

# **SAD del Subsistema de Reservas del Sistema de Gestión Hotelera**

**Daniel Perovich - Andrés Vignaga**

{perovich, avignaga}@fing.edu.uy

Grupo COAL  
Instituto de Computación  
Facultad de Ingeniería  
Universidad de la República  
Montevideo, Uruguay

# 1 Introducción

---

El Sistema de Gestión Hotelera es el caso de estudio del grupo de investigación COAL, del Instituto de Computación de Facultad de Ingeniería, de la Universidad de la República, Uruguay. Fue elegido como tal por ser una aplicación de porte empresarial de fácil entendimiento. Este sistema está conformado por varios subsistemas; entre ellos se encuentra el Subsistema de Reservas del cual este documento presenta la descripción de su arquitectura. El Subsistema de Reservas del Sistema de Gestión Hotelera fue presentado en [CD01], trabajo que fue discutido y profundizado dentro del grupo de investigación.

El presente documento provee una vista de alto nivel de la arquitectura del Subsistema de Reservas del Sistema de Gestión Hotelera, los objetivos y restricciones, los casos de uso significativos, los patrones de arquitectura aplicados y las principales decisiones de diseño. Este documento da una vista general del resto de los artefactos generados en el proceso de desarrollo.

## 1.1 Propósito

Este documento de arquitectura de software (por sus siglas en inglés, SAD) tiene como propósito brindar una visión comprensible de la arquitectura general del Subsistema de Reservas del Sistema de Gestión Hotelera, utilizando diferentes vistas de la arquitectura para ilustrar diferentes aspectos del sistema. Captura las decisiones más importantes en lo que respecta a la arquitectura del sistema que fueron tomadas en el proyecto.

Este SAD está dirigido a distintos tipos de actores: estudiantes, docentes, investigadores y desarrolladores.

Un estudiante de ingeniería de software puede utilizar este documento como base para la documentación del desarrollo de productos de software en proyectos de diferente porte.

Un docente de la misma área puede tomar este documento como base para mostrar la importancia de la arquitectura en el desarrollo de software así como el rol del arquitecto de software. Asimismo puede ser utilizado en talleres de desarrollo de software dado que el caso de estudio refiere únicamente a un subsistema del Sistema de Gestión Hotelera, y su tamaño puede adaptarse a cursos de diferente duración. A su vez, la tecnología a aplicar así como el paradigma de desarrollo puede ser adaptado a otras realidades. En el marco de la Facultad de Ingeniería se ha utilizado el caso de estudio tanto para el desarrollo orientado a objetos como basado en componentes, sin y con manejo de persistencia. Ha sido implementado, incluso, tanto en la plataforma .NET como en la plataforma J2EE [BFF03].

Un investigador puede basarse en lo presentado en este documento y así evitar dedicar esfuerzo en sus trabajos en comentar el caso de estudio. Además, muchas líneas de investigación en el grupo COAL están llevándose adelante partiendo de este caso de estudio, así como de sistemas similares, por lo que otros grupos pueden encontrar nuevas vetas en las cuales profundizar. Es de nuestro interés intercambiar con ellos ideas y resultados.

Desde el punto de vista de un desarrollador, este documento le brindará, con certeza, una buena razón para darle a la arquitectura de software la importancia que tiene en todo proyecto de desarrollo.

## 1.2 Alcance

Este SAD profundiza principalmente en las vistas de casos de uso y lógica, incluyendo algunos elementos significativos de las otras vistas.

El Subsistema de Reservas consta de seis casos de uso principales, siendo dos de ellos procesos batch. Se atacará principalmente aquellos casos de uso que involucran interacción con los actores.

Finalmente, este documento es la descripción de arquitectura del caso de estudio, no es un instructivo de cómo elaborar un SAD; en otras palabras el lector no encontrará aquí comentarios sobre qué tipo de información debe ser incluida en el documento ni cuáles son los criterios a utilizar en casos generales. Puede dirigirse a [RUP03] y [YOO03] para ahondar en dicho aspecto.

### 1.3 Referencias

- [BFF03] Implementación de un sistema de información basado en componentes con J2EE. S. Bengochea, F. Fajardo, J. Ferré. Reporte técnico 03-17, InCo-PEDECIBA, 2003.
- [CD01] UML Components. J. Cheesman, J. Daniels. Addison Wesley, 2001.
- [GHJ+95] Design Patterns. E. Gamma, et al. Addison Wesley, 1995.
- [Kru95] Architectural Blueprints—The “4+1” View Model of Software Architecture, P. Kruchten, Rational Software Corp. IEEE Software 12 (6), November 1995.
- [Lar02] Applying UML and Patterns. C. Larman. Prentice-Hall, 2002.
- [RUP03] IBM Rational Unified Process. <http://www.rational.com/rup>, 2003.
- [UML03] Unified Modeling Language. OMG. <http://www.omg.org/uml>, 2003.
- [YOO03] Yoopeedoo (UPEDU). <http://www.yoopeedoo.org>, 2003,

### 1.4 Organización

El documento esta organizado en capítulos, correspondiendo cada uno de ellos a los sugeridos en la plantilla provista en [RUP03].

