

Parte III Desarrollo de Software Orientado a Objetos

En esta tercera parte del libro describiremos las actividades más importantes relacionadas con el desarrollo de software: *Requisitos* (Capítulo 6), *Análisis* (Capítulo 7), *Diseño* (Capítulo 8), *Implementación* (Capítulo 9) y *Pruebas* (Capítulo 10).

6 Modelo de Requisitos

El modelo de requisitos tiene como objetivo delimitar el sistema y capturar la funcionalidad que debe ofrecer desde la perspectiva del usuario. Este modelo puede funcionar como un contrato entre el desarrollador y el cliente o usuario del sistema, y por lo tanto proyecta lo que el cliente desea según la percepción del desarrollador. Por lo tanto, es esencial que los clientes puedan comprender este modelo.

El modelo de requisitos es el primer modelo a desarrollarse, sirviendo de base para la formación de todos los demás modelos en el desarrollo de software. En general, el cualquier cambio en la funcionalidad del sistema es más fácil de hacer, y con menores consecuencias, a este nivel que posteriormente. El modelo de requisitos que desarrollaremos se basa en la metodología *Objectory* (Jacobson et al. 1992), basada principalmente en el modelo de *casos de uso*. Actualmente esta metodología es parte del *Proceso Unificado de Rational* (RUP). El modelo de casos de uso y el propio modelo de requisitos son la base para los demás modelos, como se describió anteriormente en el Capítulo 3 y se resume aquí:

- ?? **Requisitos:** El modelo de casos de uso sirve para expresar el modelo de requisitos, el cual se desarrolla en cooperación con otros modelos como se verá más adelante.
- ?? **Análisis:** La funcionalidad especificada por el modelo de casos de uso se estructura en el modelo de análisis, que es estable con respecto a cambios, siendo un modelo lógico independiente del ambiente de implementación.
- ?? **Diseño:** La funcionalidad de los casos de uso ya estructurada por el análisis es realizada por el modelo de diseño, adaptándose al ambiente de implementación real y refinándose aún más.
- ?? **Implementación:** Los casos de uso son implementados mediante el código fuente en el modelo de implementación.
- ?? **Pruebas:** Los casos de uso son probados a través de las pruebas de componentes y pruebas de integración.
- ?? **Documentación:** El modelo de casos de uso debe ser documentado a lo largo de las diversas actividades, dando lugar a distintos documentos como los manuales de usuario, manuales de administración, etc.

El diagrama de la Figura 6.1 ilustra los distintos modelos. Describiremos los detalles y la notación más adelante.

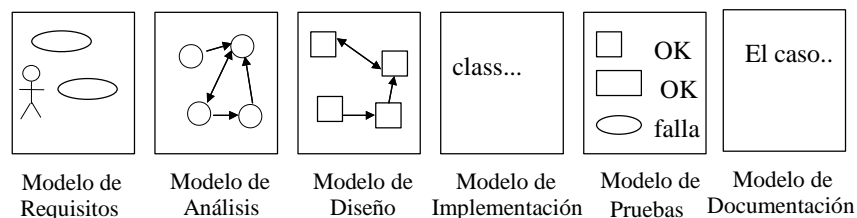


Figura 6.1 Dependencia de los distintos modelos del proceso de software del modelo de casos de uso.

El propósito del modelo de requisitos es comprender completamente el problema y sus implicaciones. Todos los modelos no solamente se verifican contra el modelo de requisitos, sino que también se desarrollan directamente de él. El modelo de requisitos sirve también como base para el desarrollo de las instrucciones operacionales y los manuales ya que todo lo que el sistema deba hacer se describe aquí desde la perspectiva del usuario. El modelo de requisitos no es un proceso mecánico, el analista debe interactuar constantemente con el cliente para completar la información faltante, y así clarificar ambigüedades e inconsistencias. El analista debe separar entre los requisitos verdaderos y las decisiones relacionadas con el diseño e implementación. Se debe indicar cuales aspectos son obligatorios y cuales son opcionales para evitar restringir la flexibilidad de la implementación. Durante el diseño se debe extender el modelo de requisitos con especificaciones de rendimiento y protocolos de interacción con sistemas externos, al igual que provisiones sobre modularidad y futuras extensiones. En ciertas ocasiones ya se puede incluir aspectos de diseño, como el uso de lenguajes de programación particulares.

En la metodología de *Objectory*, el modelo de requisitos consiste de tres modelos principales, visualmente representado por un diagrama de tres dimensiones como se muestra en la Figura 6.2.

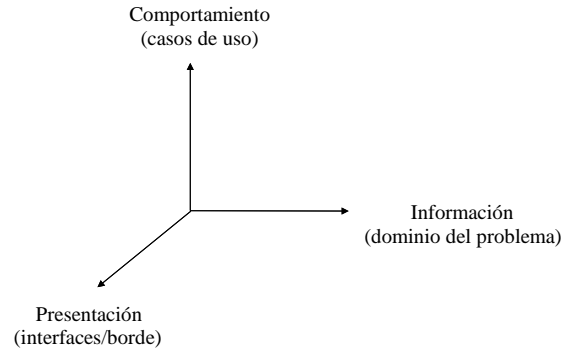


Figura 6.2 Los tres ejes de modelado del modelo de requisitos.

- ?? El modelo de *comportamiento*, basado directamente en el modelo de *casos de uso*, especifica la funcionalidad que ofrece el sistema desde el punto de vista del usuario. Este modelo utiliza dos conceptos claves: *actores* para representar los distintos papeles que los usuarios pueden jugar con el sistema, y *casos de uso* para representar qué pueden hacer los actores con respecto al sistema
- ?? El modelo de *presentación* o modelo de *interfaces* o *borde* especifica cómo interactúa el sistema con actores externos al ejecutar los casos de uso, en particular, en los sistemas de información ricos en interacción con el usuario, especifica cómo se verán visualmente las interfaces gráficas y qué funcionalidad ofrecerá cada una de ellas.
- ?? El modelo de *información* o modelo del *dominio del problema* especifica los aspectos estructurales del sistema. Este modelo conceptualiza el sistema según los objetos que representan las entidades básicas de la aplicación. Aunque en muchas metodologías se permite especificar la funcionalidad completa del sistema utilizando el modelo del dominio del problema, incluyendo operaciones formales sobre los objetos correspondientes a un modelo de requisitos expresado sin casos de uso, el modelo del dominio del problema será de mucha más ayuda como apoyo al modelo de casos de uso y no como una entidad totalmente independiente.

Es importante resaltar que esta separación en tres ejes de modelado independientes es la base para una mayor estabilidad en el desarrollo del sistema, permitiendo minimizar los efectos de cada uno sobre los otros dos. Para ilustrar el modelo de requisitos y el desarrollo de los modelos posteriores, utilizaremos el ejemplo del “Sistema de Reservaciones de Vuelo” como se mencionó anteriormente. Para tal meta, mostraremos inicialmente una *descripción del problema*. A partir de esta descripción inicial se describirán los tres modelos básicos del modelo de requisitos.

6.1 Descripción del Problema

La descripción del problema es una descripción muy preliminar de necesidades que sirve únicamente como punto de inicio para comprender los requisitos del sistema. Se trata aquí de simular una descripción preparada por un cliente la cual debe evolucionar por medio del modelo de requisitos para lograr la especificación final del sistema a desarrollarse. La descripción del problema debe ser una descripción de necesidades y no una propuesta para una solución. La descripción inicial puede ser incompleta e informal. No hay razón para esperar que la descripción inicial del problema, preparada sin un análisis completo, sea correcta.

Utilizaremos como ejemplo un *Sistema de Reservaciones de Vuelos* con acceso vía Internet. Este es un sistema que permite al usuario hacer consultas y reservas de vuelos, además de poder comprar los boletos aéreos de forma remota, sin la necesidad de recurrir a un agente de viajes humano. Se desea que el sistema de reservaciones sea accesible a través del Internet (World Wide Web). Como base para estos sistemas, existen en la actualidad múltiples bases de datos de reservaciones de vuelos que utilizan las agencias de viajes para dar servicio a los clientes, por ejemplo, *Sabre*, *Apollo*, *Worldspan*, *Amadeus*, *Sahara*, *Panorama*, *Gemini*, *Galileo*, *Axess*, etc. Muchos de estas bases de datos y sistemas correspondientes son la base para los sistemas de reservaciones de vuelos de acceso por Internet, como por ejemplo, *Travelocity*, *Expedia*, *Viajando*, *DeViaje*, etc.

La descripción del problema para nuestro sistema de reservaciones de vuelos es la siguiente:

El Sistema de Reservaciones de Vuelos es un sistema que permite al usuario hacer consultas y reservas de vuelos, además de poder comprar los boletos aéreos de forma remota, sin la necesidad de recurrir a un agente de viajes humano. Se desea que el sistema de reservaciones sea accesible a través del Internet (World Wide Web).

El sistema presenta en su pantalla principal un mensaje de bienvenida describiendo los servicios ofrecidos junto con la opción para registrarse por primera vez, o si ya se está registrado, poder utilizar el sistema de reservaciones de vuelos. Este acceso se da por medio de la inserción de un *login* previamente especificado y un *password* previamente escogido y que debe validarse.

Una vez registrado el usuario, y después de haberse validado el registro y contraseña del usuario, se pueden seleccionar las siguientes actividades:

- Consulta de vuelos
- Reserva de vuelos
- Pago de boletos

La consulta de vuelos se puede hacer de tres maneras diferentes:

- Horarios de Vuelos
- Tarifas de Vuelos
- Estado de Vuelo

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

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

El estado 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 está en hora.

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

La reservación de vuelo permite al cliente hacer una reserva para un vuelo particular, especificando la fecha y 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 reserva de vuelo previa y una tarjeta de crédito válida, adquirir los boletos aéreos.

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

Es necesario estar previamente registrados 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 el usuario validado en el sistema.

Nótese lo informal y limitado de esta descripción, la cual estaremos refinando a lo largo del capítulo. Dado que el modelo de casos de uso es el principal de todo el sistema, pues comenzaremos con él.

6.2 Modelo de Casos de Uso

El *modelo de casos de uso* describe un sistema en término de sus distintas formas de utilización, cada uno de estas formas es conocida como un *caso de uso*. Cada caso de uso o flujo se compone de una secuencia de eventos iniciada por el usuario. Dado que los casos de uso describen el sistema a desarrollarse, cambios en los requisitos significarán cambios en los casos de uso. Por ejemplo, un caso de uso para manejar un automóvil sería la secuencia de eventos desde que el conductor entra en el coche encendiendo el motor hasta llegar a su destino final. Por lo tanto, para comprender los casos de uso de un sistema primero es necesario saber quienes son sus usuarios. Por ejemplo, conducir un automóvil es distinto a arreglarlo, donde los usuarios también son distintos, el dueño del automóvil y el mecánico, respectivamente. Para ello se define el concepto de *actor*, correspondiente al tipo de usuario que está involucrado en la utilización de un sistema, siendo el actor una entidad externa al propio sistema. Juntos, el actor y el caso de uso representan los dos elementos básicos de este modelo lo cual se muestran de manera gráfica en la Figura 6.3 de acuerdo a la notación UML.

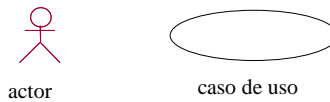


Figura 6.3 El actor y el caso de uso son las entidades básicas del modelo de casos de uso.

Los casos de uso son una idea simple y práctica que no requieren muchas habilidades tecnológicas para ser utilizadas (a diferencia de las demás actividades del desarrollo). Por el contrario, si se volvieran muy complejas se perdería un poco la importancia de los casos de uso. Dado que el modelo de requisitos es la primera actividad del desarrollo del sistema, nos podemos dar el lujo de muchos cambios en su especificación sin afectar al resto del sistema. Más aún, considerando que siempre habrá cambios, no debe hacerse demasiado trabajo muy temprano, ya que solo sería descartado. Cuando se identifican y describe los casos de uso, habrá ciertas imprecisiones que se irán resolviendo más adelante. Para ello, un enfoque incremental es lo indicado. De esta manera se puede desarrollar de forma independiente los distintos casos de uso y luego integrarlos para formar el modelo de requisitos completo. Esta habilidad para tomar parte de la funcionalidad permite un desarrollo más flexible e incluso concurrente.

Actores

Los *actores* son entidades distintas a los usuarios, en el sentido que los usuarios son las personas reales que utilizan el sistema, mientras que los actores representan un cierto papel que una persona real puede jugar. Utilizando terminología orientada a objetos, se considera al actor como una *clase* de usuario, mientras que los usuarios se consideran como *objetos* o instancias de esa clase. Incluso, una misma persona puede aparecer como diferentes instancias de diferentes actores.

Los actores modelan cualquier entidad externa que necesite intercambiar información con el sistema. Los actores no están restringidos a ser personas físicas, pudiendo representar otros sistemas externos al actual. Lo esencial es que los actores representen entidades externas al sistema. Además, cada uno de estos actores podrá ejecutar una o más tareas del sistema.

Antes de identificar los casos de uso se identifican los actores del sistema. La razón para comenzar con la identificación de los actores es para que ellos sean la herramienta principal para luego encontrar los casos de uso. Cada actor ejecuta un número específico de casos de uso en el sistema. Al definir todos los actores y casos de uso en el sistema, se define la funcionalidad completa del sistema.

Encontrar actores puede tomar trabajo y raramente se encuentran todos los actores de una vez. Por ejemplo, un sistema de computación puede tener diferentes tipos de usuarios: programadores, operadores, administradores, o usuarios generales. Cada uno de estos tipos de usuario corresponde a un actor diferente y como mencionamos anteriormente, una misma persona puede jugar, por ejemplo, el papel de programador u operador.

Para especificar los actores de un sistema, se dibuja un diagrama correspondiente a la *delimitación del sistema*, la cual representa al sistema como una “caja negra” y a los diferentes actores como entidades externas a ésta, como se muestra en la Figura 6.4.

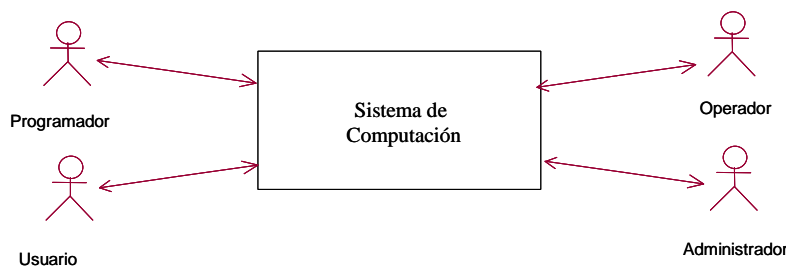


Figura 6.4 Delimitación de un sistema según los actores.

En general, no se describen los actores con demasiado detalle por ser estos externos al sistema además de que sus acciones no son deterministas, en otras palabras, un actor a diferencia del propio sistema, en cada momento puede decidir entre múltiples opciones. Por otro lado, el sistema y los casos de uso correspondientes deben ser deterministas, de lo contrario el sistema hará lo que le plazca, lo cual no es aceptable. Sin embargo, para poder identificar los casos de uso, es necesario primero identificar los actores del sistema, comenzando por aquellos que son la razón principal del sistema, conocidos como *actores primarios*. Estos actores típicamente rigen la secuencia lógica de ejecución del sistema. Además de los actores primarios existen actores supervisando y manteniendo el

sistema. Estos *actores secundarios* existen primordialmente como complemento a los actores primarios, siendo esta distinción importante para dedicarle el esfuerzo principal a las necesidades de los actores primarios. Al contrario de los actores primarios que típicamente corresponden a personas físicas, los actores secundarios corresponden por lo general a máquinas o sistemas externos, siendo estos últimos más difíciles de identificar. Los actores secundarios tienden a responder a secuencias lógicas del sistema y no tanto a inicializarlas de manera propia. En particular, existe siempre la duda, por ejemplo, de si el sistema operativo o una base de datos serían actores. La decisión depende del papel que jueguen con respecto al sistema en desarrollo, si juegan un papel activo entonces deben modelarse como actores.

Volviendo al ejemplo del *Sistema de Reservas de Vuelos*, se pueden identificar de la descripción del problema que se tiene al menos un actor, el *Usuario*, encargado de hacer las consultas y reservas con el sistema. Si se analiza un poco más, se puede identificar que las bases de datos de los sistemas externos de reservas juegan un papel muy activo con respecto al sistema en desarrollo. A este actor lo llamaremos la *Base de Datos de Reservas*, el cual mantiene la información sobre los vuelos y reservas. Más aún, podemos identificar un actor adicional representando una segunda base de datos, ésta involucrada en la información de los usuarios más que de las reservas. A este actor lo llamaremos la *Base de Datos de Registros*, encargado de mantener la información de los usuarios sobre la utilización del sistema. El diagrama de *Delimitación del Sistema* con los actores correspondientes se muestra en la Figura 6.5.

