

### 2.2.1 Asociación, agregación y composición

Ya definidas las relaciones de asociación, agregación y composición vamos a incluir un algoritmo definido dentro de la creación de un pequeño proyecto para tomar las clases de la descriptiva del proyecto y crear los diagramas de clases con las relaciones entre ellas.

#### 2.2.1.1 La descriptiva del problema

**Proyecto:**

Sistema de Reservaciones de Vuelos SRV

**Objetivo:**

El SRV permitirá al usuario hacer consultas y reservaciones de vuelos, además de comprar boletos de forma remota, sin la necesidad de recurrir a un agente de viajes.

**Equipo de Desarrollo:**

El profesor.

**Introducción:**

En la actualidad existen múltiples sistemas de reservaciones de vuelos que emplean las agencias de viajes para dar servicio a los clientes, algunos de ellos son: Sabre [1], Galileo [2], Worldspan [3] y Amadeus [4, 5], todos ellos de distribución global. También existen algunos sistemas que trabajan en internet como Travelocity [6] y Expedia [7] los cuales se basan en gran medida en los mencionados anteriormente.

**Descripción del problema:**

El SRV permite al usuario hacer consultas y reservaciones de vuelos, además de comprar los boletos aéreos de forma remota sin recurrir a una agencia de viajes, se desea que el sistema tenga acceso desde un servidor por medio de conexión punto a punto (Cliente-Servidor) y si el alcance lo permite que también trabaje en Internet.

El sistema deberá presentar en su pantalla principal la bienvenida al usuario, haciendo una descriptiva de los servicios ofrecidos junto con la opción de registrarse por primera vez, y si ya se está registrado, poder emplear el sistema de reserva de vuelos esto se hará por medio de un acceso de login especificando un nombre de usuario y una clave de acceso los cuales deberán validarse.

Una vez registrado el usuario y después de haberse validado con el registro de usuarios se pueden seleccionar las siguientes acciones:

- Consulta de vuelos.
- Reservación de vuelos.
- Pago de boletos.

La consulta de vuelos se puede hacer de tres formas distintas

1. Horario de vuelos
2. Tarifa de vuelos
3. Estado de vuelos

La consulta según horario muestra los horarios de las diferentes aerolíneas que dan servicio entre dos ciudades

La consulta según tarifas muestra los diferentes vuelos entre dos ciudades que dan prioridad a su costo.

La información de vuelo se utiliza principalmente para consultar el estado de algún vuelo, incluyendo información de disponibilidad de asientos y en el caso de un vuelo para el mismo día, si esta a tiempo. Se pueden incluir preferencias en las búsquedas, como fecha y horario deseado, categoría de asiento, aerolínea y si se desea vuelos directos.

La reservación de vuelo permite al cliente hacer una reservación para un vuelo particular, especificando la fecha y el horario bajo una tarifa establecida. Es posible reservar un itinerario compuesto de múltiples vuelos para uno o más pasajeros, además de poder reservar asientos. El pago permite al cliente, dada una reservación de vuelo previa y una tarjeta de crédito válida, adquirir los boletos aéreos.

Los boletos serán enviados al cliente posteriormente o estarán listos para ser recogidos en el mostrador del aeropuerto antes de la salida del primer vuelo.

Es necesario estar previamente registrado con un número de tarjeta de crédito válida para poder hacer compras de boletos o de lo contrario proveerla en el momento de la compra.

Además de los servicios de vuelo, el usuario podrá en cualquier momento, acceder, modificar o cancelar su propio registro, todo esto después de haber sido validado en el sistema.

**NOTA:** Esta es una descripción limitada que se irá refinando conforme el sistema se vaya desarrollando.

#### **Referencias:**

- [1] <http://es.wikipedia.org/wiki/Sabre>
- [2] [http://en.wikipedia.org/wiki/Galileo\\_CRS](http://en.wikipedia.org/wiki/Galileo_CRS)
- [3] <http://en.wikipedia.org/wiki/Worldspan>
- [4] <http://www.amadeus.net/plnext/meb/HomePageDispatcher.action?SITE=BCEUBCEU&LANGUAGE=GB>
- [5] [http://en.wikipedia.org/wiki/Amadeus\\_CRS](http://en.wikipedia.org/wiki/Amadeus_CRS)
- [6] [http://www.travelocity.com.mx/mx?gclid=CL\\_Whb\\_pg7MCFSWoPAodP2EAKw](http://www.travelocity.com.mx/mx?gclid=CL_Whb_pg7MCFSWoPAodP2EAKw)
- [7] <http://www.expedia.mx>