El capítulo 2 presenta la representación de la arquitectura para el caso de estudio, esto es, la descripción de las vistas necesarias y el framework arquitectónico utilizado. Los capítulos 3 al 10 presentan la descripción de las vistas indicadas en el capítulo 2.

# 2 Representación de la Arquitectura

---

El Subsistema de Reservas del Sistema de Gestión Hotelera es una aplicación de porte empresarial desarrollada como caso de estudio dentro del grupo de investigación. Es un sistema de información en el dominio Hotelaría, y concierne principalmente a la gestión de reservas de una cadena hotelera.

La característica de ser un sistema de información hace que la vista de casos de uso y la vista lógica sean las más relevantes y por ello serán las más extensas.

La arquitectura está representada por diferentes vistas utilizando notación UML [UML03] de forma que permitan visualizar, entender y razonar sobre los elementos significativos de la arquitectura e identificar las áreas de riesgo que requieren mayor detalle de elaboración. Este documento es una forma de comunicar el modelo del subsistema, presentando la información y discusiones estructuradamente.

La arquitectura del subsistema se descompone en las siguientes dimensiones:

- **Requerimientos:** Requerimientos funcionales y no-funcionales del sistema.
- **Elaboración:** Representación lógica del sistema y representación de tiempo de ejecución.
- **Implementación:** Vista de módulos implementados, potenciales escenarios de infraestructura y el deployment de los módulos.

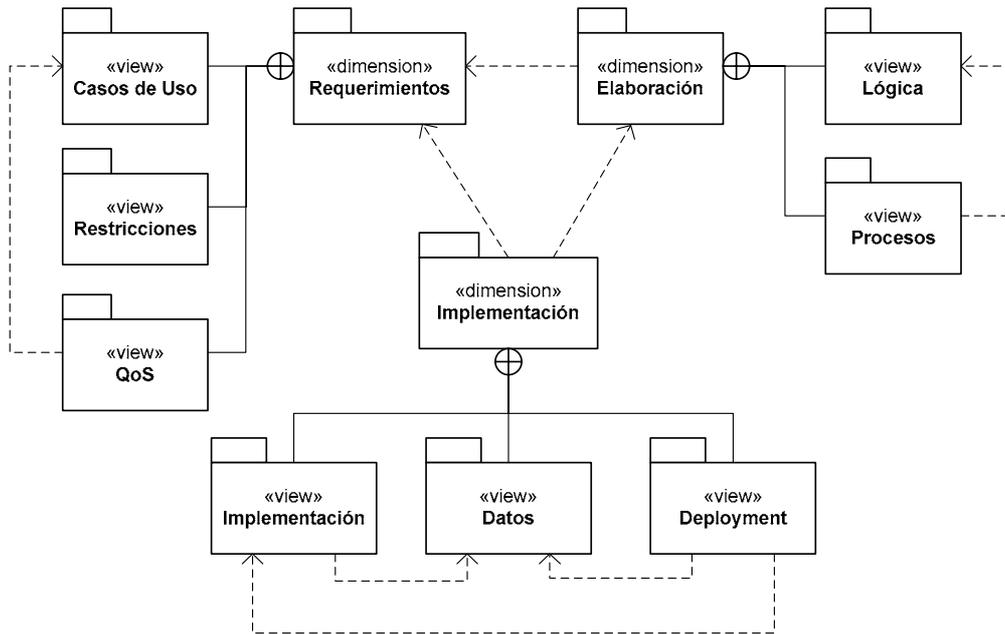
La siguiente sección detalla las vistas de la arquitectura que serán utilizadas para cubrir las dimensiones mencionadas. La sección 2.2 presenta el framework arquitectónico utilizado.

## 2.1 Representación

La arquitectura del Subsistema de Reservas está representada siguiendo las recomendaciones de [RUP03]. Las vistas necesarias para especificar dicho subsistema se enumeran a continuación:

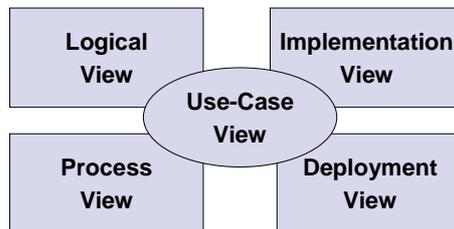
- **Vista de Casos de Uso:** Describe el proceso de negocio más significativo y el modelo del dominio. Presenta los actores y los casos de uso para el sistema.
- **Vista de Restricciones:** Describe restricciones tecnológicas, normativas, uso de estándares, entre otros, las cuales deben ser respetadas tanto por el proceso de desarrollo como por el producto desarrollado.
- **Vista QoS:** Incluye aspectos de calidad, y describe los requerimientos no-funcionales del sistema.
- **Vista Lógica:** Describe la arquitectura del sistema presentando varios niveles de refinamiento. Indica los módulos lógicos principales, sus responsabilidades y dependencias.
- **Vista de Procesos:** Describe los procesos concurrentes del sistema.
- **Vista de Implementación:** Describe los componentes de deployment construidos y sus dependencias.
- **Vista de Datos:** Presenta los modelos de datos, los servicios de persistencia y los servicios de transaccionalidad utilizados.
- **Vista de Deployment:** Presenta aspectos físicos como topología, infraestructura informática, e instalación de ejecutables. Incluye además plataformas y software de base.