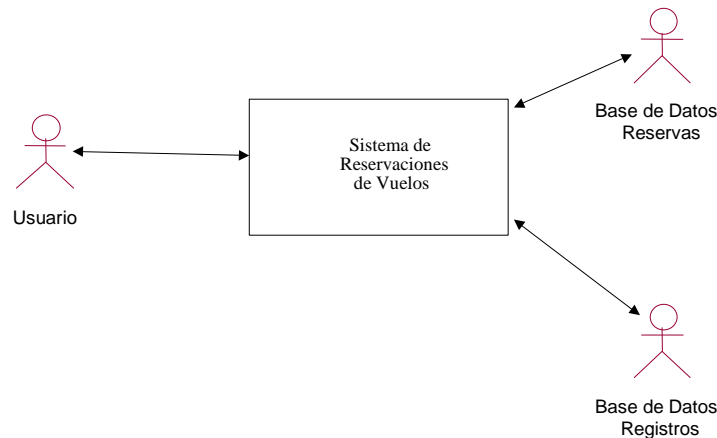


Figura 6.5 Delimitación del sistema de reservas de vuelo.

Volviendo a la distinción entre actor y persona, una misma persona puede jugar el papel del actor *Usuario* cuando hace reservas y además puede trabajar para el sistema de reservas, por ejemplo como *Operador*, correspondiente a otro actor no mostrado en nuestro ejemplo.

El actor *Usuario* se considera un actor primario, ya que el sistema se construye pensando en sus usuarios, mientras que *Base de Datos de Reservas* y *Base de Datos de Registro* son ambos actores secundarios, ya que si no existieran usuarios no habría necesidad del sistema.

Cuando diferentes actores juegan roles similares ellos pueden heredar de un *actor abstracto* común, como se muestra mediante el actor abstracto *Base de Datos* en el ejemplo de la Figura 6.6. El resto de los actores se conoce como actores concretos, utilizando terminología similar a aquella de herencia, como se vio en el Capítulo 4.

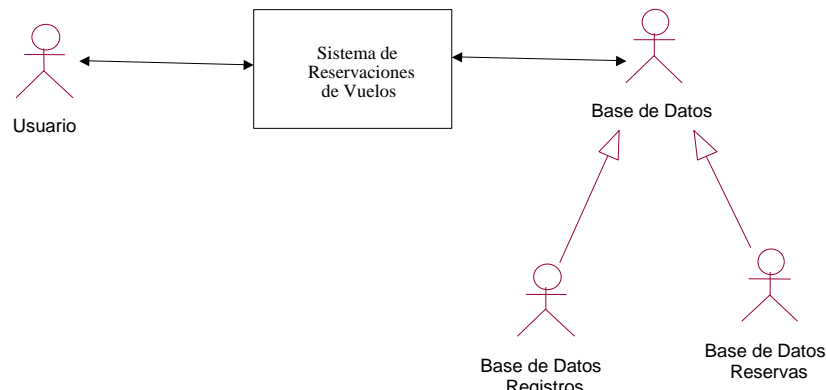


Figura 6.6 Delimitación del sistema de reservas de vuelo con ejemplo de herencia entre actores.

La ventaja de modelar actores abstractos es que expresa similitudes entre caso de uso. Si el mismo o parte del mismo caso de uso se puede ejecutar por varios actores diferentes, el caso de uso necesita ser especificado solo con respecto a un actor en lugar de varios. Por otro lado, los actores abstractos también pueden ser utilizados para especificar privilegios comunes a múltiples actores en un sistema.

Casos de Uso

Después de haber definido los actores del sistema, se define la funcionalidad propia del sistema por medio de los *casos de uso*. Utilizando terminología orientada a objetos, cada caso de uso define una clase o forma particular de usar el sistema mientras que cada ejecución del caso de uso se puede ver como una instancia del caso de uso, o sea, un objeto, con estado y comportamiento. Cada caso de uso constituye un flujo completo de eventos especificando la interacción que toma lugar entre el actor y el sistema. El actor primario es encargado de dar inicio a esta interacción, mientras que los casos de uso son instanciados como respuesta al evento anterior. Una instancia de un actor puede ejecutar varias de estas secuencias, consistiendo de diferentes acciones que a su vez deben llevarse a cabo. La instancia del caso de uso existe mientras el caso de uso siga ejecutando. La ejecución del caso de uso termina cuando el actor genere un evento que requiera un caso de uso nuevo. Las diferentes instancias de los casos de uso se conocen como *escenarios*. Como varios casos de uso pueden comenzar de una misma forma, no es siempre posible decidir que caso de uso se ha instanciado hasta que éste se haya completado.

La descripción de los casos de uso es mediante diagramas similares a los de transición de estados. Se puede ver a cada caso de uso como representando un estado en el sistema, donde un estímulo enviado entre un actor y el sistema ocasiona una transición entre estados.

En la Figura 6.7 se muestra un ejemplo de casos de uso, donde un programador escribe y depura un programa, mientras que otro usuario lo ejecuta.

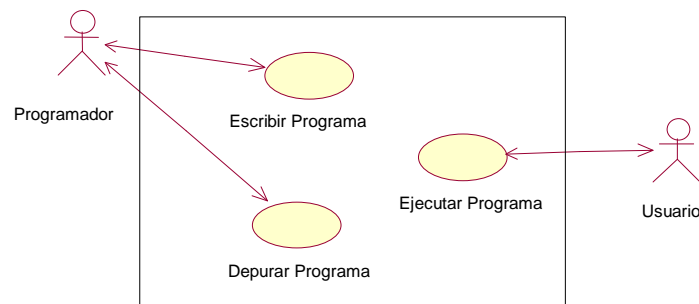


Figura 6.7 Ejemplo de casos de uso mostrando la relación con los actores.

Para identificar los casos de uso, se puede leer la descripción del problema y discutirlo con aquellos que actuarán como actores, haciendo preguntas como:

- ?? ¿Cuales son las tareas principales de cada actor?
- ?? ¿Tendrá el actor que consultar y modificar información del sistema?
- ?? ¿Tendrá el actor que informar al sistema sobre cambios externos?

?? ¿Desea el actor ser informado sobre cambios inesperados?

Por ejemplo, en un sistema de conmutación telefónica, un actor podría ser un suscriptor, y un caso de uso típico sería hacer una llamada local. El caso de uso comienza cuando el suscriptor levanta el teléfono. Otro caso de uso es ordenar una llamada de despertar. Ambos casos de uso comienzan cuando el suscriptor levanta el teléfono. Pero cuando el suscriptor levanta el teléfono no sabemos cual caso de uso desea ejecutar. Por lo tanto, los casos de uso pueden comenzar de forma similar y no podemos saber cual se está instanciando hasta que se termine. En otras palabras, el actor no requiere que un caso de uso se ejecute, sólo que inicie una secuencia de eventos que finalmente resulte en la terminación de algún caso de uso.

En el *sistema de reservaciones de vuelos* utilizamos los actores ya identificados como punto de partida. Dado que el *Usuario* es el actor primario se comienza con él. El sistema tiene que poder dar ciertos servicios al usuario, como consultas y reservas. De aquí podemos definir nuestros casos de uso principales, *Consultar Información* y *Hacer Reservación*. Nótese que los nombres de los casos de uso deben corresponder a acciones y no tanto a sustantivos. La Figura 6.8 muestra el sistema con los dos casos de uso, además de un caso de uso adicional, *Mantener el Sistema*, correspondiente a un actor secundario *Operador*. Sin embargo, para lograr una mejor especificación de requisitos y evitar complejidad adicional, no veremos ninguna funcionalidad de mantenimiento en nuestro desarrollo, un ejemplo adicional de como podemos concentrarnos en ciertos aspectos de los requisitos durante diversas etapas.

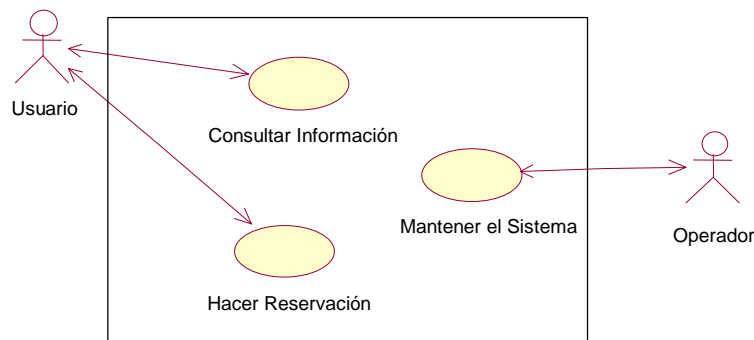


Figura 6.8 Ejemplo de casos de uso para un sistema de reservaciones de vuelo.

En la descripción del problema se menciona que para poder utilizar el sistema el usuario debe estar registrado, por lo cual agregamos un caso de uso *Registrar Usuario*. Por otro lado, se debe incluir la *Base de Datos de Reservas*, y la *Base de Datos de Registro* ya que son actores secundarios necesarios. Estos tres casos de uso se muestran en la Figura 6.9.

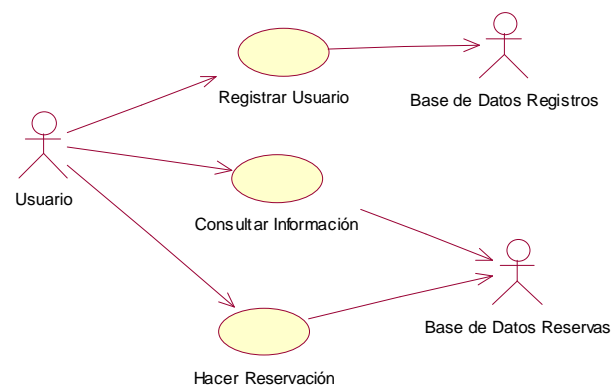


Figura 6.9 Casos de uso principales para el sistema de reservaciones de vuelo. Se eliminó en el diagrama la delimitación del sistema.

Aunque la idea del modelo de casos de uso es mantener lo más sencillo posible la complejidad en los diagramas dada la etapa en que nos encontramos del desarrollo, existen ciertas extensiones al concepto básico de caso de uso que permiten subdividirlos de manera limitada. Para ello se permite la asignación de secciones del flujo completo de

un caso de uso en casos de uso separados. Las razones para esto son aprovechar distintas variantes en los casos de uso que se repiten en la lógica del sistema de manera análoga al concepto de herencia. Lamentablemente, no es siempre obvio la funcionalidad que debe ser puesta en casos de uso separados, y qué situación es sólo una pequeña variante de un mismo caso de uso que no amerita tal división. La complejidad de los casos de uso es un factor importante para tal decisión. Existen dos enfoques distintos para expresar variantes:

1. Si las diferencias entre los casos de uso son pequeñas, estos se pueden describir como variantes de un mismo caso de uso, y se pueden definir *subflujos* separados dentro de un mismo caso de uso, dando lugar al concepto de flujos *básicos* y subflujos *alternos*. Por ejemplo, en el caso de uso registrar un usuario, crear por primera vez el registro y actualizarlo se pueden considerar como dos secuencias de eventos lógicamente similares por lo cual se tratan como subflujos alternos. En general, primero se describe el *flujo básico*, el cual es el flujo más importante dentro del caso de uso. Este flujo corresponde a la secuencia de eventos más común o típica de casos de uso. Posteriormente se describen las variantes o *flujos alternos* que incluyen flujos para el manejo de variantes en la lógica. Normalmente un caso de uso tiene sólo un curso básico, pero varios cursos alternos.
2. Si las diferencias entre los casos de uso son grandes, se deben describir como casos de uso separados. Cuando los casos de uso se dividen en casos de uso separados se utiliza principalmente las relaciones de *extensión*, *inclusión* y generalización como se describe a continuación.

La identificación de casos de uso y de los aspectos anteriores, es a menudo un proceso muy iterativo donde varios intentos se hacen hasta llegar a una solución final estable. Cuando esto ocurre, cada caso de uso debe ser descrito con más detalle.

Extensión

Un concepto importante que se utiliza para estructurar y relacionar casos de uso es la *extensión*. La extensión especifica cómo un caso de uso puede *insertarse* en otro para extender la funcionalidad del anterior. El caso de uso donde se va a insertar la nueva funcionalidad debe ser un flujo completo, por lo cual éste es independiente del caso de uso a ser insertado. De esta manera, el caso de uso inicial no requiere consideraciones adicionales al caso de uso a ser insertado, únicamente especificando su punto de inserción.

Tomando como ejemplo el *Sistema de Reservas de Vuelos*, se muestra en la Figura 6.10 la notación para extensión, utilizando la etiqueta “extiende” (“extend”), donde el caso de uso de *Hacer Reservación* es extendido mediante el caso de uso *Pagar Reservación*.

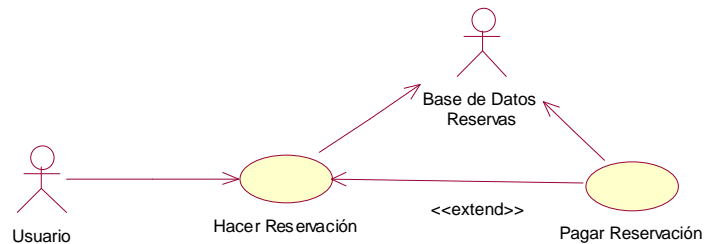


Figura 6.10 Casos de uso *Hacer Reservación* con extensión de *Pagar Reservación*.

En general la extensión se utiliza para modelar secuencias de eventos opcionales de casos de uso que al manejarse de manera independiente pueden ser agregados o eliminados del sistema de manera modular. Se puede considerar a la asociación de extensión como una interrupción en el caso de uso original que ocurre donde el nuevo caso de uso se va a insertar. Para cada caso de uso que vaya a insertarse en otro caso de uso, se especifica la posición en el caso de uso original donde el caso de uso de extensión debe insertarse. El caso de uso original se ejecuta de forma normal hasta el punto donde el caso de uso nuevo se inserta. En este punto, se continúa con la ejecución del nuevo curso. Después que la extensión se ha terminado, el curso original continúa como si nada hubiera ocurrido. Por regla general, se describe primero los casos de uso básicos totalmente independientes de cualquier extensión de funcionalidad y luego aquellos de extensión.

Inclusión

Una relación adicional entre casos de uso es la *inclusión*. A diferencia de una extensión, la inclusión se define como una sección de un caso de uso que es parte obligatoria del caso de uso básico. El caso de uso donde se va a insertar

la funcionalidad depende del caso de uso a ser insertado. Se etiqueta la relación con “include” (“include”). Por ejemplo, en el *Sistema de Reservas de Vuelos*, el caso de uso de *Consultar Información* incluye el caso de uso *Validar Usuario* como se muestra en la Figura 6.11.

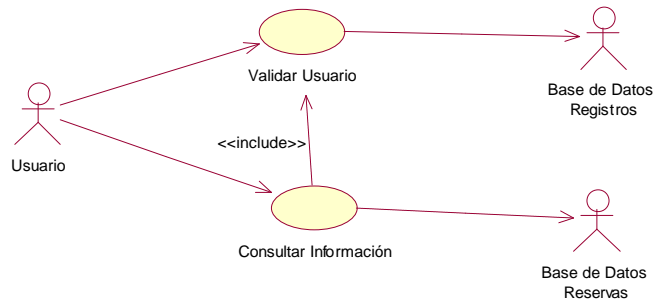


Figura 6.11 Casos de uso *Consultar Información* con inclusión de *Validar Usuario*.

Generalización

Una relación adicional entre casos de uso es la generalización la cual apoya la reutilización de los casos de uso. Mediante la relación de generalización es necesario describir las partes similares una sola vez en lugar de repetirlas para todos los casos de uso con comportamiento común. Se conoce a los casos de uso que se extraen como casos de uso *abstractos*, ya que no serán instanciados independientemente, sirviendo sólo para describir partes que son comunes a otros casos de uso. Se conoce a los casos de uso que realmente serán instanciados como casos de uso *concretos*. Las descripciones de los casos de uso abstractos se incluyen en las descripciones de los casos de uso concretos. Esto significa que, cuando una instancia de un caso de uso sigue la descripción de un caso de uso concreto, en cierto punto continúa la descripción del caso de uso abstracto en su lugar. Los casos de uso abstractos también pueden ser usados por otros casos de uso abstractos. Es difícil saber cuando ya no tiene sentido seguir extrayendo más casos de uso abstractos. Una técnica para extraer casos de uso abstractos es identificar actores abstractos. Normalmente, no hay razón para crear casos de uso abstractos que se usan en un solo caso de uso. Por lo general, comportamientos similares entre casos de uso se identifica después que se han descrito los casos de uso. Sin embargo, en algunos casos es posible identificarlos antes. De forma intuitiva, esto es un ‘herencia de secuencias’ (en lugar de operaciones en el caso normal de herencia). Por ejemplo, en el *Sistema de Reservas de Vuelos*, el caso de uso de *Consultar Información* incluye el caso de uso *Validar Usuario* como se muestra en la Figura 6.12.