#### **2.2.1.2 Identificación de clases**

La identificación de clases se lleva a cabo dentro del dominio del problema, empleando un documento textual que describa el sistema tal como se hizo anteriormente. El proceso comienza a partir de la identificación de clases candidatas explícitas o implícitas a las que se refiera la descripción del problema, se pueden emplear las siguientes consideraciones:

- I. Los sustantivos (nombres: comunes, propios, concretos, abstractos, contables, incontables, individuales, colectivos) de la descripción del problema son los posibles candidatos a clases de objetos.
- II. Se deben identificar identidades (Clase == Módulo) físicas al igual que las conceptuales.
- III. No se debe diferenciar entre clases y atributos (TODO ES UNA CLASE).
- IV. Dado que no todas las clases se describen de manera explícita, algunas son implícitas en la aplicación con lo cuál deberán añadirse las clases que puedan ser identificadas por el conocimiento que se tenga en el área en cuestión.
- V. Se deben revisar los pronombres (personales, demostrativos, relativos, interrogativos exclamativos, indefinidos, numerales) en la descripción del problema para asegurar que no

se haya perdido ningún sustantivo descrito de forma implícita.

VI. Para facilitar la identificación de clases se subrayan todos los sustantivos y pronombres de la descripción del problema.

Así que, aplicando las consideraciones anteriores a la descriptiva, queda de la siguiente forma:

El **SRV** permite al **usuario** hacer **consultas** y **reservaciones** de **vuelos**, además de comprar los **boletos aéreos** de forma remota sin recurrir a una **agencia de viajes**, se desea que el sistema tenga acceso desde un servidor por medio de **conexión** punto a punto (**Cliente-Servidor**) y si el alcance lo permite que también trabaje en **Internet**.

El **sistema** deberá presentar en su **pantalla principal** la **bienvenida** al **usuario**, haciendo una descriptiva de los **servicios** ofrecidos junto con la **opción** de registrarse por primera vez, y si ya se está registrado, poder emplear el sistema de reserva de **vuelos** esto se hará por medio de un **acceso** de **login** especificando un **nombre de usuario** y una **clave de acceso** los cuales deberán validarse.

Una vez registrado el **usuario** y después de haberse validado con el **registro** de usuarios se pueden seleccionar las siguientes acciones:

- **Consulta de vuelos.**
- **Reservación de vuelos.**
- **Pago de boletos.**

La **consulta** de **vuelos** se puede hacer de tres formas distintas

1. **Horario de vuelos.**
2. **Tarifa de vuelos.**
3. **Estado de vuelos.**

La **consulta** según **horario** muestra los **horarios** de las diferentes **aerolíneas** que dan **servicio** entre dos **ciudades**.

La **consulta** según **tarifas** muestra los diferentes **vuelos** entre dos **ciudades** que dan prioridad a su **costo**.

La **información de vuelo** se utiliza principalmente para consultar el estado de algún vuelo, incluyendo **información** de **disponibilidad** de **asientos** y en el caso de un **vuelo** para el mismo **día**, si esta a **tiempo**.

Se pueden incluir preferencias en las **búsquedas**, como **fecha** y **horario** deseado, **categoría de asiento**, **aerolínea** y si se desea **vuelos directos**.

La **reservación de vuelo** permite al **cliente** hacer una **reservación** para un **vuelo** particular, especificando la **fecha** y el **horario** bajo una **tarifa** establecida. Es posible reservar un **itinerario** compuesto de múltiples **vuelos** para uno o más **pasajeros**, además de poder reservar **asientos**.

El **pago** permite al **cliente**, dada una **reservación de vuelo** previa y una **tarjeta de crédito** válida, adquirir los **boletos aéreos**.

Los **boletos** serán enviados al **cliente** posteriormente o estarán listos para ser recogidos en el **mostrador del aeropuerto** antes de la **salida** del primer **vuelo**.

Es necesario estar previamente registrado con un **número de tarjeta de crédito** válida para poder hacer **compras de boletos** o de lo contrario proveerla en el momento de la **compra**.

Además de los **servicios de vuelo**, el **usuario** podrá en cualquier momento, acceder, modificar o cancelar su propio **registro**, todo esto después de haber sido validado en el **sistema**.