Las vistas presentadas forman en su conjunto una especificación completa del sistema, la cual se delinea en el siguiente diagrama. Las dependencias entre las vistas indican dependencias entre las vistas tanto a nivel de arquitectura como a nivel de diseño.



## 2.2 Framework Arquitectónico

La arquitectura sigue el framework “4+1” presentado en [Kru95]; este framework define cuatro vistas para la arquitectura (4) en conjunto con los escenarios (1), y es presentado en la siguiente figura.



El mapeo de las vistas utilizadas a las propuestas en el framework se presenta en la siguiente tabla.

Framework 4+1	Arquitectura
Use-Case View	Casos de Uso, Restricciones y QoS
Logical View	Lógica
Process View	Procesos
Implementation View	Implementación, Datos
Deployment View	Deployment

Los siguientes capítulos presentarán las vistas de la arquitectura del Subsistema de Reservas del Sistema de Gestión Hotelera indicadas anteriormente.

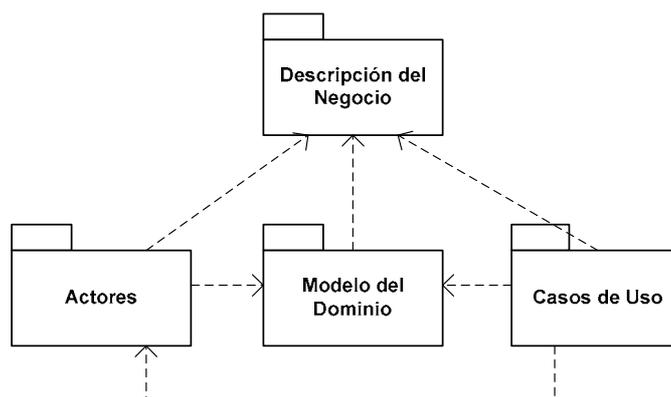
# 3 Vista de Casos de Uso

---

Esta vista presenta la percepción que tiene el usuario de las funcionalidades del sistema. Se presenta el proceso de negocio más importante y los casos de uso críticos que se derivan de éste. Este capítulo provee el contexto y determina el alcance del resto del documento.

Primeramente se describe el Negocio. Luego se presenta el modelo del dominio para el Subsistema de Reservas. Se identifican actores y se detallan los casos de uso significativos.

Este capítulo se organiza de la siguiente forma:



## 3.1 Descripción del Negocio

Primeramente se describe el Sistema de Gestión Hotelera, marco del Subsistema de Reservas. Luego se presenta una descripción de éste identificando los procesos de negocio críticos.

### Sistema de Gestión Hotelera

Una cadena hotelera desea automatizar los servicios brindados por sus hoteles. Cada hotel posee un sistema de información que satisface parcialmente los requerimientos informáticos reales de la empresa. Muchas actividades son registradas en formularios de papel y la obtención de datos estadísticos insume gran cantidad de recursos.

La gerencia general desea mantener en forma central y unificada todas las reservas que se hacen en sus hoteles. Como política de la empresa no se realiza overbooking, por lo que se quiere que dicha política sea ejecutada en todos los hoteles de la cadena. Se desea además poder sugerir a los clientes otros hoteles de la cadena cuando un hotel no tiene disponibilidad de la habitación solicitada. Es prioritario este requerimiento.

Los clientes de la empresa deben poder realizar todas sus actividades por Internet. Las estaciones de trabajo en los hoteles operarán con la misma interfaz de usuario; en cambio, en estos casos el hecho de encontrarse en un hotel determinado debe simplificar el uso del sistema. Debe proveerse además mecanismos para que las agencias de viajes interoperen con el sistema, por ejemplo mediante el uso de Web Services.

Hay fuertes restricciones de performance para los procesos de reserva, check-in y check-out.

Es importante, además, reutilizar un sistema de facturación existente. La empresa ha utilizado dicho producto en otras oportunidades y desea conservarlo y aprovecharlo en este emprendimiento.

Los empleados trabajan usualmente en el mismo hotel. Sin embargo es probable que los mismos sean rotados a otros hoteles en la región.

La gerencia general necesita información estadística. Ésta es utilizada para la apertura o clausura de hoteles en regiones donde la empresa está instalada. La información se recoge periódicamente y es analizada por economistas expertos de la empresa.

Por último, los servicios adicionales que brinda la empresa a los clientes varían según el hotel. Los mismos cubren una amplia gama de servicios como servicios a la habitación, paquetes turísticos, afiliación a sistemas de millas, etc. Estos servicios se irán incorporando y removiendo del sistema, incluso una vez que éste este en producción. El sistema debe ser capaz de incorporar nuevos módulos (subsistemas) que den soporte a nuevos servicios. Los servicios extras que se brinda a los clientes varían en el tiempo. De todas formas, el agregar o quitar un nuevo servicio no es un proceso en que el sistema propiamente participará. El sistema debe estar hecho de forma tal que simplifique la incorporación y remoción de los servicios extras brindados por la cadena hotelera.

## **Subsistema de Reservas**

El Subsistema de Reserva contempla tres de las actividades fundamentales del negocio, hacer una reserva, realizar un check-in y realizar un check-out. La empresa penalizará a aquellos clientes que no cancelen sus reservas, por lo que se les cobrará por dicho motivo.