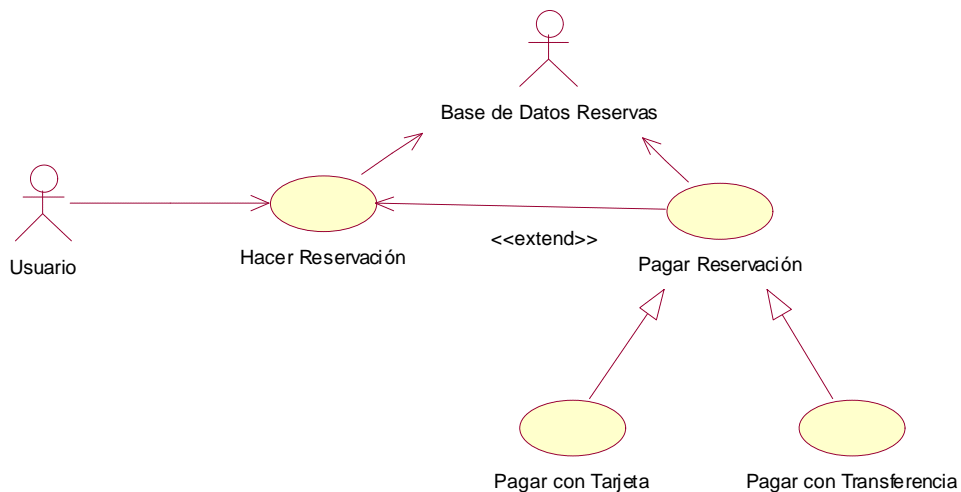


Figura 6.12 Casos de uso *Pagar Reservación* con generalización de pagos: *Pagar con Tarjeta* y *Pagar con Transferencia*.

Las relaciones de *extensión e inclusión* entre casos de uso son ambas asociaciones de clases, a diferencia de la *generalización*. En la mayoría de los casos, la selección es bastante obvia, sin embargo el criterio importante es ver qué tanto se acoplan las funcionalidades de los casos de uso. Si el caso de uso a ser extendido es útil por sí mismo, la relación debe ser descrita utilizando extensión. Si los casos de uso son fuertemente acoplados, y la inserción debe tomar lugar para obtener un curso completo, la relación debe ser descrita utilizando inclusión. Por otro lado, la generalización es utilizada cuando dos o más casos de uso comparte funcionalidad común la cual es extendida por cada uno al estilo de la generalización entre clases. También hay una diferencia en cómo se encuentran. Las extensiones se encuentran en un caso de uso existente que el usuario desea ramificar en secuencias adicionales, mientras que las inclusiones se encuentran de la extracción de secuencias comunes entre varios casos de uso diferentes. Las generalizaciones se encuentran al tratar de especializar de diferente manera un mismo caso de uso. Finalmente, se desean diagramas con baja complejidad que no sean "telarañas" pero que muestren de manera esquemática las posibles secuencias de interacciones entre los actores y el sistema.

Mostramos en la Figura 6.13 el diagrama completo de casos de uso para el *Sistema de Reservaciones de Vuelos*. Los casos de uso adicionales en este diagrama son la extensión de *Registrar Tarjeta* y *Pagar Reservación*. Este último caso de uso es interesante por que extiende *Hacer Reservación* e incluye *Registrar Tarjeta*, ambos requisitos para poder comprar un boleto con el sistema. Además de la inclusión anterior, también se incluyen los casos de uso de *Validar Usuario* y *Ofrecer Servicios* en los casos de uso básicos: *Registrar Usuario*, *Consultar Información* y *Hacer Reservación*.

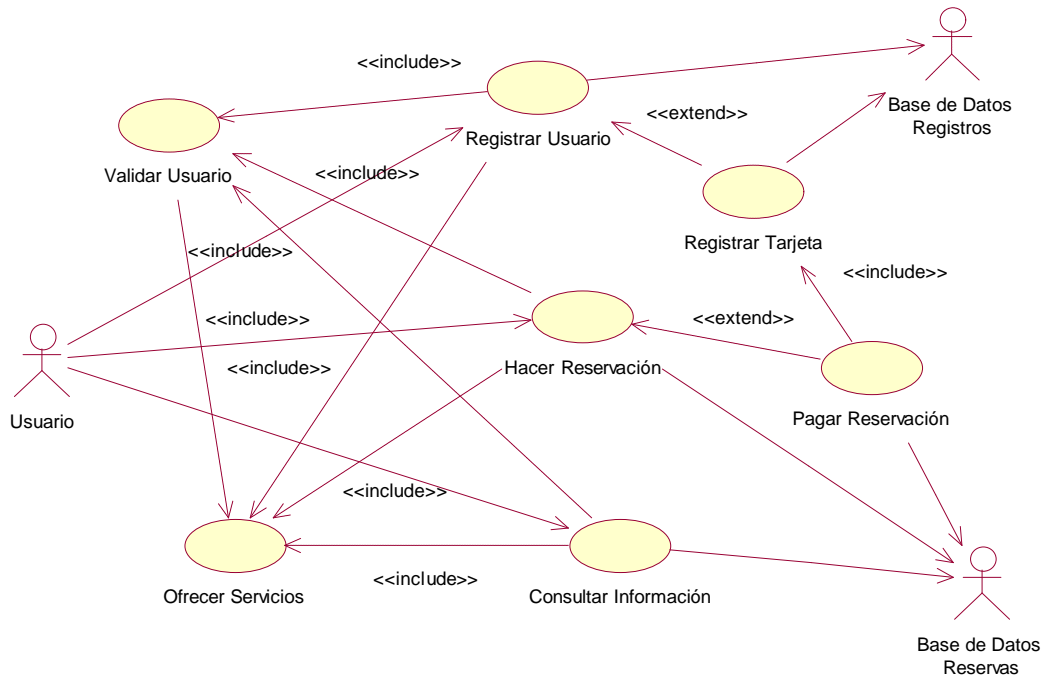


Figura 6.13 Casos de uso completos para el sistema de reservaciones de vuelo.

Documentación

Parte fundamental del modelo de casos de uso es una descripción textual detallada de cada uno de los actores y casos de uso identificados. Estos documento son sumamente críticos ya que a partir de ellos se desarrollará el sistema completo. El formato de documentación es el siguiente:

| | |
|----------------------|--|
| Actor: | Nombre del actor |
| Casos de Uso: | Nombre de los casos de uso en los cuales participa |
| Tipo: | <i>Primario</i> o <i>Secundario</i> |
| Descripción: | Breve descripción del actor |

Las descripciones de los casos de uso representan todas las posibles interacciones de los actores con el sistema para los eventos enviados o recibidos por los actores. En esta etapa no se incluyen eventos internos al propio sistema ya que esto será tratado durante el análisis y únicamente agregaría complejidad innecesaria en esta etapa. El formato de documentación es el siguiente:

| | |
|-------------------------|---|
| Caso de Uso: | Nombre del caso de uso |
| Actores: | Actores primarios y secundarios que interactúan con el caso de uso. |
| Tipo: | Tipo de flujo: <i>Básico, Inclusión, Extensión, Generalización</i> , o algún otro. |
| Propósito: | Razón de ser del caso de uso. |
| Resumen: | Resumen del caso de uso. |
| Precondiciones: | Condiciones que deben satisfacerse para poder ejecutar el caso de uso. |
| Flujo Principal: | El flujo de eventos más importante del caso de uso, donde dependiendo de las acciones de los actores se continuará con alguno de los subflujos. |
| Subflujos: | Los flujos secundarios del caso de uso, numerados como (S-1), (S-2), etc. |
| Excepciones: | Excepciones que pueden ocurrir durante el caso de uso, numerados como (E-1), (E-2), etc. |

Dado que el modelo de casos de uso está motivado y enfocado principalmente hacia los sistemas de información donde los usuarios juegan un papel primordial, es importante ya relacionarse con las interfaces a ser diseñadas en el sistema. Estas interfaces sirven para apoyar de mejor manera la descripción de los casos de uso además de servir de base para prototipos iniciales.

Un comentario importante sobre la especificación de estos documentos es que se sigue un proceso iterativo para definir cada uno de ellos, pudiéndose modificar o refinar posteriormente. Obviamente, cuanto más tarde ocurran estos cambios más costoso será implementarlos. Nótese que en las siguientes descripciones ya se hace referencia a pantallas de interacción con el usuario, las cuales serán mostradas en un diseño preliminar en la siguiente sección. Los actores y casos de uso del sistema de reservaciones de vuelo son descritos en la sección 6.4.

6.3 Modelo de Interfaces

El *modelo de interfaces* describe la presentación de información entre los actores y el sistema. Se especifica en detalle como se verán las interfaces de usuario al ejecutar cada uno de los casos de uso. Si se trata de *Interfaz Humano Computadora* (“HCI - Human Computer Interface”) se puede usar esquemas de cómo vería el usuario las pantallas cuando se ejecuta cada caso de uso. También se puede generar una simulación más sofisticada usando un *Sistema Manejador de Interfaces de Usuario* (“UIMS - User Interface Management System”). Normalmente, un prototipo funcional de requisitos mostrando las interfaces de usuario es una estrategia importante. Esto ayuda al usuario a visualizar los casos de uso según serán mostrados por el sistema a ser construido. Tal enfoque elimina muchas posibilidades de malos entendimientos. Cuando se diseñan las interfaces de usuario, es esencial tener a los usuarios involucrados, siendo esencial que las interfaces reflejen la visión lógica del sistema. Esto es realmente uno de los principios fundamentales del diseño de interfaces humanas, donde debe existir consistencia entre la imagen conceptual del usuario y el comportamiento real del sistema. Si las interfaces son protocolos de hardware, se puede referir a los diferentes estándares, como protocolos de comunicación. Estas descripciones de interfaces son por lo tanto partes esenciales de las descripciones de los casos de uso y las deben acompañar. En estas etapas iniciales del desarrollo el diseño de las pantallas no es tan importante como el manejo de información que se ofrece el cual debe corresponder a las necesidades de cada caso de uso, algo que se mostrará a continuación para el *Sistema de Reservaciones de Vuelos*.

6.4 Actores y Casos de Uso para el Sistema de Reservaciones de Vuelos

Tomando como ejemplo el sistema de reservaciones de vuelo mostraremos la documentación de los actores y casos de uso junto con el diseño de las interfaces que serán usadas como prototipo del sistema. Estos diseños pueden hacerse en papel o aprovechar una herramienta que simplifique la tarea del diseño de pantallas. El objetivo primordial es la lógica de “navegación” la cual debe basarse en el modelo de casos de uso más que la sofisticación del diseño gráfico.

Actores

Se describen en total tres actores en el sistema de reservaciones de vuelos. El *Usuario* interactúa con todos los casos de uso, aunque todas las asociaciones no fueron diagramadas de manera explícita en la Figura 6.13.

| | |
|----------------------|--|
| Actor: | <i>Usuario</i> |
| Casos de Uso: | <i>Validar Usuario, Registrar Usuario, Registrar Tarjeta, Consultar Información, Hacer Reservación, Pagar Reservación, Ofrecer Servicios</i> |
| Tipo: | <i>Primario</i> |
| Descripción: | Es el actor principal y representa a cualquier persona que desee utilizar del sistema de reservaciones. |

La *Base de Datos de Registro* interactúa con los casos de uso relacionados exclusivamente con registro.

| | |
|----------------------|--|
| Actor: | <i>Base de Datos de Registro</i> |
| Casos de Uso: | <i>Validar Usuario, Registrar Usuario, Registrar Tarjeta</i> |
| Tipo: | <i>Secundario</i> |
| Descripción: | Es un actor secundario y representa a la base de datos donde se guarda toda la información relacionada con los usuarios pero independiente de las reservaciones. |

La *Base de Datos de Reservaciones* interactúa con los casos de uso relacionados exclusivamente con reservaciones.

| | |
|----------------------|--|
| Actor: | <i>Base de Datos de Reservaciones</i> |
| Casos de Uso: | <i>Consultar Información, Hacer Reservación, Pagar Reservación</i> |
| Tipo: | <i>Secundario</i> |
| Descripción: | Es un actor secundario y representa a la base de datos donde se guarda toda la información relacionada con las reservaciones pero independiente de los propios usuarios del sistema. |

Casos de Uso

Como apoyo en la descripción de los casos de uso mostraremos las diversas pantallas diseñadas, comenzando con la pantalla inicial. Dado que el requisito de uso del sistema es que todo usuario deba haberse registrado, la pantalla principal debe dar dos opciones: (i) registrarse por primera vez y (ii) validar un registro existente como se muestra en la Figura 6.14.

Figura 6.14. Pantalla Principal del Sistema (P-1).

Nótese que el caso de uso *Registrar Usuario*, como lo describiremos en nuestro ejemplo, agrupa la lógica relacionada con un usuario ya registrado junto con otro que se registra por primera vez. Dado que la opción para registrarse por primera vez debe aparecer en la pantalla inicial (P-1), la opción en la pantalla de servicios (P-2) permite únicamente obtener el registro de un usuario ya registrado. Y aunque se podría haber definido dos casos de uso separados para la lógica de registro, por ejemplo, *Crear Registro Usuario* y *Obtener Registro Usuario*, consideramos que estos dos son más bien subflujos de un mismo caso de uso. Vale la pena un comentario adicional. Existen muchas maneras de describir un sistema similar, donde por lo general una forma no es necesariamente más correcta que la otra, siendo más bien una cuestión de estilo o creencias del analista. En nuestro caso buscamos soluciones compactas reduciendo el número total de casos de uso, siempre y cuando la lógica esté muy relacionada. Se incluyen además varios subflujos para permitir llamar secciones del flujo desde distintos lugares ya que no se puede comenzar un flujo o subflujo en la mitad. A continuación describimos los distintos casos de uso con las pantallas correspondientes. Se excluyen pantallas menores como aquellas con mensajes de error o confirmación del éxito de la operación. Comenzamos describiendo los flujos *Validar Usuario* y *Ofrecer Servicios* que son incluidos por los diversos casos de uso. Posteriormente describimos cada uno de los casos básicos y de extensión.

Validar Usuario

El caso de uso *Validar Usuario* está vinculado con la pantalla principal (P-1) y es llamada a partir de los casos de uso *Registrar Usuario*, *Consultar Información* y *Hacer Reservación* como se mostró en el diagrama de la Figura 6.13. Dado que este caso de uso es insertado en los anteriormente mencionados, depende de estos insertarlo y llamarlo apropiadamente.

| | |
|--------------------|---|
| Caso de Uso | <i>Validar Usuario</i> |
| Actores | <i>Usuario, Base de Datos Registros</i> |
| Tipo | <i>Inclusión</i> |
| Propósito | Validar a un usuario ya registrado para el uso del sistema de reservaciones de vuelo. |

| | |
|------------------------|---|
| Resumen | Este caso de uso es iniciado por el <i>Usuario</i> . Valida al usuario mediante un <i>login</i> y <i>password</i> a ser validado con su respectivo registro de usuario para así poder utilizar el sistema de reservaciones. |
| Precondiciones | Se requieren haber ejecutado anteriormente el caso de uso <i>Registrar Usuario</i> subflujo <i>Crear Registro Usuario</i> . |
| Flujo Principal | Se presenta al usuario la Pantalla Principal (P-1). El Usuario puede seleccionar entre las siguientes opciones: "Registrarse por Primera Vez", "OK" y "Salir". Si la actividad seleccionada es "Registrarse por Primera Vez", se ejecuta el caso de uso <i>Registrar Usuario</i> , subflujo <i>Crear Registro Usuario</i> (S-1). Si la actividad seleccionada es "OK", se valida el registro de usuario mediante un <i>login</i> y un <i>password</i> insertados por el usuario en la Pantalla Principal (P-1). Una vez validado el usuario (E-1), se continúa con el caso de uso <i>Ofrecer Servicios</i> . Si la actividad seleccionada es "Salir" se saldrá del sistema. |
| Subflujos | Ninguno. |
| Excepciones | <i>E-1 no hubo validación</i> : El <i>login/password</i> no se validó correctamente. Se solicita al usuario volver a registrarse. Después tres intentos se saldrá del sistema. |

Ofrecer Servicios

Si nos referimos al diagrama de casos de uso de la Figura 6.13 vemos que existen tres casos de uso básicos que el usuario puede instanciar, *Registrar Usuario*, *Hacer Reservación* y *Consultar Información*. El caso de uso *Ofrecer Servicio* es incluido por estos tres casos de uso básicos para delegar de manera adecuada a ellos según las opciones seleccionadas por el usuario. También es incluido por el caso de uso *Validar Usuario* luego de la validación de un usuario.

| | |
|------------------------|---|
| Caso de Uso | <i>Ofrecer Servicios</i> |
| Actores | <i>Usuario</i> |
| Tipo | <i>Inclusión</i> |
| Propósito | Ofrecer los diversos servicios a un usuario ya registrado para el uso del sistema de reservaciones de vuelo. |
| Resumen | Este caso de uso es iniciado por el <i>Usuario</i> . Tiene opciones para utilizar las diversas opciones del sistema de reservaciones. |
| Precondiciones | Se requieren haber la validación correcta del usuario. |
| Flujo Principal | Se presenta al usuario la Pantalla Servicios (P-2). El usuario puede seleccionar entre las siguientes actividades: "Consultar Información", "Hacer Reservación", "Obtener Registro" y "Salir". Si la actividad seleccionada es "Consultar Información", se continua con el caso de uso <i>Consultar Información</i> , subflujo <i>Consultar</i> (S-1). Si la actividad seleccionada es "Hacer Reservación", se continua con el caso de uso <i>Hacer Reservación</i> , subflujo <i>Solicitar Clave Reservación</i> (S-1). Si la actividad seleccionada es "Obtener Registro", se continúa con el caso de uso <i>Registrar Usuario</i> , subflujo <i>Obtener Registro Usuario</i> (S-2). Si la actividad seleccionada es "Salir" se saldrá del sistema. |
| Subflujos | Ninguno. |
| Excepciones | Ninguno. |

Por tal motivo la siguiente pantalla del sistema debe permitir al usuario seleccionar las opciones correspondientes, como se muestra en la Figura 6.15.