### **2.2.1.3 Selección de las clases**

A partir de la identificación de clases se puede partir tomando en cuenta las siguientes consideraciones:

- I. Todas las clases deberán tener sentido en el área de aplicación, la relevancia del problema debe ser el único criterio para la selección.
- II. Como regla general se debe escoger los nombres de las clases con cuidado, que no sean ambiguos y que mejor describan al problema. Los nombres deben ser establecidos con un formato consistente (por lo regular nombres propios, en singular y sin guiones bajos o caracteres especiales como acento (en vocal) o tilde (en consonante)).
- III. Durante esta etapa no hay que preocuparse por las relaciones entre las clases (Herencia simple, Herencia múltiple, Composición, Agregación y Asociación), lo importante es elegir las clases adecuadas.
- IV. Ante la duda conserve las clases, luego si no tienen injerencia en el sistema pueden eliminarse.
- V. Eliminar clases redundantes si expresan la misma información. La clase más descriptiva deberá ser guardada.
- VI. Eliminar clases irrelevantes, que tienen poco o nada que ver con el problema. Esto requiere de juicio en el contexto y hay que evaluar la clase para ver relevancia.
- VII. Se deben identificar las clases imprecisas (eliminar). Algunas pueden no tener límites definidos, estar mal definidos o ser demasiado generales.
- VIII. Es necesario eliminar las clases que debieran ser atributos más que clases evaluando si son primitivos o compuestos (DUDA RAZONABLE CONSERVAR).
- IX. Eliminar las clases que parecen ser roles más que clases, cuando los nombres corresponden a la función que tienen las clases mas que a entidades independientes.
- X. Suprimir las clases que debieran ser operaciones mas que clases, si las entidades representan operaciones que se aplican a los objetos y no a entidades manipuladas por si mismas.
- XI. Hay que eliminar las clases que se parecen más a una subrutina, listas, arreglos, etc.
- XII. Se deben eliminar las clases que correspondan a aspectos de interfaces de usuario (GUI) y no de la aplicación.
- XIII. Se deben eliminar las clases que correspondan a todo un sistema completo.
- XIV. Suprimir las clases que correspondan más a actores del sistema (usuarios del sistema)
- XV. Agregar las clases implícitas que no aparezcan en la descripción del problema.

#### **2.2.1.3.1 Clasificación de las clases.**

**Clases redundantes:**

Cliente, usuario y pasajero ambos son redundantes o equivalentes, Pasajero es más descriptivo por ser la persona que utiliza el sistema.

#### **Clases irrelevantes:**

Mostrador del Aeropuerto, Agencia de Viajes y Boleto Aéreo.

#### **Clases imprecisas:**

Sistema, Servicios, Actividad, Preferencia, Búsqueda, Información, Estado, Disponibilidad, Opción, Acceso e Itinerario son clases imprecisas. Puede ser necesario incluir herencia para compartir aspectos comunes a estas clases.

#### **Nombres de clases:**

Aeropuerto en lugar de Ciudad

#### **Clases que son atributos:**

Número de tarjeta de Crédito es un atributo de Tarjeta de Crédito, Categoría de Asiento lo es de Asiento, Información de Vuelo de Vuelo, así como Horario de Vuelo.

#### **Clases que son operaciones:**

Consulta, Pago y Reservaciones.

#### **Clases de interfaces graficas de usuario:**

Mensaje de Bienvenida y Pantalla Principal

#### **Clases de sistema completo:**

SRV (Sistema de reservaciones de vuelo) SistemaDeReservacionDeVuelo { main() }

#### **Clases de actores:**

Cliente (NO), Usuario (NO), Pasajero (Clase)

#### **2.2.1.3.2 Crear una tabla de clases candidatas y otra de clases clasificadas (Clases identificadas)**

La tabla de clases candidatas se muestra a continuación con las siguientes consideraciones  
SC = Sistema Completo, I = Imprecisa, A = Atributo (Estos se conservan para incluirlos en alguna clase si son primitivos o si son compuestos será otra Clase), R = Redundante, Ac = Actor, O = Operación (Estos pueden ser funciones (métodos) de alguna clase), GUI = Interface grafica de usuario,

<b>Clases candidatas</b>	<b>Modificación</b>	<b>Clases candidatas</b>	<b>Modificación</b>
SRV	Eliminar (SC)	Informacion de vuelo	Eliminar (A)
Sistema	Eliminar (I)	Horario	Eliminar (R)
Usuario	Eliminar (Ac)	Aerolinea	<a href="#">Aerolinea (6)</a>
Consulta	Eliminar (O)	Ciudad (A)	<a href="#">Aeropuerto (7)</a>
Reserva	Eliminar (O)	Tarifa	Eliminar (R)
Vuelo	<a href="#">Vuelo (1)</a>	Costo	Eliminar (R)
Boleto aereo	Eliminar (I)	Estado	Eliminar (I)