La cadena hotelera es una empresa de gran dinamismo, donde nuevos hoteles son incorporados a la misma, e incluso algunos podrían ser vendidos y quitados del sistema. En cambio, no es común el realizar reformas edilicias, por lo que los detalles de cada hotel tienen muy baja frecuencia de cambios.

## **Procesos de Negocio**

Los siguientes procesos de negocio son relativos al Subsistema de Reservas:

- Gerenciamiento de la cadena hotelera (P1)

Este proceso involucra un conjunto de procesos simples encargados del gerenciamiento. Permite la incorporación de nuevos hoteles al sistema, así como la eliminación de los mismos. Se encarga además de la administración del personal de la cadena de hoteles.

- Reserva de Habitación (P2)

Este proceso administra todas las actividades de reserva por parte de los clientes. Involucra modificaciones y cancelaciones de reservas, así como la detección de aquellos clientes que no tomaron su reserva. La actividad de check-in está incluida en este proceso, siendo un camino al estado final del mismo.

- Check-out y Facturación (P3)

Este proceso cubre el check-out de los huéspedes, así como la facturación de los servicios contratados por ellos. La contratación de servicios por parte de los huéspedes no forma parte de este proceso.

- Consultas Estadísticas (P4)

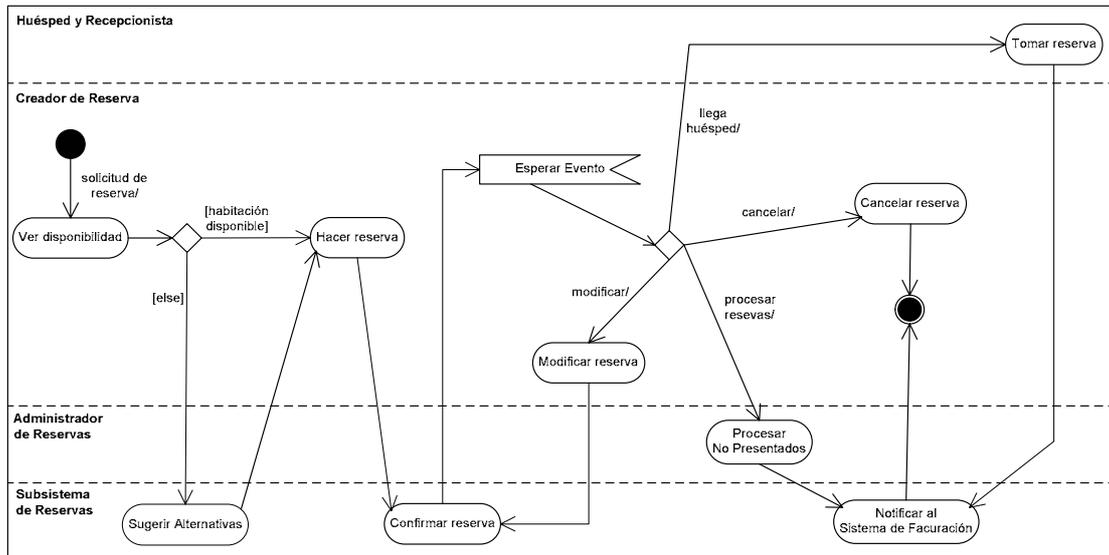
Este proceso ocurre cuando la gerencia general realiza un estudio de la situación de la cadena hotelera. Mediante este proceso se extraerá la información del sistema que sea útil para crear un DataWarehouse sobre el cual realizar variados tipos de estudios.

Los procesos (P1) y (P4), a pesar de estar relacionados con el Subsistema de Reservas, no son realmente parte de éste. Dichos procesos son más generales y pueden enmarcarse en otro subsistema del Sistema de Gestión Hotelera. Por lo tanto, el Subsistema de Reservas no realizará estos procesos de negocio.

Los procesos (P2) y (P3) conforman el corazón del Subsistema de Reservas. Estos procesos presentan exigencias de performance; en (P2) las reservas por Internet deben realizarse en menos de 5 segundos, una vez que el cliente llene su formulario. De igual manera la

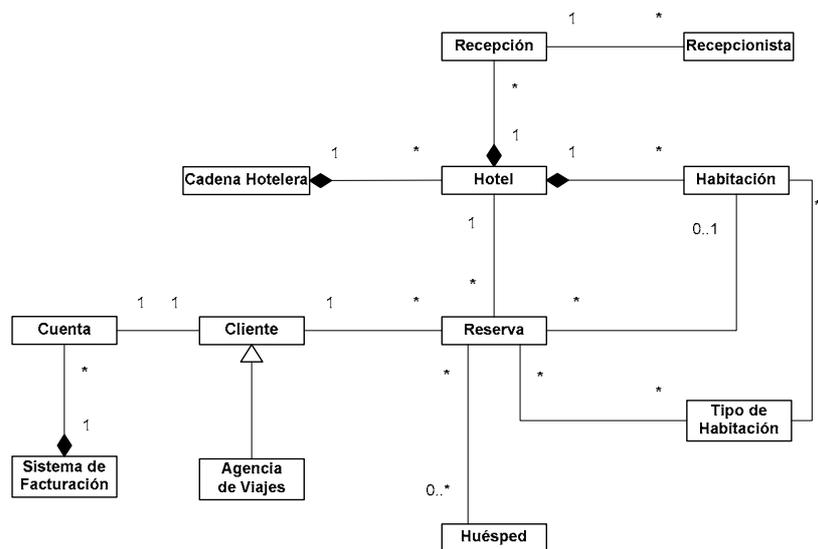
interoperabilidad con las agencias de viajes para realizar reservas tiene las mismas exigencias de tiempo de respuesta. El tiempo de realización de una reserva debe ser menor a 3 minutos cuando el cliente la realiza en la recepción del hotel o telefónicamente; este tiempo de respuesta incluye el llenado del formulario. Para ello, es necesario mantener toda la información posible de los clientes capturadas en visitas anteriores.