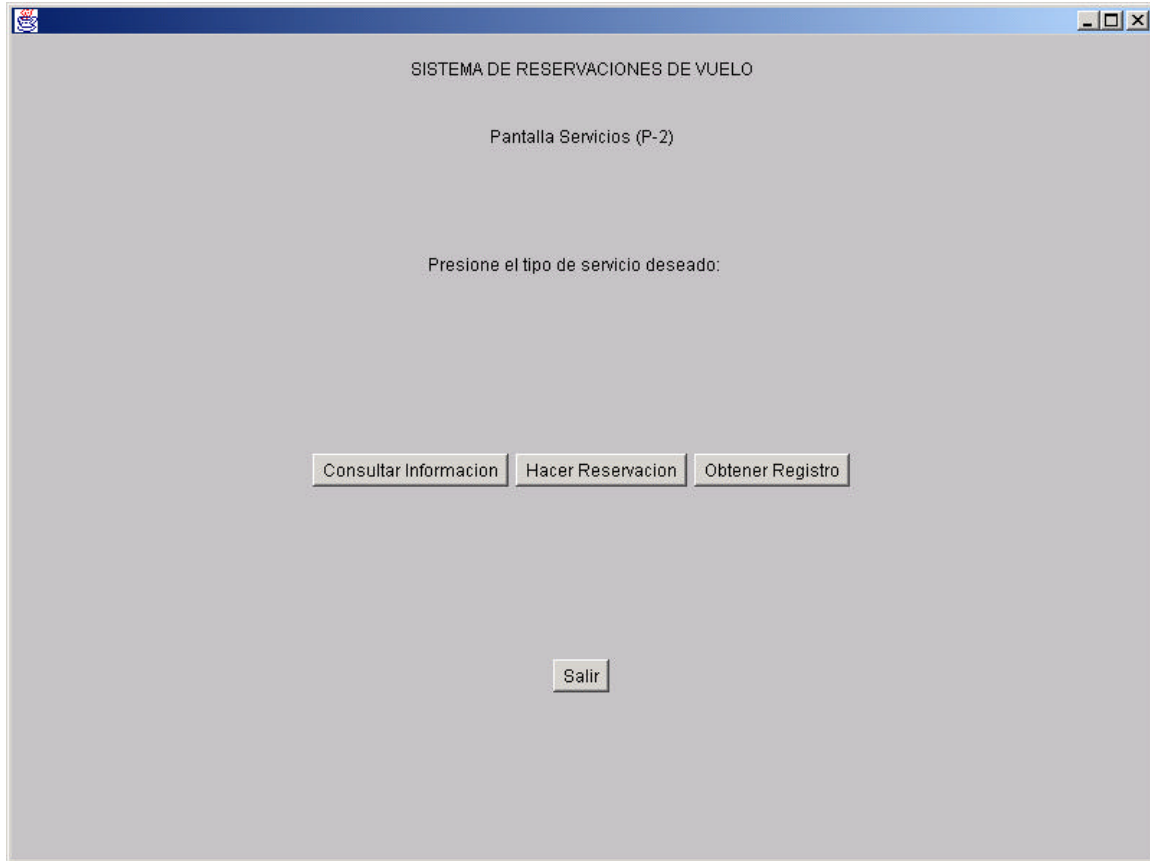


Figura 6.15. Pantalla de Menú de Servicios (P-2).

Registrar Usuario

El caso de uso *Registrar Usuario* está vinculado con el registro inicial del usuario y la modificación de la información de registro. Además, deben ya incluirse los puntos de inclusión y extensión para los casos de uso *Validar Usuario*, *Ofrecer Servicios* y *Registrar Tarjeta*, respectivamente.

Se describe a continuación las secciones iniciales del caso de uso, con las secciones restantes, subflujos y excepciones, más adelante.

| | |
|------------------------|--|
| Caso de Uso | <i>Registrar Usuario</i> |
| Actores | <i>Usuario, Base de Datos Registros</i> |
| Tipo | <i>Básico</i> |
| Propósito | Permitir a un usuario registrarse con el sistema de reservaciones de vuelo para su uso posterior. |
| Resumen | Este caso de uso es iniciado por el <i>Usuario</i> . Ofrece funcionalidad para crear, modificar y eliminar el registro de usuario con el sistema de reservaciones. |
| Precondiciones | Todos los subflujos, con excepción de <i>Crear Registro Usuario (S-1)</i> , requieren ejecutar inicialmente el caso de uso <i>Validar Usuario</i> . |
| Flujo Principal | Se ejecuta el caso de uso <i>Validar Usuario</i> . Dependiendo de las opciones seleccionadas por el Usuario, se continuará con los diversos subflujos de este caso de uso. |

El diseño de la pantalla de registrarse por primera vez se muestra en la Figura 6.16.

SISTEMA DE RESERVACIONES DE VUELO
Pantalla Crear Registro Usuario (P-3)

Nombre: Apellido:

Calle: Colonia:

Ciudad: País: Código Postal:

Tel Casa: Tel Of.: Fax:

Login: E-Mail:

Password Repetir Password

Figura 6.16. Pantalla de Registro de Usuario por Primera Vez (P-3).

El subflujo *Crear Registro Usuario (S-1)* es instanciado al presionar el botón correspondiente de la pantalla principal (P-1) descrito a continuación.

| Subflujos | S-1 <i>Crear Registro Usuario</i> |
|-----------|--|
| | <p>Se presenta al usuario la Pantalla Crear Registro Usuario (P-3). Esta pantalla contiene información de registro que debe ser llenada por el usuario, lo cual incluye nombre, apellido, calle, colonia, ciudad, país, código postal, teléfonos de la casa y oficina, número de fax, <i>login</i>, email, <i>password</i> y una entrada adicional de repetir <i>password</i> para asegurarse de su corrección. El <i>login</i> y el <i>password</i> serán utilizados por el sistema para validar al usuario. El usuario puede seleccionar entre las siguientes actividades: "Registrar" y "Salir". Si el usuario selecciona "Registrar", el sistema genera un nuevo registro de usuario (E-1, E-2, E-3, E-4). Se continúa con el subflujo <i>Administrar Registro Usuario (S-3)</i>. Si la actividad seleccionada es "Salir" se saldrá del sistema (si aún no se ha presionado "Registrar", la información será perdida).</p> |

En la Figura 6.17 se muestra el diseño de la pantalla para la modificación de un registro existente. Nótese que la información que ofrece es exactamente igual a la anterior. La única diferencia son las opciones que se ofrecen, eliminar y actualizar en lugar de registrar.

SISTEMA DE RESERVACIONES DE VUELO
Pantalla Obtener Registro Usuario (P-4)

Nombre: Apellido:

Calle: Colonia:

Ciudad: Pais: Codigo Postal:

Tel Casa: Tel Of: Fax:

Login: E-Mail:

Password Repetir Password

Figura 6.17. Pantalla de Obtener Registro (P-4).

El resto de los subflujos se describen a continuación. El subflujo *Obtener Registro Usuario (S-2)* es instanciado al presionar el botón correspondiente de la pantalla de servicios (P-2). El subflujo *Administrar Registro Usuario (S-3)* es instanciado una vez se presente la información correspondiente en la pantalla de registro (P-4). El subflujo *Actualizar Registro Usuario (S-4)* es instanciado al presionar el botón correspondiente de la pantalla de registro (P-4). El subflujo *Eliminar Registro Usuario (S-5)* es instanciado al presionar el botón correspondiente de la pantalla de registro (P-4).

| | |
|----------------------------|--|
| Subflujos (cont) | <p><i>S-2 Obtener Registro Usuario</i> El sistema obtiene el registro de usuario de la base de datos de registro. Se continúa con el subflujo <i>Administrar Registro Usuario (S-3)</i>.</p> <p><i>S-3 Administrar Registro Usuario</i> Se presenta al usuario la Pantalla Obtener Registro Usuario (P-4) con la información de registro de usuario. El usuario podrá seleccionar entre las siguientes actividades: "Eliminar", "Actualizar", "Registrar Tarjeta", "Servicios" y "Salir". Si el usuario presiona "Actualizar" se ejecuta el subflujo <i>Actualizar Registro Usuario (S-4)</i>. Si el usuario selecciona "Eliminar" se ejecuta el subflujo <i>Eliminar Registro Usuario (S-5)</i>. Si el usuario presiona "Registrar Tarjeta" se continúa con el caso de uso <i>Registrar Tarjeta</i>. Si la actividad seleccionada es "Servicios", se continúa con el caso de uso <i>Ofrecer Servicios</i>. Si la actividad seleccionada es "Salir" se saldrá del sistema (si aún no se ha presionado "Actualizar", la nueva información será perdida).</p> <p><i>S-4 Actualizar Registro Usuario</i> Se actualiza el registro de usuario con la información modificada (E-1, E-3, E-4). Se continúa con el subflujo <i>Administrar Registro Usuario (S-3)</i>.</p> <p><i>S-5 Eliminar Registro Usuario</i> Se elimina el registro de usuario y se continúa con el subflujo <i>Crear Registro Usuario (S-1)</i>.</p> |
|----------------------------|--|

Las excepciones del caso de uso son las siguientes.

| | |
|--------------------|--|
| Excepciones | <p><i>E-1 información incompleta:</i> Falta llenar información en el registro de usuario. Se vuelve a solicitar al usuario que complete el registro.</p> <p><i>E-2 registro ya existe:</i> Si ya existe un registro bajo ese <i>login</i>, se solicitará al usuario que lo cambie o que termine el caso de uso.</p> <p><i>E-3 login incorrecto:</i> El <i>login</i> no es válido. Se le solicita al usuario que corrija el registro.</p> <p><i>E-4 contraseña incorrecta:</i> La contraseña escogida es muy sencilla o no se validó correctamente. Se solicita al usuario que corrija el registro.</p> |
|--------------------|--|

Registrar Tarjeta

El caso de uso *Registrar Tarjeta* es una extensión del caso de uso *Registrar Usuario*. De manera similar a *Registrar Usuario*, el caso de uso *Registrar Tarjeta* está compuesto por un registro inicial y por la modificación de la información ya registrada.

Se describe a continuación las secciones iniciales del caso de uso, con las secciones restantes, subflujos y excepciones, más adelante. Nótese que el flujo principal es llamado al presionar “Registrar Tarjeta” en las pantallas de obtener registro de usuario (*P-4*).

| | |
|------------------------|---|
| Caso de Uso | <i>Registrar Tarjeta</i> |
| Actores | <i>Usuario, Base de Datos Registros</i> |
| Tipo | <i>Extensión</i> |
| Propósito | Permitir a un usuario registrar una tarjeta de créditos con el sistema de reservaciones de vuelo para poder pagar boletos. |
| Resumen | Este caso de uso es iniciado por el <i>Usuario</i> . Ofrece funcionalidad para crear, modificar y eliminar el registro de tarjeta usuario para poder pagar las reservaciones directamente con el sistema de reservaciones. |
| Precondiciones | El usuario ya se debe haberse registrado mediante la activación del caso de uso <i>Registrar Usuario</i> . |
| Flujo Principal | Se continúa con el subflujo <i>Obtener Registro Tarjeta (S-2)</i> . Si no existe un registro de tarjeta válido se continúa con el subflujo <i>Crear Registro Tarjeta (S-1)</i> . De lo contrario, si ya existe uno, se continúa con el subflujo <i>Administrar Registro Tarjeta (S-3)</i> . |

El diseño de la pantalla para registrar la tarjeta por primera vez se muestra en la Figura 6.18.

Figura 6.18. Pantalla de Registro de Tarjeta por Primera Vez (P-5).

El subflujo *Registrar Tarjeta (S-1)* es instanciado al presionar el botón correspondiente de la pantalla servicios (P-5) descrito a continuación.

| | |
|------------------|---|
| Subflujos | <p><i>S-1 Crear Registro Tarjeta</i></p> <p>Se presenta la Pantalla Crear Registro Tarjeta (P-5). La pantalla incluye el nombre como aparece en la tarjeta, número de tarjeta, el tipo de tarjeta, y la fecha de vencimiento.</p> <p>El usuario puede seleccionar entre las actividades "Registrar", "Servicios", "Salir".</p> <p>Si el usuario presiona "Registrar", el sistema verifica la información (E-1), se continúa con el subflujo <i>Administrar Registro Tarjeta (S-3)</i>.</p> <p>Si la actividad seleccionada es "Servicios", se continúa con el caso de uso <i>Ofrecer Servicios</i>.</p> <p>Si la actividad seleccionada es "Salir" se saldrá del sistema (si aún no se ha presionado "Registrar", la nueva información será perdida).</p> |
|------------------|---|

En la Figura 6.19 se muestra el diseño de la pantalla para la modificación de un registro de tarjeta existente. Nótese que la información que ofrece es exactamente igual a la anterior. La única diferencia son las opciones que se ofrecen, eliminar y actualizar en lugar de registrar.

SISTEMA DE RESERVACIONES DE VUELO

Pantalla Obtener Registro Tarjeta (P-6)

Nombre:

Numero de Tarjeta (requerido para pagos):

Tipo Fecha Vencimiento

Eliminar Actualizar Servicios Salir

Figura 6.19. Pantalla de Registro de Tarjeta (P-6).

El resto de los subflujos se describen a continuación. Los subflujos *Obtener Registro Tarjeta (S-2)* y *Administrar Registro Tarjeta (S-3)* son utilizados para leer y manipular el registro de tarjeta, respectivamente. El subflujo *Actualizar Registro Tarjeta (S-4)* es instanciado presionando el botón correspondiente de la pantalla de registro (P-6). El subflujo *Eliminar Registro Tarjeta (S-5)* es instanciado presionando el botón correspondiente de la pantalla de registro (P-6).

| | |
|----------------------------|--|
| Subflujos (cont) | <i>S-2 Obtener Registro Tarjeta</i> El sistema obtiene el registro de tarjeta de la base de datos de registro. Se regresa al flujo anterior. |
| | <i>S-3 Administrar Registro Tarjeta</i> Se presenta la Pantalla Obtener Registro Tarjeta (P-6). La pantalla incluye el nombre como aparece en la tarjeta, número de tarjeta, el tipo de tarjeta, y la fecha de vencimiento. El usuario podrá seleccionar entre las actividades "Eliminar", "Actualizar", "Servicios" y "Salir". Si el usuario presiona "Actualizar" se ejecuta el subflujo <i>Actualizar Registro Tarjeta (S-4)</i> . Si el usuario presiona "Eliminar" se ejecuta el subflujo <i>Eliminar Registro Tarjeta (S-5)</i> . Si la actividad seleccionada es "Servicios", se continúa con el caso de uso <i>Ofrecer Servicios</i> . Si la actividad seleccionada es "Salir" se saldrá del sistema. |
| | <i>S-4 Actualizar Registro Tarjeta</i> Se actualiza el registro de tarjeta con la información modificada (E-1). Se continúa con el subflujo <i>Administrar Registro Tarjeta (S-2)</i> . |
| | <i>S-5 Eliminar Registro Tarjeta</i> Se elimina el registro de tarjeta y se continúa con el subflujo <i>Crear Registro Tarjeta (S-1)</i> . |

Las excepciones del caso de uso son las siguientes.

| | |
|--------------------|---|
| Excepciones | <i>E-1 información incompleta:</i> Falta llenar información indispensable para completar el registro de tarjeta. Se le vuelve a pedir al usuario que complete el registro de tarjeta. |
|--------------------|---|

Consultar Información

El caso de uso *Consultar Información* es instanciado a partir de la pantalla de servicios (P-2) una vez se haya validado el usuario y haya seleccionado el botón correspondiente. Este caso de uso es el más complejo de todos los del sistema ya que incluye tres tipos de consultas distintas: consulta de vuelos, consulta de tarifas y consulta de estado de un vuelo.

El caso de uso se describe a continuación, excluyendo los subflujos y excepciones que se describen más adelante.

| | |
|------------------------|---|
| Caso de Uso | <i>Consultar Información</i> |
| Actores | <i>Usuario, Base de Datos Reservas</i> |
| Tipo | <i>Básico</i> |
| Propósito | Permitir a un usuario consultar información con el sistema de reservaciones de vuelo. |
| Resumen | Este caso de uso es iniciado por el <i>Usuario</i> . Ofrece funcionalidad para consultar información de horarios, tarifas y estado de vuelos con el sistema de reservaciones. |
| Precondiciones | Se requieren haber ejecutado anteriormente el caso de uso <i>Validar Usuario</i> . |
| Flujo Principal | Se ejecuta el caso de uso <i>Validar Usuario</i> . Dependiendo de las opciones seleccionadas por el Usuario, se continuará con los diversos subflujos de este caso de uso. |

Antes de proseguir con la descripción del caso de uso, mostramos la pantalla que permite tomar esta decisión, como se muestra en la Figura 6.20.