Agencia de viajes	Eliminar (I)	Disponibilidad	Eliminar (I)
Sistema de reservaciones	Eliminar (SC)	Informacion	Eliminar (I)
Internet	Eliminar (SC o I)	Asiento	Asiento (8)
<b>Clases candidatas</b>	<b>Modificación</b>	<b>Clases candidatas</b>	<b>Modificación</b>
Pantalla principal	Eliminar (GUI)	Dia	Eliminar (A)
Mensaje de bienvenida	Eliminar (GUI)	Hora	Eliminar (A)
Servicios	Eliminar (I)	Preferencia	Eliminar (I)
Opcion	Eliminar (I)	Busqueda	Eliminar (O)
Reservacion de vuelos	Reservacion (2)	Fecha	Eliminar (A)
Acceso	Eliminar (I)	Categoria de asiento	Eliminar (A)
Login	Eliminar (A)	Vuelo directo	Eliminar (A)
email	Eliminar (A)	Cliente	Eliminar (Ac)
Clave de acceso	Eliminar (A)	Itinerario	Eliminar (I)
Registro	RegistroUsuario (3)	Pasajero	Pasajero (9)
Actividad	Eliminar (I)	Compra	Compra (10)????
Consulta de vuelos	Eliminar (O)	Tarjeta de credito	RegistroTarjeta (11)
Pago de boletos	Eliminar (O)	Boleto	Eliminar (I)
Horario de vuelos	Horario (4)	Mostrador del aeropuerto	Eliminar (I)
Tarifa de vuelos	Tarifa (5)	Numero de tarjeta de credito	Eliminar (A)

La tabla de clases identificadas se muestra a continuación, Im = Clase Implícita, estas clases pertenecen a la **lógica del negocio del sistema**.

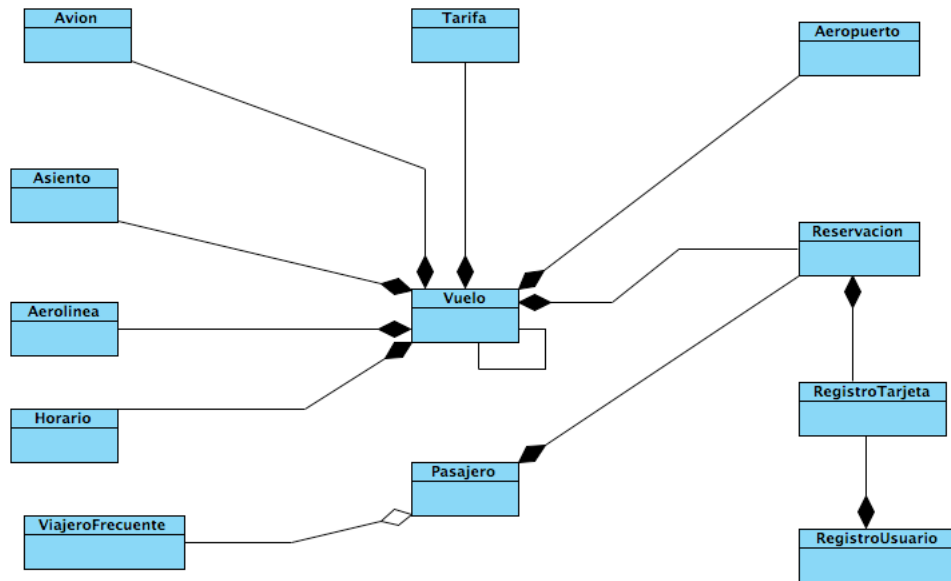
Clase identificada	Clase identificada	Clase identificada	Clase identificada
Vuelo	Tarifa	Pasajero	ViajeroFrecuente (Im)
Reservacion	Aerolinea	Compra???	
RegistroUsuario	Aeropuerto	RegistroTarjeta	
Horario	Asiento	Avion (Im)	

#### 2.2.1.4 Aplicar las relaciones entre clases para hacer el diagrama de clases

Hay que describir en forma oral la asociación que hay entre clases, es decir hacer una oración que implique a dos clases diferentes para verificar si hay alguna relación entre ellas. La clase esta entre paréntesis en la oración, recordar que aunque están definidas en el idioma español no deben llevar acentos ni tildes.