Ambos procesos tienen gran impacto en la arquitectura del sistema; en cambio, las repercusiones sobre ésta es similar en ambos casos. Por ello este documento se concentrará únicamente en el proceso Reserva de Habitación (P2). La siguiente figura presenta las actividades realizadas en este proceso detallando que actores las realizan.



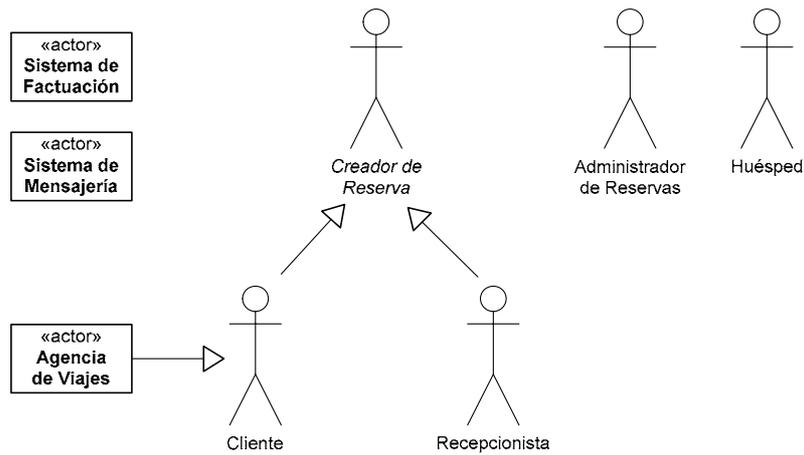
### 3.2 Modelo de Dominio

El modelo del dominio incluye aquel vocabulario del dominio significativo desde el punto de vista de la arquitectura, aquél que ayude al entendimiento de la misma.



### 3.3 Actores

Los siguientes actores son los que interactuarán con el Subsistema de Reservas una vez realizado el deploy.