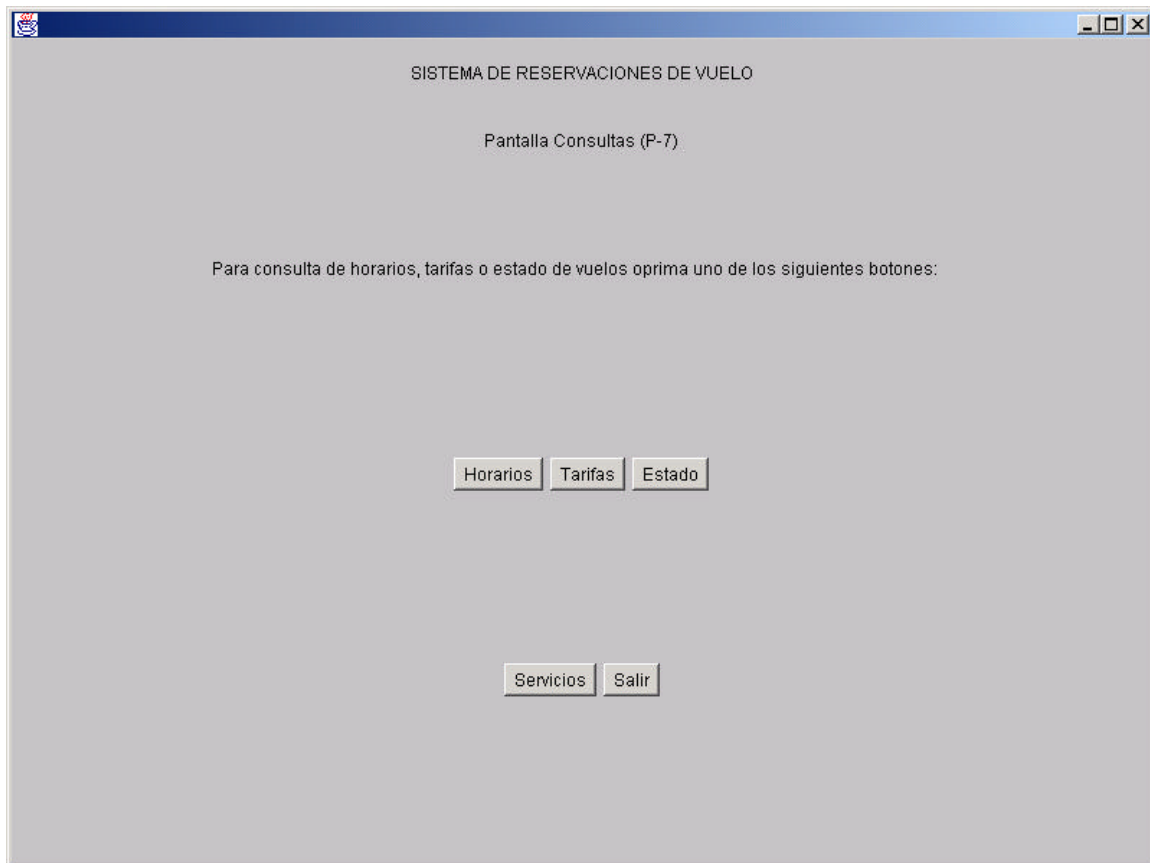


Figura 6.20. Pantalla de Selección de Tipo de Consulta (P-7).

Dado que el usuario puede iniciar una consulta de varias páginas distintas, como se verá posteriormente, se incluye un primer subflujo para iniciar estas consultas llamado *Consultar* (S-1), como se muestra a continuación.

| | |
|------------------|---|
| Subflujos | <p><i>S-1 Consultar</i> Se despliega la Pantalla Consultas (P-7). El usuario puede seleccionar entre las siguientes actividades: "Horarios", "Tarifas", "Estado", "Servicios" y "Salir". Si el usuario presiona "Horarios", se activa el subflujo <i>Consultar Horarios (S-2)</i>. Si el usuario presiona "Tarifas", se activa el subflujo <i>Consultar Tarifas (S-4)</i>. Si el usuario presiona "Estado", se activa el subflujo <i>Consultar Estado (S-6)</i>. Si el usuario presiona "Servicios", se continúa con el caso de uso <i>Ofrecer Servicios</i>. Si el usuario presiona "Salir" se sale del sistema.</p> |
|------------------|---|

En la Figura 6.21 se muestra el diseño de la pantalla para la consulta de horarios de vuelos, correspondiente al subflujo *Consultar Horarios (S-2)*.

SISTEMA DE RESERVACIONES DE VUELO

Pantalla Consulta Horarios (P-8)

Origen (Ciudad o Código): Destino (Ciudad o Código):

PREFERENCIAS

Aerolínea: Horario:

Solo Directo

Figura 6.21. Pantalla de Consulta de Horarios de Vuelos (P-8).

En la Figura 6.22 se muestra el diseño de la pantalla para el resultado de la consulta de horarios de vuelos, correspondiente al subflujo *Devolver Horarios (S-3)*.

Figura 6.22. Pantalla de Resultado de Consulta de Horarios de Vuelos (P-9).

Los subflujos *Consultar Horarios (S-2)* y *Devolver Horarios (S-3)* se muestran a continuación.

| | |
|-------------------------|--|
| <p>Subflujos</p> | <p><i>S-2 Consultar Horarios</i> Se presenta al usuario la Pantalla Consulta Horarios (P-8). Esta pantalla debe ser llenada con información de ciudad de origen y destino, y preferencias opcionales de: aerolínea, horario y una opción de vuelo sólo directo. El usuario puede seleccionar de las siguientes actividades: "Consultar", "Servicios" y "Salir". Si el usuario presiona "Consultar", el sistema recibe la información (E-1,E-2), se continúa con el subflujo <i>Devolver Horarios (S-3)</i>. Si el usuario presiona "Servicios", se continúa con el caso de uso <i>Ofrecer Servicios</i>. Si el usuario presiona "Salir" se sale del sistema.</p> |
| | <p><i>S-3 Devolver Horarios</i> Se presenta la Pantalla Resultado Horarios (P-9) conteniendo información sobre los diferentes vuelos encontrados. La información incluye la aerolínea, vuelo, días, horario, y restricciones, tales como fecha de inicialización o terminación del vuelo. Al principio de cada fila se encuentra una opción de selección para obtener información adicional sobre el vuelo. El usuario puede seleccionar entre las siguientes opciones: "+", "-", "Nueva Consulta", "Servicios" y "Salir". Si el usuario presiona "+" se muestran resultados adicionales de horarios. Se continúa al inicio de este subflujo. Si el usuario presiona "-" se muestran resultados anteriores de horarios. Se continúa al inicio de este subflujo. Si el usuario presiona "Nueva Consulta", se continúa con el subflujo <i>Consultar Horarios (S-2)</i>. Si la actividad seleccionada es "Servicios", se continúa con el caso de uso <i>Ofrecer Servicios</i>. Si el usuario presiona "Salir" se sale del sistema.</p> |

En la Figura 6.23 se muestra el diseño de la pantalla para la consulta de tarifas de vuelos, correspondiente al subflujo *Consultar Tarifas (S-4)*.

SISTEMA DE RESERVACIONES DE VUELO
Pantalla Consulta Tarifas (P-10)

Origen (Ciudad o Código): Destino (Ciudad o Código):

PREFERENCIAS

Fecha de Salida: Fecha de Regreso:

Aerolínea: Clase:

Menor Tarifa Solo Directo Redondo

Figura 6.23. Pantalla de Consulta de Tarifas de Vuelos (P-10).

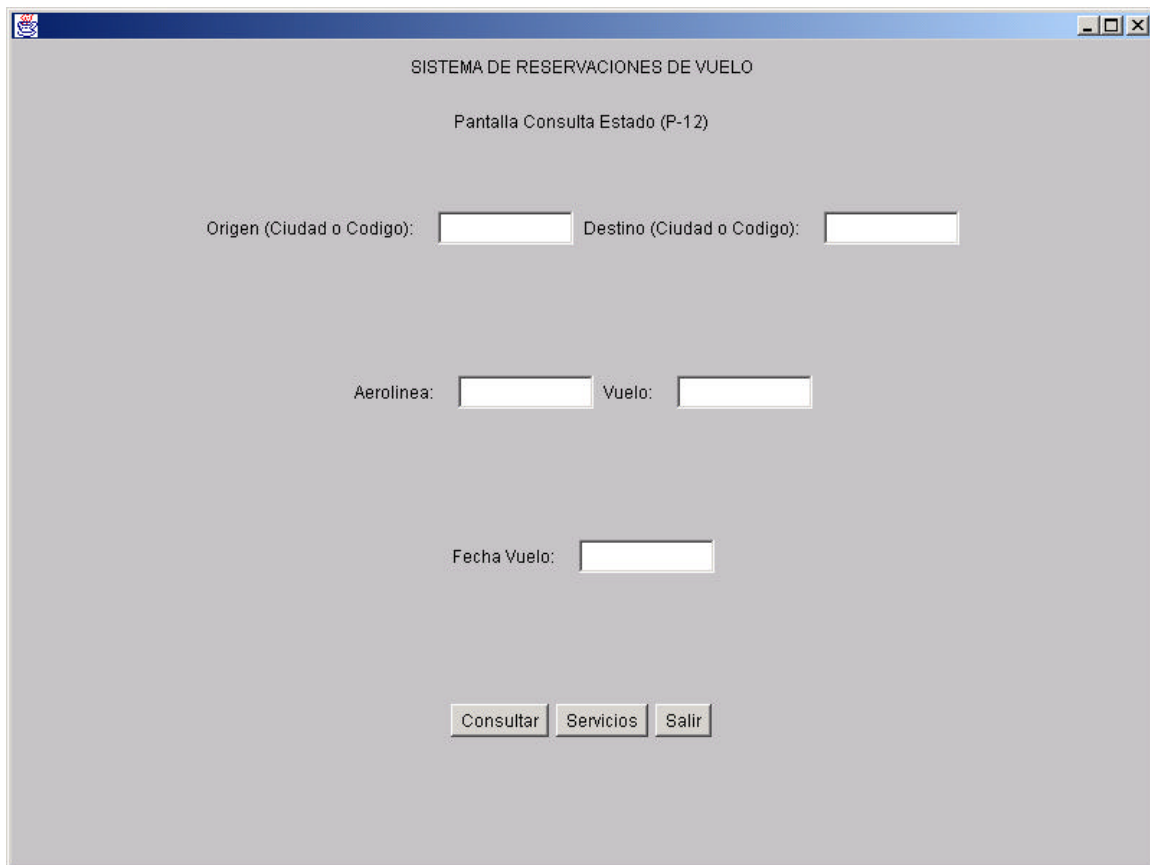
En la Figura 6.24 se muestra el diseño de la pantalla para el resultado de la consulta de tarifas de vuelos, correspondiente al subflujo *Devolver Tarifas (S-5)*.

Figura 6.24. Pantalla de Resultado de Consulta de Tarifas de Vuelos (P-11).

Los subflujos *Consultar Tarifas (S-4)* y *Devolver Tarifas (S-5)* se muestran a continuación.

| | |
|------------------------------------|--|
| <p>Subflujos (cont)</p> | <p><i>S-4 Consultar Tarifas</i> Se presenta al usuario la Pantalla Consultar Tarifas (P-10). Esta pantalla debe ser llenada con información de ciudad de origen y destino, y preferencias opcionales: fecha de salida, fecha de regreso, aerolínea, clase, y las opciones de organizar la información por menor tarifa, una opción de vuelo sólo directo, y si la tarifa se basa en viaje redondo. El usuario puede seleccionar de las siguientes actividades: "Consultar", "Servicios" y "Salir". Si el usuario presiona "Consultar", el sistema recibe la información (E-1,E-2), se continúa con el subflujo <i>Devolver Tarifas (S-5)</i>. Si el usuario presiona "Servicios", se continúa con el caso de uso <i>Ofrecer Servicios</i>. Si el usuario presiona "Salir" se sale del sistema.</p> <hr/> <p><i>S-5 Devolver Tarifas</i> Se presenta la Pantalla Resultado Tarifas (P-11) conteniendo información sobre los diferentes vuelos encontrados. La información incluye la aerolínea, vuelo, días, horario, tarifa ida e ida y vuelta y restricciones correspondientes. Al principio de cada fila se encuentra una opción para seleccionar en caso de hacer consultas o reservas sobre los vuelos obtenidos. El usuario puede seleccionar entre las siguientes opciones: "+", "-", "Nueva Consulta", "Servicios" y "Salir". Si el usuario presiona "+" se muestran resultados adicionales de horarios. Se continúa al inicio de este subflujo. Si el usuario presiona "-" se muestran resultados anteriores de horarios. Se continúa al inicio de este subflujo. Si el usuario presiona "Nueva Consulta" se continúa con el subflujo <i>Consultar Tarifas (S-4)</i>. Si el usuario presiona "Servicios", se continúa con el caso de uso <i>Ofrecer Servicios</i>. Si el usuario presiona "Salir" se sale del sistema.</p> |
|------------------------------------|--|

En la Figura 6.25 se muestra el diseño de la pantalla para la consulta de estado de vuelo, correspondiente al subflujo *Consultar Estado (S-6)*.



SISTEMA DE RESERVACIONES DE VUELO

Pantalla Consulta Estado (P-12)

Origen (Ciudad o Código): Destino (Ciudad o Código):

Aerolínea: Vuelo:

Fecha Vuelo:

Figura 6.25. Pantalla de Consulta de Estado de Vuelo (P-12).

En la Figura 6.26 se muestra el diseño de la pantalla para el resultado de la consulta de estado de vuelo, correspondiente al subflujo *Devolver Estado (S-7)*.

Figura 6.26. Pantalla de Resultado de Consulta de Estado de Vuelo (P-13).

Los subflujos de *Consultar Estado (S-6)* y *Devolver Estado (S-7)* se muestran a continuación.

| | |
|------------------------------------|--|
| <p>Subflujos (cont)</p> | <p><i>S-6 Consultar Estado</i> Se presenta al usuario la Pantalla Consultar Estado (P-12). Esta pantalla debe ser llenada con información de ciudad de origen y destino, la aerolínea, el número de vuelo, y la opción de vuelo de hoy. El usuario puede seleccionar de las siguientes actividades: "Consultar", "Servicios" y "Salir". Si el usuario presiona "Consultar", el sistema recibe la información (E-1,E-2) y se continúa con el subflujo <i>Devolver Estado (S-7)</i>. Si el usuario presiona "Servicios", se continúa con el caso de uso <i>Ofrecer Servicios</i>. Si el usuario presiona "Salir" se sale del sistema.</p> <hr/> <p><i>S-7 Devolver Estado</i> Se presenta la Pantalla Resultado Estado (P-13) conteniendo información sobre los diferentes vuelos encontrados. La pantalla contiene información sobre el vuelo, incluyendo su estado, por ejemplo, confirmado, retrasado, cancelado. Al principio de cada fila se encuentra una opción para seleccionar en caso de hacer consultas o reservas sobre los vuelos obtenidos. El usuario puede seleccionar entre las siguientes opciones: "Nueva Consulta", "Servicios" y "Salir". Si el usuario presiona "Nueva Consulta", se continúa con el subflujo <i>Consultar Estado (S-6)</i>. Si la actividad seleccionada es "Servicios", se continúa con el caso de uso <i>Ofrecer Servicios</i>. Si el usuario presiona "Salir" se sale del sistema.</p> |
|------------------------------------|--|

Las excepciones del caso de uso son las siguientes.

| | |
|--------------------|---|
| Excepciones | <p><i>E-1 información incompleta:</i> Falta llenar información indispensable, ciudad de origen o de destino. Se le vuelve a pedir al usuario la información.</p> <p><i>E-2 información inválida:</i> Una de las entradas de la solicitud es incorrecta.</p> |
|--------------------|---|

Hacer Reservación

El caso de uso *Hacer Reservación* es instanciado a partir de la pantalla de servicios (*P-2*) una vez se haya validado el usuario y haya seleccionado el botón correspondiente.

El caso de uso se describe a continuación, excluyendo los subflujos y excepciones que se describen más adelante.

| | |
|------------------------|--|
| Caso de Uso | <i>Hacer Reservación</i> |
| Actores | <i>Usuario, Base de Datos Reservas</i> |
| Tipo | <i>Básico</i> |
| Propósito | Permitir a un usuario hacer reservaciones con el sistema de reservaciones de vuelo. |
| Resumen | Este caso de uso es iniciado por el <i>Usuario</i> . Ofrece funcionalidad para crear, obtener, modificar y eliminar reservaciones de vuelos con el sistema de reservaciones. |
| Precondiciones | Se debe haber ejecutado anteriormente el caso de uso <i>Validar Usuario</i> . |
| Flujo Principal | Se ejecuta el caso de uso <i>Validar Usuario</i> . Dependiendo de las opciones seleccionadas por el Usuario, se continuará con los diversos subflujos de este caso de uso. |

En la Figura 6.27 se muestra el diseño de la pantalla para crear u obtener una reservación si se cuenta con una clave.