- Un vuelo (Vuelo) requiere de reservaciones (Reservacion)
- Un vuelo (Vuelo) se dirige a un aeropuerto (Aeropuerto)
- Un vuelo (Vuelo) tiene tarifas (Tarifa)
- Un vuelo (Vuelo) se efectúa en un avión (Avion)
- Un vuelo (Vuelo) tiene asientos (Asiento)
- Un vuelo (Vuelo) pertenece a una aerolínea (Aerolinea)
- Un vuelo (Vuelo) tiene horario (Horario)
- Un pasajero (Pasajero) efectúa reservaciones (Reservacion)
- Un pasajero (Pasajero) puede acumular millas como viajero frecuente (ViajeroFrecuente)
- Una reservación (Reservacion) requiere de un registro de tarjeta de crédito (RegistroTarjeta)
- Un registro de tarjeta (RegistroTarjeta) pertenece a un registro de usuario (RegistroUsuario)
- Un vuelo (Vuelo) tiene conexión con otro vuelo (Vuelo)

Con el conocimiento de las relaciones UML entre clase procedemos a crear el diagrama de clases solo con las relaciones, ello se muestra en la figura 2.21



**Figura 2.21** Diagrama de clases con relaciones de la lógica de negocio del sistema.

#### 2.2.1.5 Diagrama de clases con roles

Un rol es el papel que desempeña una clase en la relación que tiene con otra, así que para describir el rol entre dos clases hay que describir de nuevo con oraciones y encontrar el nombre representativo del mismo (esta entre paréntesis en la oración, sin acentos ni tildes).

Un vuelo contiene reservaciones (reservacion)

Un vuelo llega a un aeropuerto de destino (destino)

Un vuelo tiene un aeropuerto de origen (origen)

Un vuelo puede hacer escalas en otros aeropuertos (escala)

Un vuelo tiene tarifas de ida y de vuelta (ida, vuelta)

Un vuelo se efectúa en un avión (avion)

Un vuelo tiene asientos (asiento)

Un vuelo pertenece a una aerolínea (aerolinea)

Un vuelo tiene horario de llegada (llegada)

Un vuelo tiene horario de salida (salida)

Un pasajero puede acumular millas como viajero frecuente (viajero)

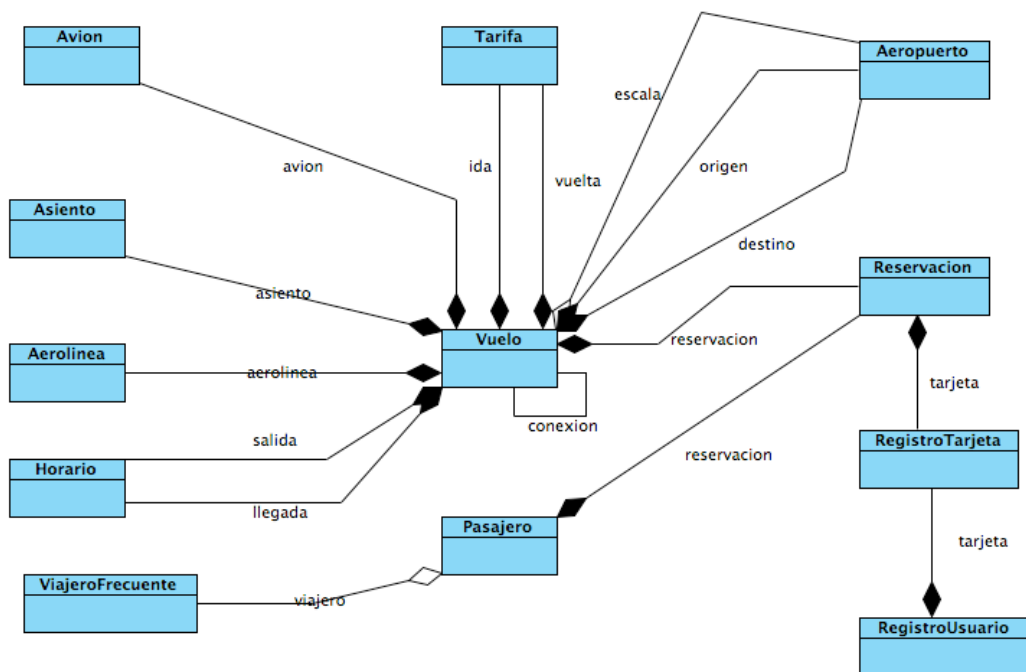
Un pasajero efectúa reservaciones (reservacion)