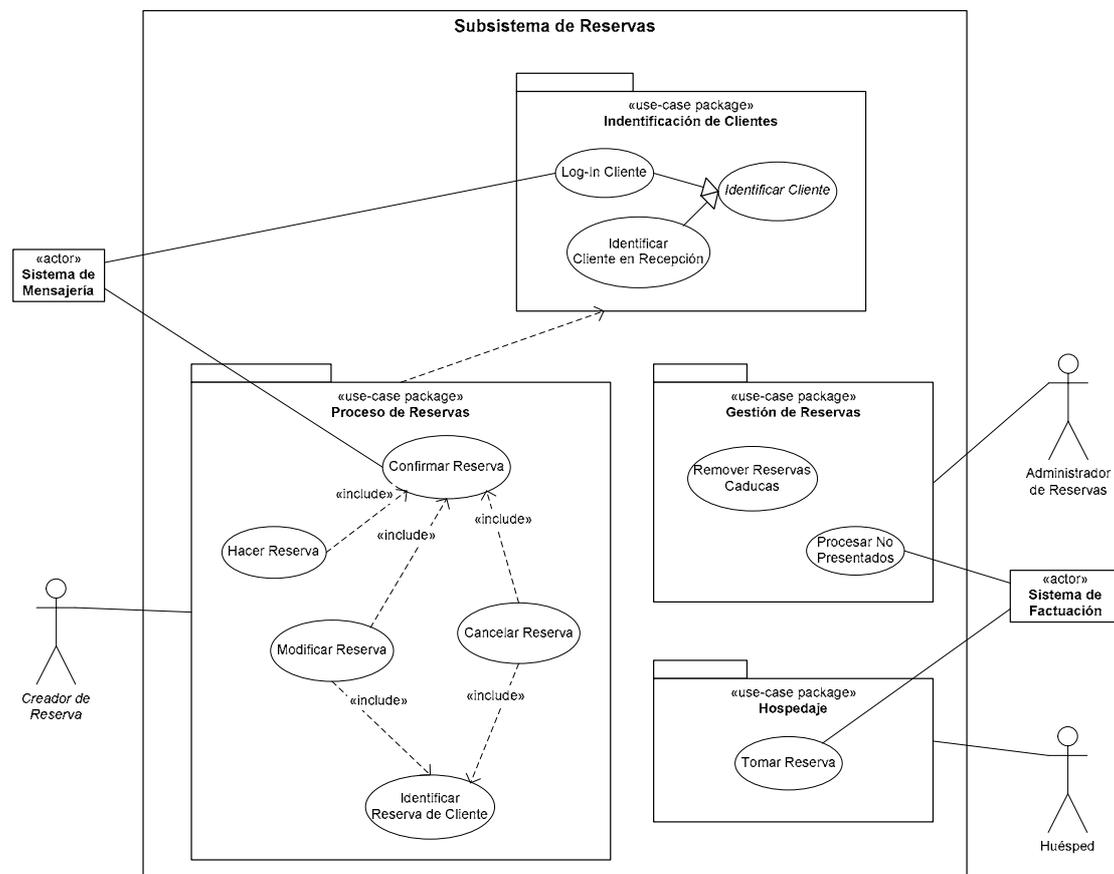


Siguiendo la notación propuesta en [Lar02], se utiliza la representación canónica para actores que representan a sistemas informáticos.

### 3.4 Casos de Uso

Los casos de uso críticos para el proceso (P2) se describen en esta sección. Primero se indica las relaciones entre los casos de usos detectados y luego se presenta la versión expandida de los mismos.

#### Modelo de Casos de Uso



## Hacer Reserva

Nombre	Hacer Reserva (CU1)
Actores	Creador de Reserva, Sistema de Mensajería
Actividades	Ver Disponibilidad, Sugerir Alternativas, Hacer Reserva, Confirmar Reserva
Sinopsis	Este caso de uso comienza cuando el Creador de Reserva solicita crear una reserva. El sistema chequea la disponibilidad de una habitación en un hotel solicitado. Si hay disponibilidad el Sistema hace la reserva y le confirma la misma al cliente. Si no hay disponible una habitación, el sistema sugiere hoteles alternativos.
Curso Típico de Eventos	
	<ol style="list-style-type: none"> <li>1. Incluir Identificar Cliente (CU8/CU9).</li> <li>2. Creador de Reserva indica hotel (en caso de estar en la Recepción de un hotel esta información se provee automáticamente), tipo de habitación y duración de la estadía.</li> <li>3. Sistema confirma disponibilidad.</li> <li>4. Sistema registra la reserva.</li> <li>5. Incluir Confirmar Reserva (CU10).</li> </ol>
Extensiones	
	<ol style="list-style-type: none"> <li>1a. No existe el cliente:             <ol style="list-style-type: none"> <li>1. Incluir Alta de Cliente.</li> <li>2. Resume 2.</li> </ol> </li> <li>3a. No hay disponibilidad:             <ol style="list-style-type: none"> <li>1. Sistema busca disponibilidad en otros hoteles.                 <ol style="list-style-type: none"> <li>1a. No hay disponibilidad en ningún hotel:                     <ol style="list-style-type: none"> <li>1. Sistema notifica a Creador de Reserva.</li> <li>2. Resume 2.</li> </ol> </li> <li>2. Creador de Reserva indica un hotel de su conveniencia.</li> </ol> </li> <li>2a. Creador de Reserva prefiere cambiar datos de la reserva:                     <ol style="list-style-type: none"> <li>1. Resume 2.</li> </ol> </li> </ol> </li> <li>3. Resume 4.</li> </ol>

## Modificar Reserva

Nombre	Modificar Reserva (CU2)
Actores	Creador de Reserva, Sistema de Mensajería
Actividades	Modificar Reserva, Confirmar Reserva
Sinopsis	El caso de uso comienza cuando Creador de Reserva solicita modificar los datos de la reserva. Se solicitan los nuevos datos y se verifica disponibilidad. En caso de éxito se registra los cambios y se confirma la reserva. En caso de fallo no se realiza ningún cambio en la reserva.
Curso Típico de Eventos	
	<ol style="list-style-type: none"> <li>1. Incluir Identificar Cliente (CU 8/CU9).</li> <li>2. Incluir Identificar Reserva de Cliente (CU7).</li> <li>3. Creador de Reserva modifica los datos de la reserva.</li> <li>4. Sistema verifica disponibilidad.</li> <li>5. Sistema registra la reserva.</li> <li>6. Incluir Confirmar Reserva (CU10).</li> </ol>
Extensiones	
	<p>3a. Creador de Reserva decide no modificar la reserva:</p> <ol style="list-style-type: none"> <li>1. Stop.</li> </ol> <p>4a. No hay disponibilidad:</p> <ol style="list-style-type: none"> <li>1. Sistema busca disponibilidad en otros hoteles. <ol style="list-style-type: none"> <li>1a. No hay disponibilidad en ningún hotel: <ol style="list-style-type: none"> <li>1. Sistema notifica a Creador de Reserva.</li> <li>2. Resume 3.</li> </ol> </li> <li>2. Creador de Reserva indica un hotel de su conveniencia. <ol style="list-style-type: none"> <li>2a. Creador de Reserva prefiere cambiar datos de la reserva: <ol style="list-style-type: none"> <li>1. Resume 3.</li> </ol> </li> </ol> </li> </ol> </li> <li>3. Resume 5.</li> </ol>