SISTEMA DE RESERVACIONES DE VUELO

Pantalla Clave Reserva (P-14)

Inserte la Clave de la Reserva si ya existe:

Clave:

Crear Obtener

Servicios Salir

Figura 6.27. Pantalla de Inserción Clave de Reserva (*P-14*).

El subflujo *Solicitar Clave Reservación (S-1)* se muestra a continuación.

| | |
|------------------|--|
| Subflujos | <p><i>S-1 Solicitar Clave Reservación</i></p> <p>Se presenta al usuario la Pantalla Clave Reserva (P-14). El usuario puede seleccionar entre las siguientes actividades: "Crear", "Obtener", "Servicios" y "Salir".</p> <p>Si el usuario presiona "Crear" se ejecutará el subflujo <i>Crear Reservación (S-2)</i>.</p> <p>Si el usuario presiona "Obtener" (E-1) se ejecuta el subflujo <i>Obtener Reservación (S-3)</i>.</p> <p>Si la actividad seleccionada es "Servicios", se continúa con el caso de uso <i>Ofrecer Servicios</i>.</p> <p>Si la actividad seleccionada es "Salir" se saldrá del sistema.</p> |
|------------------|--|

En la Figura 6.28 se muestra el diseño de la pantalla para la solicitud de reserva de vuelos, utilizado por los distintos subflujos.

SISTEMA DE RESERVACIONES DE VUELO
Pantalla Crear Reserva (P-15)

Apellido: Nombre:

Numero de Viajero Frecuente:

Aerolinea: Vuelo:

Sale de: Llega a:

Solicitar Asiento Fecha: Clase:

Asiento: Ventana Pasillo Comida: Veg. Carne

Agregar

Figura 6.28. Pantalla de Solicitud de Reserva de Vuelos (P-15).

En la Figura 6.29 se muestra el diseño de la pantalla para el récord de reserva de vuelos, utilizado por los distintos subflujos.

Figura 6.29. Pantalla de Récord de Reserva de Vuelos (P-16).

Los demás subflujos se muestra a continuación.

| | |
|------------------------------------|--|
| <p>Subflujos (cont)</p> | <p><i>S-2 Crear Reservación</i> Se presenta la Pantalla Crear Reserva (P-15). Esta pantalla debe ser llenada con información de apellido y nombre del pasajero, un número de viajero frecuente opcional, aerolínea, número de vuelo, ciudad de origen y destino, fecha, clase, una opción de solicitar asiento y si desea ventana o pasillo, y opcionalmente comida vegetal o carne. El usuario puede seleccionar entre las siguientes actividades: "Agregar", "Borrar", "+", "-", "Reservar", "Servicios" y "Salir". Si el usuario selecciona "Agregar", el sistema agrega una nueva Pantalla Crear Reserva (P-15) para ser llenada por el usuario. Si el usuario selecciona "Borrar", el sistema borra los datos recién insertados y se continúa con la creación de reservas. Si el usuario selecciona "+", el sistema avanza a la siguiente pantalla de reservación. Si el usuario selecciona "-", el sistema retrocede a la pantalla anterior de reservación. Si el usuario selecciona "Reservar", el sistema acepta la solicitud (E-2,E-3), enviándola a la base de datos del sistema de reservaciones (E-5), se continua con el subflujo <i>Administrar Reservación (S-3)</i>. Si la actividad seleccionada es "Servicios", se continúa con el caso de uso <i>Ofrecer Servicios</i>. Si la actividad seleccionada es "Salir" se saldrá del sistema.</p> <hr/> <p><i>S-3 Obtener Reservación</i> El sistema obtiene el récord de reservación de la base de datos de registro. Se continúa con el subflujo <i>Administrar Reservación (S-4)</i>.</p> |
|------------------------------------|--|

| |
|---|
| <p>S-4 Administrar Reservación Se presenta la Pantalla Record Reserva (P-16) con la opción a modificar la información (E-1). El usuario puede seleccionar entre las siguientes selecciones: "Eliminar", "Actualizar", "+", "-", "Nueva Reserva", "Pagar", "Reembolsar", "Servicios" y "Salir". Si el usuario presiona "Actualizar" se ejecuta el subflujo <i>Actualizar Reservación</i> (S-5). Si el usuario presiona "Eliminar" se ejecuta el subflujo <i>Eliminar Reservación</i> (S-6). Si el usuario selecciona "+", el sistema avanza a la siguiente pantalla de reservación. Si el usuario selecciona "-", el sistema retrocede a la pantalla anterior de reservación. Si el usuario selecciona "Nueva Reserva", se continúa con el subflujo <i>Crear Reservación</i> (S-2). Si el usuario selecciona "Pagar", el sistema se continúa con el caso de uso <i>Pagar Reservación</i>. Si el usuario selecciona "Reembolsar", el sistema se continúa con el caso de uso <i>Pagar Reservación</i>. Si la actividad seleccionada es "Servicios", se continúa con el caso de uso <i>Ofrecer Servicios</i>. Si la actividad seleccionada es "Salir" se saldrá del sistema.</p> |
| <p>S-5 Actualizar Reservación Se actualiza el récord de reserva (E-2, E-3, E-4). Se continúa con el subflujo <i>Administrar Reservación</i> (S-3).</p> |
| <p>S-6 Eliminar Reservación Se elimina el récord de reserva (E-5). Se continúa con el subflujo <i>Crear Reservación</i> (S-2).</p> |

Las excepciones del caso de uso son las siguientes.

| | |
|--------------------|--|
| Excepciones | <p><i>E-1 récord inválido:</i> No existe el récord especificado. <i>E-2 información incompleta:</i> Falta llenar información indispensable, ciudad de origen o de destino. Se le vuelve a pedir al usuario la información. <i>E-3 información inválida:</i> Una de las entradas de la solicitud es incorrecta. <i>E-4 reserva sin éxito:</i> No se logró obtener una reserva. <i>E-5 eliminación reserva sin éxito:</i> No se logró eliminar la reserva.</p> |
|--------------------|--|

Pagar Reservación

El caso de uso *Pagar Reservación* es instanciado a partir de la pantalla de reservaciones (P-14) una vez se haya seleccionado el botón correspondiente.

El caso de uso se describe a continuación, excluyendo los subflujos y excepciones que se describen más adelante.

| | |
|------------------------|---|
| Caso de Uso | <i>Pagar Reservación</i> |
| Actores | <i>Usuario, Base de Datos Reservas</i> |
| Tipo | <i>Extensión</i> |
| Propósito | Permitir a un usuario pagar reservaciones con el sistema de reservaciones de vuelo. |
| Resumen | Este caso de uso es iniciado por el <i>Usuario</i> . Ofrece funcionalidad para pagar y reembolsar pagos de reservaciones de vuelos con el sistema de reservaciones mediante tarjetas de crédito registradas con el sistema. |
| Precondiciones | Se requieren haber ejecutado anteriormente el caso de uso <i>Validar Usuario</i> y tener ya hecha una reserva mediante el caso de uso <i>Hacer Reservación</i> . |
| Flujo Principal | Se obtiene el registro de tarjeta ejecutando el caso de uso <i>Registrar Tarjeta</i> , subflujo <i>Obtener Registro Tarjeta</i> (S-2). Dependiendo si la solicitud original fue pagar, se continúa con el subflujo <i>Pagar Reservación</i> (S-1), si fue reembolsar se continúa con el subflujo <i>Reembolsar Pago</i> (S-2). |

En la Figura 6.30 se muestra el diseño de la pantalla para el pago de reserva de vuelos, utilizado el subflujo *Pagar Reservación* (S-1).

SISTEMA DE RESERVACIONES DE VUELO

Pantalla Pagar Tarjeta (P-17)

Nombre:

Numero de Tarjeta (requerido para pagos):

Tipo Fecha Vencimiento

Tarifa:

Figura 6.30. Pantalla de Pago de Reserva de Vuelos (P-17).

El subflujo *Pagar Reservación (S-1)* se muestra a continuación.

| | |
|------------------|--|
| Subflujos | <p><i>S-1 Pagar Reservación</i></p> <p>Se presenta al usuario la Pantalla Pagar Registro Tarjeta (P-17), la cual incluye información de nombre como aparece en la tarjeta, número de tarjeta, el tipo de tarjeta, la fecha de vencimiento y la cantidad a pagar (E-1).</p> <p>El <u>Usuario</u> podrá seleccionar entre las siguientes actividades: "Pagar", "Servicios" y "Salir".</p> <p>Si la actividad seleccionada es "Pagar", el sistema utiliza los datos de la tarjeta registrada por el usuario y envía una pantalla de confirmación al usuario (E-2). Se continúa con el caso de uso <i>Hacer Reservación</i>, subflujo <i>Solicitar Clave Reservación (S-1)</i>.</p> <p>Si la actividad seleccionada es "Servicios", se continúa con el caso de uso <i>Ofrecer Servicios</i>.</p> <p>Si la actividad seleccionada es "Salir" se sale del sistema.</p> |
|------------------|--|

En la Figura 6.31 se muestra el diseño de la pantalla para el pago de reserva de vuelos, utilizado el subflujo *Reembolsar Pago (S-2)*.

SISTEMA DE RESERVACIONES DE VUELO

Pantalla Reembolsar Tarjeta (P-18)

Nombre:

Numero de Tarjeta (requerido para pagos):

Tipo Fecha Vencimiento

Tarifa:

Reembolsar Servicios Salir

Figura 6.31. Pantalla de Reembolso de Reserva de Vuelos (P-18).

El subflujo *Reembolsar Pago (S-2)* se muestra a continuación.

| | |
|------------------------------------|---|
| <p>Subflujos (cont)</p> | <p><i>S-2 Reembolsar Pago</i></p> <p>Se presenta al usuario la Pantalla Reembolsar Registro Tarjeta (P-18), la cual incluye información de nombre como aparece en la tarjeta, número de tarjeta, el tipo de tarjeta, la fecha de vencimiento y la cantidad a reembolsar (E-1).</p> <p>El <u>Usuario</u> podrá seleccionar entre las siguientes actividades: "Reembolsar", "Servicios" y "Salir".</p> <p>Si la actividad seleccionada es "Reembolsar", el sistema utiliza los datos de la tarjeta registrada por el usuario y envía una pantalla de confirmación al usuario (E-3, E-4). Se continúa con el caso de uso <i>Hacer Reservación</i>, subflujo <i>Solicitar Clave Reservación (S-1)</i>.</p> <p>Si la actividad seleccionada es "Servicios", se continúa con el caso de uso <i>Ofrecer Servicios</i>.</p> <p>Si la actividad seleccionada es "Salir" se sale del sistema.</p> |
|------------------------------------|---|

Las excepciones del caso de uso son las siguientes.

| | |
|---------------------------|---|
| <p>Excepciones</p> | <p><i>E-1 récord inválido:</i> No existe el récord especificado. El usuario deberá insertar los datos de la tarjeta.</p> <p><i>E-2 pago inválido:</i> El pago no tuvo éxito o la información de pago está incompleta.</p> <p><i>E-3 pago inexistente:</i> La reserva no ha sido aún pagada.</p> <p><i>E-4 reembolso inválido:</i> No se pudo hacer el reembolso del pago.</p> |
|---------------------------|---|

6.5 Modelo del Dominio del Problema

El *modelo del dominio del problema* define un modelo de clases común para todos los involucrados en el modelo de requisitos, analistas al igual que clientes. Este modelo de clases consiste de los objetos del dominio del problema, o

sea objetos que tienen una correspondencia directa en el área de la aplicación. Como los usuarios y clientes deberían reconocer todos los conceptos, se puede desarrollar una terminología común al razonar sobre los casos de uso, y por lo tanto disminuyendo la probabilidad de malos entendimientos entre el analista y el usuario. Al discutirlo, se evolucionará el modelo del dominio del problema. Una técnica utilizada cuando se trabaja con tal modelo es darle al cliente un papel y un lápiz y pedirle que dibuje su visión del sistema.

Históricamente, el modelo del dominio del problema se utilizaba como el modelo de requisitos fundamental en ciertas metodologías, tales como Coad y Yourdon (1991), Booch (1991), y Rumbaugh (1991). Sin embargo, dadas sus limitaciones que impedía obtener los requisitos funcionales de un sistema, el modelo del dominio del problema dejó de ser la base única para el desarrollo completo del sistema y pasó a ser un elemento adicional en la especificación de los sistemas, como en el modelo de casos de uso. El propósito principal del dominio del problema en el modelo de requisitos de nuestra metodología es formar una base común de entendimiento del desarrollo y no para definir el sistema completo. Por lo tanto, se pueden aprovechar algunas de las heurísticas de los métodos anteriores para la identificación de los objetos en el dominio del problema, logrando un *glosario o diccionario de clases* que sirve como común denominador a todos los componentes del sistema, incluyendo a las diversas personas involucradas a lo largo del desarrollo. A diferencia de los métodos anteriores, el modelo del dominio del problema no debe ser demasiado extenso, ya que varios grados de refinamiento serán hechos posteriormente. Y aunque es suficiente describir el dominio del problema en término de objetos o clases, es posible refinar más aún mediante la inclusión de asociaciones, atributos, herencia y operaciones, siempre y cuando esto ayude a comprender mejor el problema y no se vuelva un esfuerzo demasiado grande durante esta etapa. Se debe tener cuidado con hacer demasiado trabajo temprano ya que esto puede incluso dificultar su modificación posterior durante el modelo de análisis. La experiencia muestra que muchos (si no todos) los objetos del dominio podrán aparecer durante el modelo de análisis. Sin embargo, pueden haber cambios durante el modelo de análisis, incluyendo la eliminación de clase identificadas durante esta etapa, o incluso la incorporación de clases adicionales.

En esta sección describiremos cómo identificar las clases del dominio del problema junto con aspectos adicionales, cómo asociaciones y atributos. Lo que definitivamente no se hará aquí, y que era parte esencial de las metodologías anteriores, es identificar herencia y las operaciones durante esta etapa. La herencia y en especial las operaciones de un sistema son los aspectos de mayor complejidad, algo que nosotros elaboraremos de manera muy cuidadosa durante el diseño del sistema.

Identificación de Clases

La identificación de clases del dominio del problema se obtiene principalmente de algún documento textual que describa el sistema. Aunque pudiéramos tomar como punto de partida los documentos desarrollados para el modelo de casos de uso, a menudo la descripción original del problema es suficiente. Se comienza este proceso mediante la identificación de las clases *candidatas*, explícitas o implícitas, a las que se refiera la descripción del problema. Para ello se extraen todos los sustantivos de la descripción del problema o de algún otro documento similar, de acuerdo a las siguientes consideraciones.

- ?? Los sustantivos en la descripción del problema son los posibles candidatos a clases de objetos. Por ejemplo en "Un sistema de reservaciones que vende boletos para funciones a varios teatros", las clases candidatas serían, *Sistema de Reservaciones, Boletos, Función y Teatro*.
- ?? Durante esta etapa, se debe identificar entidades físicas al igual que entidades conceptuales.
- ?? No se debe tratar de diferenciar entre clases y atributos durante ésta etapa.
- ?? Dado que no todas las clases se describen de manera explícita, siendo algunas implícitas en la aplicación, será necesario añadir clases que pueden ser identificadas por nuestro conocimiento del área.
- ?? Se debe revisar los pronombres en la descripción del problema para asegurar que no se haya perdido ningún sustantivo descrito de forma implícita.
- ?? Para facilitar la identificación de clases, se subrayan todos los sustantivos de la descripción del problema.

En el caso del sistema de reservaciones de vuelos, partimos de la descripción del problema y subrayamos todos los sustantivos, como se ve a continuación:

El Sistema de Reservaciones de Vuelos es un sistema que permite al usuario hacer consultas y reservas de vuelos, además de poder comprar los boletos aéreos de forma remota, sin la necesidad de recurrir a un agente de viajes humano. Se desea que el sistema de reservaciones sea accesible a través del Internet (World Wide Web).

El sistema presenta en su pantalla principal un mensaje de bienvenida describiendo los servicios ofrecidos junto con la opción para registrarse por primera vez, o si ya se está registrado, poder utilizar el sistema de reservaciones de vuelos. Este acceso se da por medio de la inserción de un login previamente especificado y un password previamente escogida y que debe validarse.

Una vez registrado el usuario, y después de haberse validado el registro y contraseña del usuario, se pueden seleccionar de las siguientes actividades:

Consulta de vuelos

Reserva de vuelos

Compra de boletos

La consulta de vuelos se puede hacer de tres maneras diferentes:

Horarios de Vuelos

Tarifas de Vuelos

Estado de Vuelo

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

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

La información de vuelo se utiliza principalmente para consultar el estado de algún vuelo, incluyendo información de si existen asientos disponibles y de si, en el caso de un vuelo para el mismo día, si éste está en hora.

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

La reservación de vuelo permite al cliente hacer una reserva para un vuelo particular, especificando la fecha y 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 reserva de vuelo previa y una tarjeta de crédito válida, adquirir los boletos aéreos.