Una reservación requiere de un registro de tarjeta de crédito (tarjeta)

Un registro de tarjeta de pertenece a un registro de usuario (tarjeta)

Un vuelo puede tener múltiples conexiones (conexion)

El diagrama de clases con roles se muestra en la figura 2.22



**Figura 2.22** Diagrama de clases de relaciones y roles de la lógica de negocio.

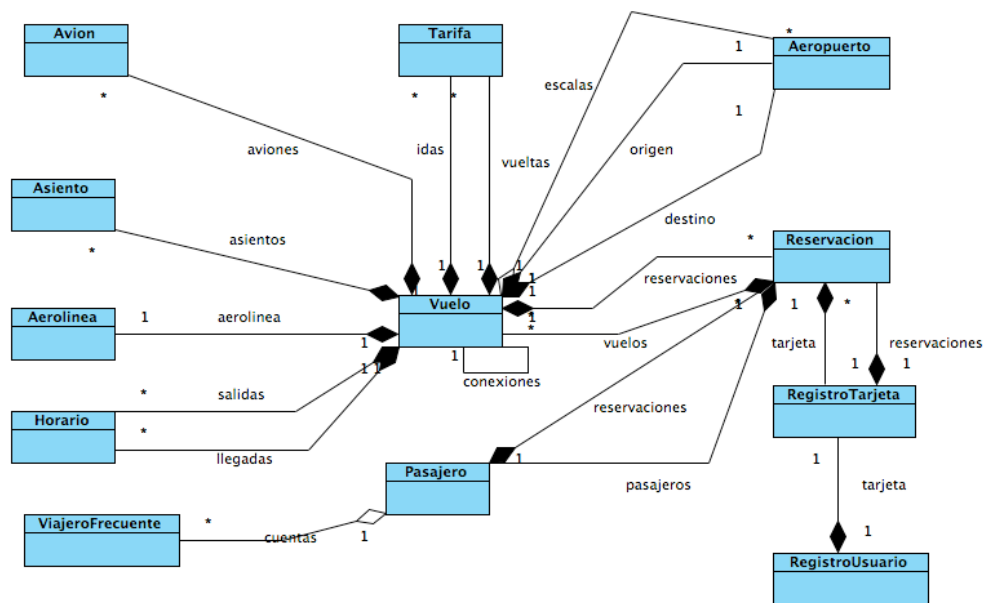
#### 2.2.1.6 Encontrar la multiplicidad (cardinalidad) entre clases.

De igual forma hay que describir con una oración cada relación entre clases describiendo con numerales (están entre paréntesis y los numerales subrayados).

- Un vuelo contiene **múltiples** reservaciones (1.....\*)
- Una reservación se relaciona con **múltiples** vuelos (1.....\*)
- Un vuelo tiene **un** aeropuerto de destino (1.....1)
- Un vuelo tiene **un** aeropuerto de origen (1.....1)
- Un vuelo puede hacer escalas en **múltiples** aeropuertos (1.....\*)
- Un vuelo tiene **múltiples** tarifas de ida y de vuelta (1.....\*)
- Un vuelo se efectúa en **múltiples** aviones (dependiendo del día) (1....\*)
- Un vuelo tiene **múltiples** asientos (1....\*)
- Un vuelo pertenece a **una** aerolínea (1....1)
- Un vuelo tiene **múltiples** horarios de llegada (correspondiendo a diferentes destinos) (1....\*)
- Un vuelo tiene **múltiples** horarios de salida (correspondiendo a diferentes escalas) (1....\*)
- Un pasajero puede acumular millas en **múltiples** cuentas de viajero frecuente (1....\*) (cuentas=nuevo rol)
- Un pasajero efectúa **múltiples** reservaciones (1....\*)
- Múltiples** pasajeros pueden pertenecer a **una** misma reservación (\*.....1)
- Múltiples** reservaciones pueden requerir de **un** mismo registro de tarjeta de crédito (\*.....1)
- Un registro de tarjeta de pertenecen a **un** registro de usuario (1....1)
- Un vuelo puede tener **múltiples** conexiones, (conexiones nuevo rol) (1....\*)

El diagrama completo de clases con las relaciones de asociación, agregación, composición, roles y multiplicidad es el que se muestra en la figura 2.23





**Figura 2.23** Diagrama de clases con relaciones, roles y multiplicidad de la lógica de negocio