## Cancelar Reserva

Nombre	Cancelar Reserva (CU3)
Actores	Creador de Reserva, Sistema de Mensajería
Actividades	Cancelar Reserva, Sistema de Mensajería
Sinopsis	El caso de uso comienza cuando Creador de Reserva decide cancelar una reserva. El sistema elimina la reserva y notifica la cancelación.
Curso Típico de Eventos	
	<ol style="list-style-type: none"><li>1. Incluir Identificar Cliente (CU8/CU9).</li><li>2. Incluir Identificar Reserva de Cliente (CU7).</li><li>3. Creador de Reserva confirma cancelación.</li><li>4. Sistema marca la reserva como cancelada.</li><li>5. Incluir Confirmar Reserva (CU10)</li></ol>
Extensiones	
	<ol style="list-style-type: none"><li>3a. Creador de Reserva no confirma la cancelación:<ol style="list-style-type: none"><li>1. Fallo.</li></ol></li></ol>

## Tomar Reserva

Nombre	Tomar Reserva (CU4)
Actores	Huésped, Recepcionista, Sistema de Facturación
Actividades	Tomar Reserva, Notificar al Sistema de Facturación
Sinopsis	Este caso de uso comienza cuando Huésped llega al hotel. Indica la reserva que está a su nombre. El Huésped indica sus datos personales para registrarlos en la reserva. El Sistema le asigna una habitación y notifica al Sistema de Facturación que debe abrirse una cuenta para el cliente asociado a la reserva.
Curso Típico de Eventos	
	<ol style="list-style-type: none"> <li>1. Huésped llega al hotel e indica que desea tomar una reserva.</li> <li>2. Sistema muestra las reservas aún no tomadas del hotel para la fecha actual.</li> <li>3. Recepcionista elige una reserva en la lista.</li> <li>4. Huésped indica los datos personales.</li> <li>5. El Sistema le asigna una habitación.</li> <li>6. El Sistema notifica al Sistema de Facturación que una estadia ha dado comienzo.</li> </ol>
Extensiones	
	<p>1a/2a/3a.</p> <p>Huésped no tiene reserva/No hay reservas aún no tomadas/La reserva buscada no aparece en la lista:</p> <ol style="list-style-type: none"> <li>1. Incluir Hacer Reserva (CU1).</li> <li>2. Resume 4.</li> </ol> <p>4a. Huésped desea modificar los datos de la reserva:</p> <ol style="list-style-type: none"> <li>1. Recepcionista corrobora que el cliente asociado a la reserva permita al Huésped cambiar los datos de la misma.             <ol style="list-style-type: none"> <li>1a. El Huésped no tiene permitido cambiar los datos de la reserva:                 <ol style="list-style-type: none"> <li>1. Recepcionista informa el hecho al Huésped.</li> <li>2. Resume 4.</li> </ol> </li> </ol> </li> <li>2. Incluir Modificar Reserva (CU3).</li> <li>3. Resume 4.</li> </ol>

## Procesar No Presentados

Nombre	Procesar No Presentados (CU5)
Actores	Administrador de Reservas, Sistema de Mensajería, Sistema de Facturación
Actividades	Procesar No Presentados, Notificar al Sistema de Facturación
Sinopsis	El caso de uso comienza cuando el Administrador de Reservas decide procesar las reservas no tomadas. El sistema indica la cantidad de reservas no tomadas para el período indicado. El Administrador de Reservas confirma la acción y el Sistema notifica al Sistema de Facturación que debite el monto correspondiente a cada cliente y al Sistema de Mensajería que notifique el hecho al cliente.
Curso Típico de Eventos	
	<ol style="list-style-type: none"> <li>1. Administrador de Reservas indica el período a procesar.</li> <li>2. Sistema muestra las reservas a procesar.</li> <li>3. Administrador de Reservas confirma el proceso.</li> <li>4. Sistema notifica al Sistema de Facturación que debite a cada cliente el monto correspondiente.</li> <li>5. Sistema notifica al Sistema de Mensajería que envíe a cada cliente un aviso de reserva procesada como no tomada indicando el monto debitado.</li> </ol>
Extensiones	
	<ol style="list-style-type: none"> <li>1a. El período no corresponde al pasado:             <ol style="list-style-type: none"> <li>1. Sistema notifica el error.</li> <li>2. Resume 1.</li> </ol> </li> <li>3a. Administrador de Reservas no confirma el proceso:             <ol style="list-style-type: none"> <li>1. Fallo.</li> </ol> </li> </ol>

## Remover Reservas Caducas

Nombre	Remover Reservas Caducas (CU6)
Actores	Administrador de Reservas
Actividades	N/A
Sinopsis	El sistema debe mantener registro de las reservas realizadas por un tiempo determinado. Mediante este caso de uso el Administrador de Reservas le indica al sistema que reservas caducaron, i.e. no es necesario mantener registro, para que el sistema las elimine.
Curso Típico de Eventos	
	<ol style="list-style-type: none"><li>1. Administrador de Reservas indica la fecha a partir de la cual el sistema debe mantener registro.</li><li>2. Sistema calcula la cantidad de reservas a eliminar.</li><li>3. Sistema pide confirmación de eliminación.</li><li>4. Administrador de Reservas confirma eliminación.</li><li>5. Sistema elimina todas las reservas previas (estrictamente) a la fecha indicada.</li></ol>
Extensiones	
	<ol style="list-style-type: none"><li>2a. No hay reservas a eliminar:<ol style="list-style-type: none"><li>1. Fallo.</li></ol></li><li>4a. Administrador de Reservas no confirma eliminación:<ol style="list-style-type: none"><li>1. Fallo.</li></ol></li></ol>