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

Es necesario estar previamente registrados 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 el usuario validado en el sistema.

A partir de estos sustantivos se prepara una lista inicial de clases candidatas, como se muestra en la Tabla 6.1. Se debe excluir clases repetidas, manteniendo todos los nombres en singular.

| Clases Candidatas | | |
|-----------------------------------|--------------------|----------------------|
| Sistema de Reservaciones de Vuelo | Login | Información |
| Sistema | Email | Asiento |
| Usuario | Password | Día |
| Consulta | Registro | Hora |
| Reserva | Actividad | Preferencia |
| Vuelo | Consulta de Vuelos | Búsqueda |
| Boleto Aéreo | Reserva de Vuelos | Fecha |
| Agente de Viajes Humano | Compra de Boletos | Categoría de Asiento |

| | | |
|--|--|--|
| Sistema de Reservas Internet World Wide Web Pantalla Principal Mensaje de Bienvenida Servicios Opción Reservaciones de Vuelos Acceso | Horario de Vuelos Tarifa de Vuelos Información de Vuelo Horario Aerolínea Ciudad Tarifa Costo Estado | Vuelo Directo Cliente Itinerario Pasajero Pago Tarjeta de Crédito Boleto Mostrador del Aeropuerto Número de Tarjeta de Crédito |
|--|--|--|

Tabla 6.1. Clases Candidatas para el sistema de reservas de vuelo identificadas de la descripción del problema.

Selección de Clases

A partir de las clases candidatas, se debe seleccionar las clases relevantes tomando en cuenta los siguientes consideraciones:

- ?? Todas las clases deben tener sentido en el área de la aplicación, la relevancia al problema debe ser el único criterio para la selección.
- ?? Como regla general, se debe escoger los nombres para las clases con cuidado, que no sean ambiguos y que mejor se apliquen al problema. Este es uno de los procesos más difíciles. Los nombres deben ser establecidos con un formato consistente (e.g. nombres en singular).
- ?? No hay que preocuparse durante esta etapa sobre asociación, agregación, o herencia. Primero hay que tener las clases específicas correctas para no suprimir detalles en el intento de ajustarse a estructuras preconcebidas.
- ?? Ante la duda, se deben conservar las clases, ya que posteriormente siempre habrá oportunidad para eliminarlas.
- ?? Se deben eliminar clases redundantes, si estas expresan la misma información. La clase más descriptiva debe ser guardada.
- ?? Se deben eliminar clases irrelevantes, que tienen poco o nada que ver con el problema. Esto requiere juicio porque en un contexto una clase puede ser importante mientras que en otro contexto la clase podría no serlo.
- ?? Se deben clarificar las clases imprecisas. Algunas clases pueden tener bordes mal definidos o demasiado generales.
- ?? Se deben eliminar las clases que debieran ser atributos más que clases, cuando los nombres corresponden a propiedades más que entidades independientes.
- ?? Se deben eliminar las clases que debieran ser roles más que clases, cuando los nombres corresponden al papel que juegan las clases más que entidades independientes.
- ?? Se deben eliminar las clases que debieran ser operaciones más que clases, si las entidades representan operaciones que son aplicadas a los objetos y no entidades manipuladas por si mismas.
- ?? Se deben eliminar las clases que corresponden a construcciones de implementación si están alejadas del mundo real, por lo cual deben ser eliminadas del análisis. No se necesita una clase para representar una entidad física. Por ejemplo: *subrutinas, listas, y arreglos*, son clases típicas de implementación; *Internet y World Wide Web* son el medio de implementación del sistema
- ?? Se debe eliminar clases que correspondan aspectos de interfaces de usuario y no de la aplicación.
- ?? Se debe eliminar clases que correspondan a todo un sistema completo.
- ?? Se debe eliminar clases que correspondan a actores del sistema.
- ?? Se deben agregar clases implícitas que no aparezcan en la descripción del problema.

Tomaremos algunas de las clases candidatas del sistema de reservas de vuelo identificadas anteriormente y seleccionamos las que mejor se apliquen a nuestro problema, como se ve a continuación:

- ?? Clases redundantes: *Cliente* y *Usuario*. Esta decisión puede ir para ambos lados de igual manera. En el caso del Sistema de Reservas, consideramos que *Usuario* es más descriptivo por ser la persona que utilice el sistema y se guarda.
- ?? Clases irrelevantes: *Mostrador del Aeropuerto*, *Agente de Viajes Humano* y *Boleto Aéreo*. Si el sistema generara o se refiriera a un boleto aéreo, esta clase se mantendría.
- ?? Clases imprecisas: *Sistema*, *Servicios*, *Actividad*, *Preferencia*, *Búsqueda*, *Información*, *Estado*, *Opción*, *Acceso*, *Itinerario*, son clases imprecisas. Durante la introducción de herencia puede que sea necesario una clase para compartir aspectos comunes a ambas 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* (*Asiento*), *Información de Vuelo* (*Vuelo*), y *Horario de Vuelo* (*Vuelo*)
- ?? Clases que son operaciones: *Consulta*, *Pago*, *Reserva*.
- ?? Clases de interfaces de usuario: *Mensaje de Bienvenida*, *Pantalla Principal*.
- ?? Clases del sistema completo: *Sistema de Reservas*.
- ?? Clases actores: *Cliente*.

La Tabla 6.2 muestra las modificaciones a las clases candidatas originales.

| Clases Candidatas | Modificación |
|------------------------------|------------------------------------|
| Sistema de Reservas de Vuelo | Eliminada (sistema completo) |
| Sistema | Eliminada (imprecisa) |
| Usuario | Eliminada (actor) |
| Consulta | Eliminada (operación) |
| Reserva | Eliminada (operación) |
| Vuelo | |
| Boleto Aéreo | Eliminada (irrelevante) |
| Agente de Viajes Humano | Eliminada (irrelevante) |
| Sistema de Reservas | Eliminada (sistema completo) |
| Internet | Eliminada (implementación) |
| World Wide Web | Eliminada (implementación) |
| Hoja Principal | Eliminada (interface) |
| Mensaje de Bienvenida | Eliminada (interface) |
| Servicios | Eliminada (imprecisa) |
| Opción | Eliminada (imprecisa) |
| Reservaciones de Vuelos | Renombrada: Reservación |
| Acceso | Eliminada (imprecisa) |
| Login | Eliminada (atributo) |
| Email | Eliminada (atributo) |
| Password | Eliminada (atributo) |
| Registro | Renombrada: RegistroUsuario |
| Actividad | Eliminada (imprecisa) |
| Consulta de Vuelos | Eliminada (operación) |
| Reserva de Vuelos | Eliminada (operación) |
| Pago de Boletos | Eliminada (operación) |
| Horario de Vuelos | Eliminada (duplicada con Horario) |
| Tarifa de Vuelos | Eliminada (duplicada con Tarifa) |
| Información de Vuelo | Eliminada (atributo) |
| Horario | |
| Aerolínea | |
| Ciudad | Renombrada: Aeropuerto |
| Tarifa | |
| Costo | Eliminada (redundante) |
| Estado | Eliminada (imprecisa) |
| Información | Eliminada (imprecisa) |
| Asiento | |
| Día | Eliminada (atributo) |
| Hora | Eliminada (atributo) |
| Preferencia | Eliminada (imprecisa) |
| Búsqueda | Eliminada (operación) |
| Fecha | Eliminada (atributo) |
| Categoría de Asiento | Eliminada (atributo) |
| Vuelo Directo | Eliminada (atributo) |
| Cliente | Eliminada (redundante y actor) |
| Itinerario | Eliminada (imprecisa) |
| Pasajero | |
| Compra | Eliminada (operación) |
| Tarjeta de Crédito | Renombrada: RegistroTarjeta |
| Boleto | Eliminada (irrelevante) |
| Mostrador del Aeropuerto | Eliminada (irrelevante) |
| Número de Tarjeta de Crédito | Eliminada (atributo) |

Tabla 6.2. Clases Candidatas para el sistema de reservas de vuelo identificadas de la descripción del problema.

Las clases identificadas se muestran en la Tabla 6.3. Nótese que se agregaron dos nuevas clases, Avión y ViajeroFrecuente, que no aparecían en la descripción del problema. Esto se hizo para lograr un dominio más

completo. En general, distintos analistas identificarán listas similares de clases, aunque siempre con alguna variación.

| Clases Identificadas | |
|----------------------|------------------|
| Vuelo | Tarifa |
| Reservación | Asiento |
| RegistroUsuario | Pasajero |
| Horario | RegistroTarjeta |
| Aerolínea | Avión |
| Aeropuerto | ViajeroFrecuente |

Tabla 6.3 Clases identificadas para el sistema de reservaciones de vuelo.

Diagrama de Clases

Después de haber identificado y seleccionado las clases, se debe construir el diagrama de clases para el dominio del problema. Este diagrama se muestra en la Figura 6.32 y puede ayudar a identificar clases adicionales, y servirá de base para encontrar las atributos y asociaciones entre ellas.

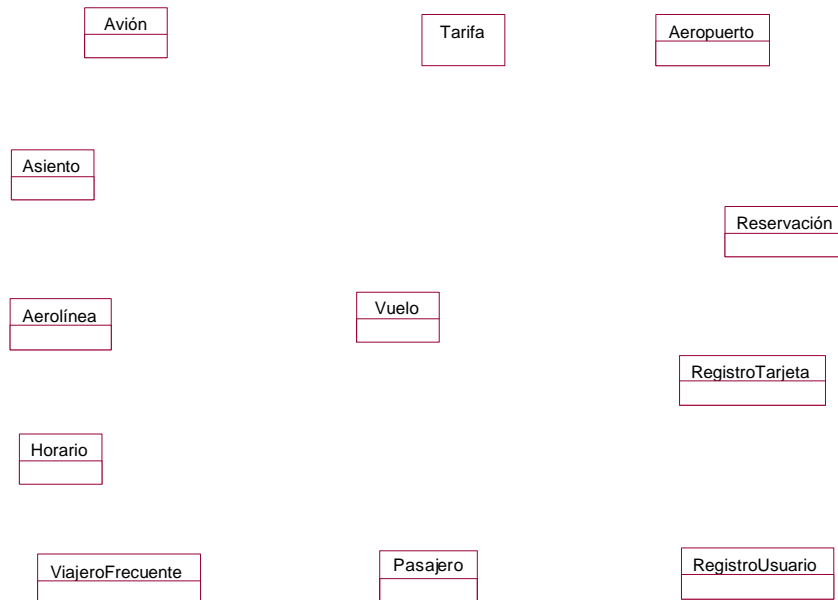


Figura 6.32. Diagrama de clases identificadas para el sistema de reservaciones de vuelo.

Aunque se puede parar aquí el proceso de desarrollo del dominio del problema, continuaremos con la identificación de asociaciones y atributos para el sistema de reservaciones de vuelo.

Identificación de Asociaciones

El proceso de identificación de asociaciones es bastante similar al de identificación de clases, sólo que en lugar de sustantivos buscamos frases que relacionen sustantivos correspondientes a clases ya identificadas. Lamentablemente, hasta allí llega la similitud, que se vuelve bastante difícil esta identificación por lo restringido de la descripción del problema. El documento de casos de uso tiende a ser mucho más importante para la identificación de estas frases. Dado que en nuestra metodología las asociaciones y operaciones del sistema serán identificadas durante el modelo de diseño, aquí nos limitaremos a dar un pequeño ejemplo de la identificación de asociaciones en base a nuestro conocimiento del dominio del problema. (En cierta manera escogimos como ejemplo el desarrollo de un sistema de reservaciones de vuelos por ser un dominio al cual la mayoría de los lectores podrán fácilmente relacionarse.)

Diagrama de Clases con Asociaciones

En lugar de tomar como punto de partida la descripción del problema o los documentos de casos de uso, simplemente identificamos nuestras propias frases correspondientes al dominio del problema del sistema de reservaciones, como se muestran en la Tabla 6.4. Este proceso de identificación es sencillo cuando el problema es muy limitado y el dominio es fácil de analizar. De lo contrario se requiere un proceso de identificación mucho más extenso como veremos en la etapa de diseño.

| Asociaciones Identificadas |
|---|
| Un vuelo contiene reservaciones |
| Un vuelo se dirige a un aeropuerto |
| Un vuelo contiene tarifas |
| Un vuelo se efectúa en un avión |
| Un vuelo contiene asientos |
| Un vuelo pertenece a una aerolínea |
| Un vuelo tiene un horario |
| Un pasajero puede acumular millas como viajero frecuente |
| Un pasajero efectúa reservaciones |
| Una reservación requiere de un registro de tarjeta de crédito |
| Un registro de tarjeta pertenece a un registro de usuario |

Tabla 6.4 Asociaciones identificadas para relacionar clases en el dominio del problema.

Después de haber identificado las asociaciones, se debe construir una versión del diagrama de clases que incluya estas asociaciones, como se muestra en la Figura 6.33.

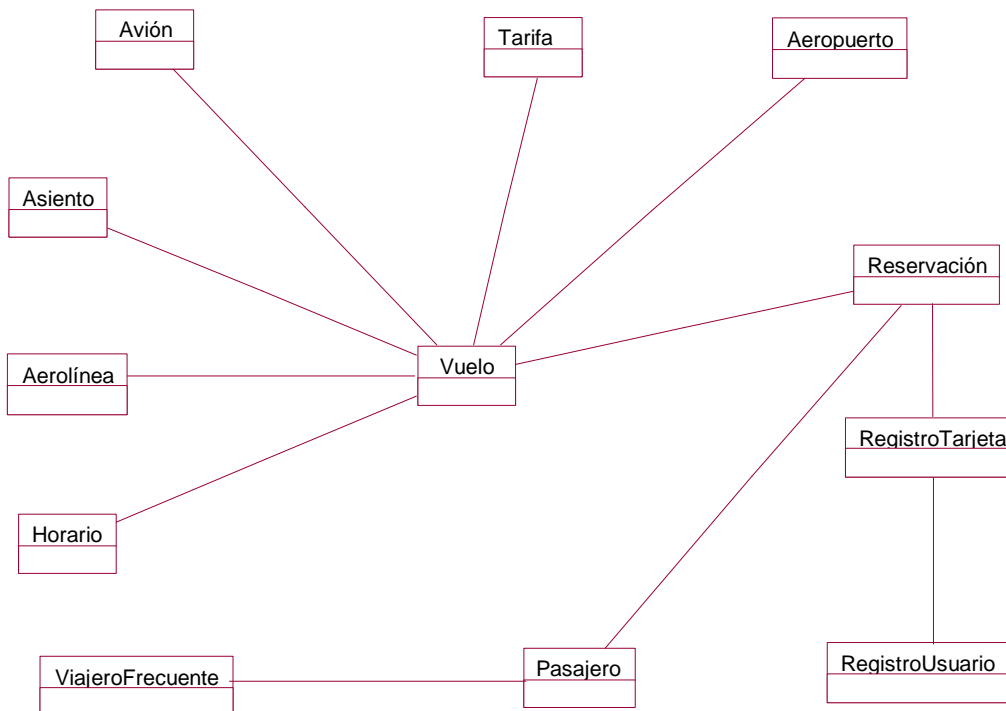


Figura 6.33. Diagrama de clases con asociaciones entre clases identificadas. Se omiten los nombres de las asociaciones.

Diagrama de Clases con Roles

Después de haber hecho el diagrama de clases con asociaciones, se puede construir una versión del diagrama de clases incluyendo roles. Como se explicó en el capítulo 4, el nombre del rol describe el papel que juega una clase en la asociación desde el punto de vista de la otra clase. Si hay una sola asociación entre un par de clases, y el nombre de la clase describe adecuadamente su rol, se puede omitir los nombres de rol.

Para tal propósito, modificamos levemente las asociaciones identificadas anteriormente, como se muestran en la Tabla 6.5.

| Asociaciones Identificadas con Roles |
|---|
| Un vuelo contiene reservaciones |
| Un vuelo tiene un aeropuerto de destino |
| Un vuelo tiene un aeropuerto de origen |
| Un vuelo puede hacer en escalas en otros aeropuertos |
| Un vuelo contiene tarifas de ida y vuelta |
| Un vuelo contiene tarifas solamente de ida |
| Un vuelo se efectúa en un avión |
| Un vuelo contiene asientos |
| Un vuelo pertenece a una aerolínea |
| Un vuelo tiene un horario de llegada |
| Un vuelo tiene un horario de salida |
| Un pasajero puede acumular millas como viajero frecuente |
| Un pasajero efectúa reservaciones |
| Una reservación requiere de un registro de tarjeta de crédito |
| Un registro de tarjeta pertenece a un registro de usuario |
| Un vuelo puede tener múltiples conexiones |

Tabla 6.5 Asociaciones identificadas con roles para relacionar clases en el dominio del problema.

Se extiende el diagrama de clases con asociaciones para incluir roles, como se muestra en la Figura 6.34.

