número de aprobaciones que tenga, puede volverse fija.
 - Lista de atributos:
 - **PropuestaFija**: Este atributo indica si la ruta dinámica hecha por un usuario es propuesta para ser una ruta fija. En el caso de ser propuesta, debe analizarse el número de aprobaciones que tiene por parte de otros usuarios que la recorran.
 - Lista de operaciones:
 - **calcularAprobaciones()**: NUMERIC
- Ruta Fija: Ruta fija es una especialización de ruta. Una ruta es fija cuando esta ya está preestablecida con Hitos y Vías incluidas.

6. Decisiones de Diseño

1. Al momento de trabajar en la clase Ruta se tenía que modelar que las rutas podían ser fijas o dinámicas y sus interacciones con los otros entes del UD. Para lograrlo surgieron dos opciones:
 - a. Crear la clase Ruta con el atributo tipo cuyo dominio será “fija” y “dinámica”. Luego al tener la asociación “crea” entre Ruta y Usuario hubiera sido necesaria una restricción explícita, dado que un usuario solo puede crear rutas dinámicas. También, la asociación 'incluye' entre Ruta y Paquete hubiese necesitado una restricción dado que un paquete sólo puede incluir una ruta fija.
 - b. Crear la clase Ruta y especializarla con las clases Fija y Dinámica. Luego crear las asociaciones: 'crea' entre Usuario y Dinámica; 'incluye' entre Paquete y Dinámica. Se decidió tomar la segunda opción ya que se considera un diseño más expresivo porque representa a las rutas fijas y dinámicas con la importancia que tienen, tomando en cuenta que las mismas tienen un comportamiento individual.
2. Con respecto a la asociación valoración, se tenía que modelar la relación con Hito, Ciudad y Usuario. Al principio se modelaron las siguientes relaciones binarias: Valoración-Hito, Valoración-Ciudad, Valoración-Ruta y Usuario-Valoración. Luego de un análisis, se decidió diseñar las siguientes relaciones ternarias: Usuario-Valoración-Hito, Usuario-

Valoración- Ciudad y Usuario- Valoración - Ruta. Con esa decisión , es mas intuitivo comprender las relaciones.

3. La especialización de paquete trajo algunas decisiones importantes en el diseño. Un paquete relacionado con un usuario debe tomar en cuenta si el usuario desea visitar una ciudad con sus acompañantes. Estos acompañantes pueden ser niños, personas pertenecientes a la tercera edad o usuario estándar. Acompañante es un atributo de la asociación 'contrata' y esta diseñado para obtener el número y el tipo de acompañante que tiene un usuario, y de esta forma poder calcular un paquete con un precio especial para el usuario y sus acompañantes.
4. Las rutas están conformadas por Hitos y Vías en un enfoque conceptual, y para modelar esto se comenzó creando una asociación entre Ruta e Hito y otra entre Ruta y Via. Luego se consideró que era más adecuado utilizar una agregación para ambos casos ya que hace que el modelo sea más autoexplicativo y expresivo dado que una ruta no está simplemente asociada a un hito o a una vía, sino que los mismo son parte de ella.

8. Conclusión

Una vez realizado el diagrama OMT (Object Modeling Technique), puede verse de forma clara y concisa cómo se comportan los entes en el universo de discurso. El diagrama representa un modelo de análisis y diseño de la base de datos YourCity. Además, se puede ver los atributos y las operaciones de cada clase y de cada asociación, los cuales juegan un papel fundamental en los requerimientos de la aplicación.

Durante la elaboración de este documento y el diseño del esquema conceptual OMT se presentaron una serie de dificultades al momento de realizar las restricciones explícitas y modelar de forma adecuada las clases y las asociaciones. Entre las dificultades principales que surgieron destaca la selección de como utilizar las herramientas del OMT para cumplir con las normas de calidad y simplificar el diagrama.

Como recomendaciones finales para futuras mejoras del diagrama OMT se recomienda mantener el portal lo más simple posible. Es importante tener en cuenta que lo más relevante en este trabajo es la realización de un diseño conceptual que cumpla con todos los requerimientos del universo de discurso y con las normas de calidad.

9. Bibliografía

[1] Soraya Abad-Mota, Lineamientos sobre cómo escribir informes técnicos, mayo 2007.

[2] Soraya Abad-Mota, La Base de Datos Musical: Esquema Conceptual en el Modelo ER-E (Cuarta Edición). Abril 2013.

[3] Soraya Abad-Mota, Paradigmas de Modelado de BD I (CI-5311): Caso de Estudio del Proyecto. Rutas Turísticas Urbanas (RTU), 24 de abril de 2013.

[4] Michael Blaha and William Premerlani, Object-Oriented Modeling and Design for Database Applications. Prentice Hall, Upper Saddle River, New Yersey 07458. // FALTA EL ANO AQUI