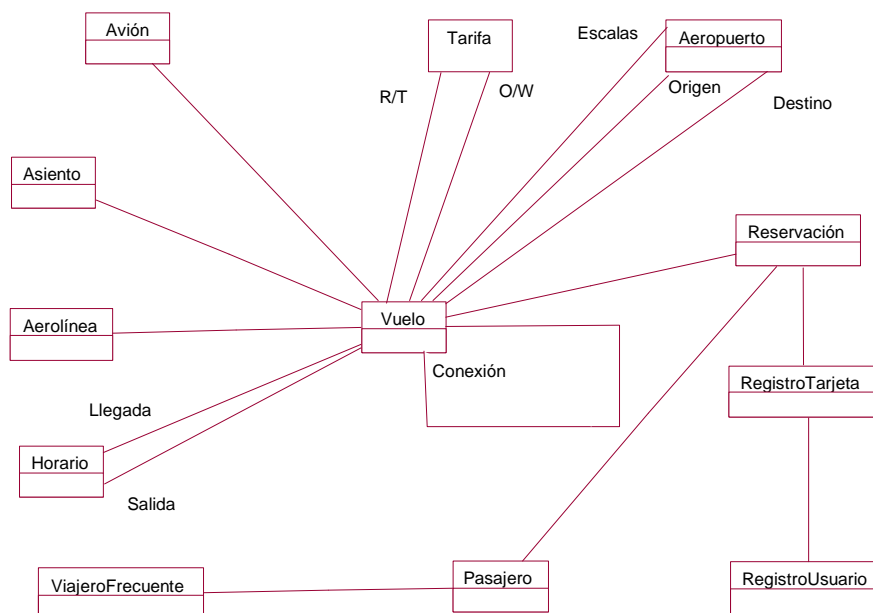


Figura 6.34 Diagrama de clases con asociaciones y roles entre clases identificadas. Se omiten los nombres de las asociaciones. (O/W representa “one way” o “solamente ida”, mientras que R/T representa “round trip” o “viaje redondo”.)

Diagrama de Clases con Multiplicidad

Después de haber hecho el diagrama de clases con asociaciones y roles, se puede construir una versión del diagrama de clases incluyendo multiplicidad. Se determina la multiplicidad para cada asociación.

Para tal propósito, modificamos levemente las asociaciones identificadas anteriormente, como se muestran en la Tabla 6.6. Nótese que la multiplicidad, al igual que las relaciones entre las clases, son bastante subjetivas pudiendo variar entre analistas. Dentro de esa subjetividad todas deben transmitir un dominio del problema similar.

| Asociaciones Identificadas con Roles y Multiplicidad |
|--|
| Un vuelo contiene múltiples reservaciones |
| Una reservación se relaciona con múltiples vuelos |
| Un vuelo tiene un aeropuerto de destino |
| Un vuelo tiene un aeropuerto de origen |
| Un vuelo puede hacer escalas en múltiples aeropuertos |
| Un vuelo contiene múltiples tarifas de ida y vuelta |
| Un vuelo contiene múltiples tarifas solamente de ida |
| Un vuelo se efectúa en múltiples aviones (dependiendo del día) |
| Múltiples vuelos se efectúan en un mismo avión (en diferente momento) |
| Un vuelo contiene múltiples asientos |
| Un vuelo pertenece a una aerolínea |
| Un vuelo tiene múltiples horarios de llegada (correspondiendo a diferentes destinos) |
| Un vuelo tiene múltiples horarios de salida (correspondiendo a diferentes destinos) |
| Un pasajero puede acumular millas en múltiples cuentas de viajero frecuente |
| Un pasajero efectúa múltiples reservaciones |
| Múltiples pasajeros pueden pertenecer a una misma reservación |
| Múltiples reservaciones pueden requerir de un mismo registro de tarjeta de crédito |
| Un registro de tarjeta pertenece a un registro de usuario |
| Un vuelo puede tener múltiples conexiones |

Tabla 6.6 Asociaciones identificadas con roles y multiplicidad para relacionar clases en el dominio del problema.

Se extiende el diagrama de clases con asociaciones y roles, para incluir multiplicidad, como se muestra en la Figura 6.35.

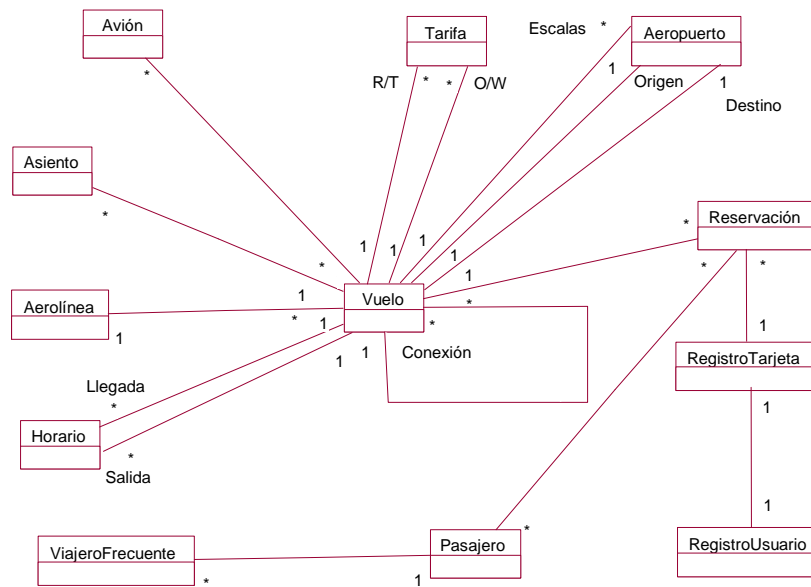


Figura 6.35 Diagrama de clases con asociaciones, roles y multiplicidad entre clases identificadas. Se omiten los nombres de las asociaciones.

Identificación de Atributos

De manera similar a asociaciones se puede determinar los atributos del dominio del problema. Este proceso tiene una complejidad similar al de identificación de las asociaciones. Sin embargo, identificar atributos puede resultar hasta más difícil de lograrlo a través de un proceso de búsqueda a partir de la descripción del problema e incluso del documento de casos de uso. En lugar de esto, simplemente identificamos nuestros propios atributos para las distintas clases identificadas para el dominio del problema del sistema de reservaciones, como se muestran en la Tabla 6.7.

| Clases | Atributos |
|------------------|---|
| Vuelo | Número |
| Reservación | Clave |
| RegistroUsuario | Nombre, Dirección, Colonia, Ciudad, País, Código Postal, Teléfono Casa, Teléfono Oficina, Fax, Email, Login, Password |
| Horario | Día, Hora |
| Aerolínea | Nombre |
| Aeropuerto | Nombre, Ciudad, País |
| Tarifa | Clase, Precio, Impuestos |
| Asiento | Fila, Letra |
| Pasajero | Nombre |
| RegistroTarjeta | Nombre, Número, Expedidor, Vencimiento |
| Avión | Fabricante, Modelo |
| ViajeroFrecuente | Número, Aerolínea |

Tabla 6.7 Atributos identificados para las clases identificadas en el sistema de reservaciones de vuelo.

Nuevamente, este proceso de identificación es sencillo cuando el problema es muy limitado y el dominio es fácil de analizar. De lo contrario se requiere un proceso de identificación mucho más extenso como veremos en la etapa de diseño. No es necesario listar todos los atributos, solamente los atributos más relevantes, omitiendo atributos menores.

Diagrama de Clases con Atributos

Se extiende el diagrama de clases con asociaciones, roles y multiplicidad, para incluir los atributos principales, como se muestra en la Figura 6.36.

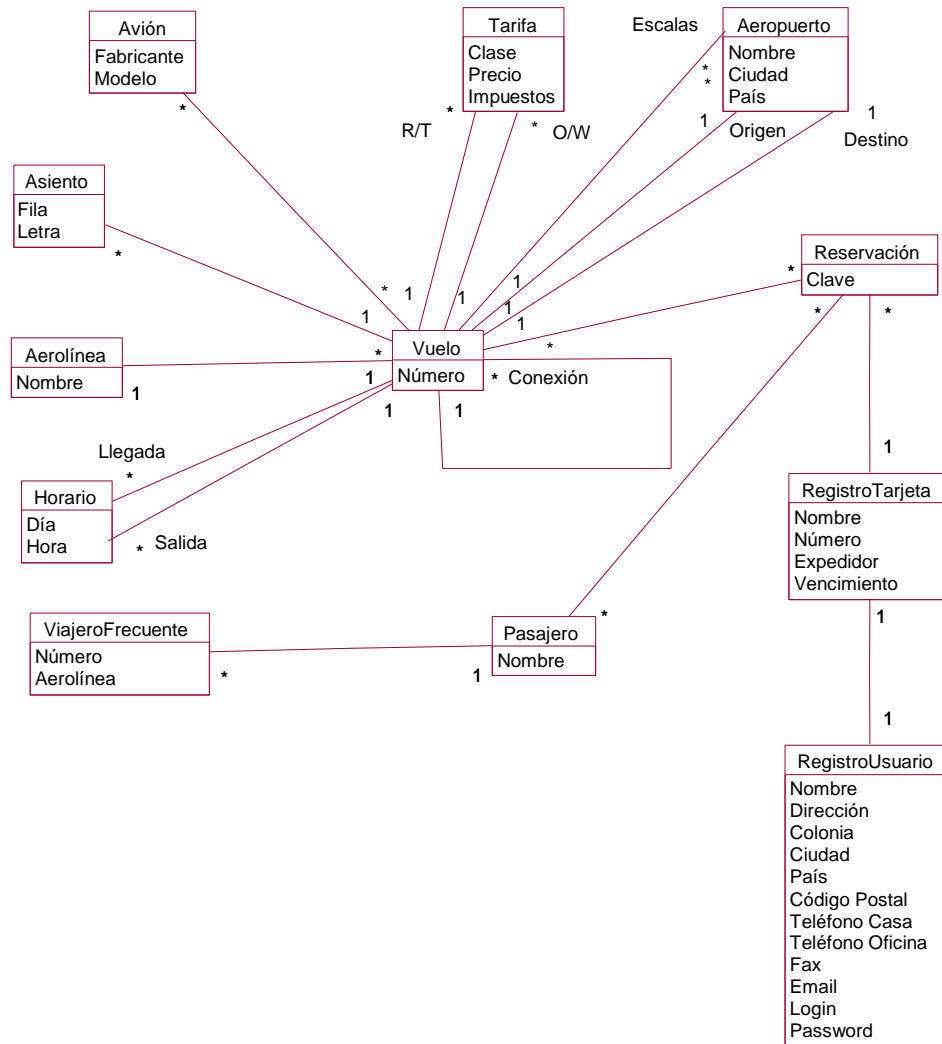


Figura 6.36 Diagrama de clases con asociaciones, roles, multiplicidad y atributos para clases identificadas. Se omiten los nombres de las asociaciones y sólo se muestran los atributos más relevantes.

Diccionario de Clases

El *diccionario de clases* o *diccionario de datos* describe textualmente las clases identificadas durante el modelo del dominio del problema. Este diccionario sirve como un glosario de términos y se muestra a continuación.

- ?? **Vuelo** - Se denomina por medio de un número. El vuelo tiene como origen un aeropuerto en una ciudad y tiene como destino un aeropuerto de otra ciudad. Un vuelo puede tener múltiples escalas y múltiples vuelos se relacionan por medio de conexiones. El vuelo pertenece a una aerolínea y puede operar varios días a la semana teniendo un horario de salida y otro de llegada.
- ?? **Reservación** - Para poder tomar un vuelo es necesario contar con una reservación previa, la cual debe pagarse antes de una fecha límite, que puede ser el propio día del vuelo. Una reservación puede hacerse para múltiples vuelos y múltiples pasajeros. La reservación cuenta con una clave identificando un récord de reservación particular.
- ?? **RegistroUsuario** - Para poder utilizar el sistema de reservaciones, el usuario debe estar registrado con el sistema. El registro contiene información acerca del usuario que incluye nombre, dirección, colonia, ciudad, país, código postal, teléfono de casa, teléfono de oficina, fax, email, login y password.
- ?? **Horario** - El horario de un vuelo se determina por su hora de salida y hora de llegada durante los días que opera.
- ?? **Aerolínea** - La aerolínea provee servicio de múltiples vuelos entre diferentes ciudades bajo diferentes horarios. La aerolínea se identifica por un nombre.

- ?? **Aeropuerto** - El aeropuerto sirve como origen, destino y escalas de un vuelo. El aeropuerto se encuentra en una ciudad de un país determinado.
- ?? **Tarifa** - Los diferentes vuelos tienen múltiples tarifas para compra de boleto, variando según la clase de boleto, si son de ida o de ida y vuelta, y dependiendo de las diversas restricciones y ofertas existentes.
- ?? **Asiento** - Una reservación de vuelo puede incluir la asignación de asiento, especificada mediante una fila y un número. El número de asientos disponibles en un vuelo particular dependen del tipo de avión que opere ese día.
- ?? **Pasajero** - Para poder hacer una reservación se requiere dar el nombre del pasajero. Varios pasajeros pueden aparecer bajo una sola reservación.
- ?? **RegistroTarjeta** - Para poder hacer un pago con una tarjeta de crédito, se debe tener un registro de tarjeta. El registro contiene información acerca de la tarjeta incluyendo nombre, número, expedidor y vencimiento. LA tarjeta está ligada a un registro de usuario.
- ?? **Avión** - Un vuelo en una fecha determinada se hace en un tipo de avión particular. El tipo de avión define la cantidad máxima de pasajeros que pueden viajar en ese vuelo para esa fecha.
- ?? **ViajeroFrecuente** - El pasajero tiene la opción de acumular millas para un vuelo particular si cuenta con una tarjeta de viajero frecuente para la aerolínea correspondiente.

6.6 Dominio del Problema para el Sistema de Reservaciones de Vuelos

El modelo del dominio del problema puede hacerse bastante complejo en el caso de sistema de gran tamaño, para lo cual es necesario separar las clases en *módulos*. De tal manera, el modelo completo se dividiría en una colección de módulos, donde cada módulo es una agrupación lógica de clases y sus asociaciones correspondientes. Para el sistema de reservaciones de vuelo podemos identificar dos módulos principales para el dominio del problema de acuerdo a la relación lógica entre las clases. Estos módulos son *Registro*, conteniendo las clases que guardan información sobre el usuario del sistema y *Servicios* conteniendo las clases que guardan información sobre los vuelos, pasajeros y reservaciones. En otras palabras, las clases para el módulo de registro se relacionan con la utilización del sistema ligados al actor *Base de Datos de Registro*, mientras que las clases para el módulo de servicios se relacionan con el propio sistema de reservaciones ligados al actor *Base de Datos de Reservaciones*. La razón de separar en dos módulos va muy de la mano con la existencia de estos dos actores secundarios, ya que al corresponder cada actor secundario a una base de datos, los módulos afianzan esta correspondencia. Sin embargo, esto no tiene por qué ser realmente así, pudiendo existir un sólo módulo para una sistema con múltiples actores secundarios o incluso múltiples módulos por cada actor secundario. A continuación describimos los módulos para el sistema de reservaciones de vuelo.

Servicios

En la Figura 6.37 se muestra las clases pertenecientes al módulo de *Servicios* del sistema de reservaciones.

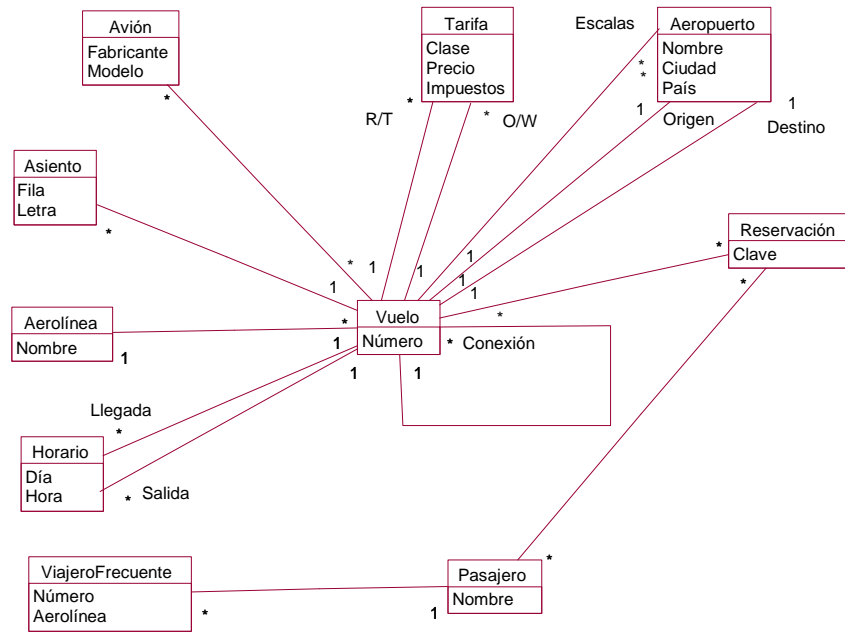


Figura 6.37 Diagrama de clases para el módulo de *Servicios* del sistema de reservaciones de vuelo.

Registro

En la Figura 6.38 se muestra las clases pertenecientes al módulo de *Registro* del sistema de reservaciones.



Figura 6.38 Diagrama de clases para el módulo de *Registro* del sistema de reservaciones de vuelo.