## Casos de Uso de Solo Inclusión

### Identificar Reserva de Cliente

Nombre	Identificar Reserva de Cliente (CU7)
Actores	Creador de Reserva
Actividades	Identificar Reserva
Sinopsis	Identifica una reserva activa del cliente.
Curso Típico de Eventos	
	<ol style="list-style-type: none"><li>1. Sistema muestra las reservas activas (pendiente o en curso con check-in en el futuro) del cliente.</li><li>2. Creador de Reserva elige la reserva en la lista.</li><li>3. Sistema localiza la reserva.</li></ol>
Extensiones	
	<ol style="list-style-type: none"><li>1a. El cliente no tiene reservas:<ol style="list-style-type: none"><li>1. Fallo.</li></ol></li><li>2a. La reserva buscada no aparece en la lista:<ol style="list-style-type: none"><li>1. Fallo.</li></ol></li></ol>

### Identificar Cliente en Recepción

Nombre	Identificar Cliente en Recepción (CU8) / Identificar Cliente
Actores	Recepcionista
Actividades	N/A
Sinopsis	Localiza un cliente registrado.
Curso Típico de Eventos	
	<ol style="list-style-type: none"><li>1. Recepcionista provee los datos del cliente.</li><li>2. Sistema muestra los clientes que coinciden con los datos provistos.</li><li>3. Recepcionista elige el cliente en la lista.</li><li>4. Sistema localiza al cliente.</li></ol>
Extensiones	
	<ol style="list-style-type: none"><li>2a. Sistema no encuentra clientes que coincidan con los datos provistos:<ol style="list-style-type: none"><li>1. Sistema notifica el error.</li><li>2. Resume 1.</li></ol></li><li>3a. Ningún cliente corresponde al cliente buscado:<ol style="list-style-type: none"><li>1. Recepcionista informa el hecho al Sistema.</li><li>2. Resume 1.</li></ol></li></ol>

## Log-In Cliente

Nombre	Log-In Cliente (CU9) / Identificar Cliente
Actores	Cliente, Sistema de Mensajería
Actividades	N/A
Sinopsis	Identifica al actor como cliente registrado.
Curso Típico de Eventos	
	<ol style="list-style-type: none"><li>1. Cliente provee el nombre de usuario y el password.</li><li>2. Sistema localiza al cliente.</li><li>3. Sistema comprueba el password.</li></ol>
Extensiones	
	<ol style="list-style-type: none"><li>1a. Cliente no conoce el nombre de usuario ni el password:<ol style="list-style-type: none"><li>1. Fallo.</li></ol></li><li>1b. Cliente no conoce el password:<ol style="list-style-type: none"><li>1. Sistema localiza al cliente.</li><li>2. Sistema notifica al Sistema de Mensajería que envíe e-mail al cliente con el password.</li><li>3. Sistema notifica al Cliente que el password ha sido enviado por e-mail.</li><li>4. Resume 1.</li></ol></li><li>2a. Sistema no encuentra un cliente con el identificador indicado:<ol style="list-style-type: none"><li>1. Sistema notifica el error.</li><li>2. Resume 1.</li></ol></li><li>3a. El password ingresado es incorrecto:<ol style="list-style-type: none"><li>1. Sistema notifica el error.</li><li>2. Resume 1.</li></ol></li></ol>

## Confirmar Reserva

Nombre	Confirmar Reserva (CU10)
Actores	Sistema de Mensajería
Actividades	Confirmar Reserva
Sinopsis	Notifica al cliente cambios en una reserva. El mecanismo de comunicación puede ser e-mail, beeper, mensaje al celular o fax, en función de los datos que se tenga del cliente y el modo de comunicación elegido. Si el cliente es extranjero solo puede utilizarse e-mail.
Curso Típico de Eventos	
	<ol style="list-style-type: none"> <li>1. Sistema identifica el mecanismo de comunicación con el cliente.</li> <li>2. Sistema prepara información de la reserva.</li> <li>3. Sistema solicita al Sistema de Mensajería el envío del mensaje al cliente.</li> </ol>
Extensiones	
	N/A

## 3.5 Interfaz de Usuario

Esta sección presenta la captura de pantalla para algunos de los casos de uso presentados en la sección anterior.

### Hacer Reserva

#### Ingrese los datos de la reserva



Cliente: ID

Nombre

Usuario

e-mail

Teléfono

Fax

Ciudad:

Hotel:

Tipo de Habitación:

Check-in

Check-out

May	Junio de 2003					Jul	May	Junio de 2003					Jul
L	M	M	J	V	S	D	Lun	Mar	Mié	Jue	Vie	Sáb	Dom
26	27	28	29	30	31	1	26	27	28	29	30	31	1
2	3	4	5	6	7	8	2	3	4	5	6	7	8
9	10	11	12	13	14	15	9	10	11	12	13	14	15
16	17	18	19	20	21	22	16	17	18	19	20	21	22
23	24	25	26	27	28	29	23	24	25	26	27	28	29
30	1	2	3	4	5	6	30	1	2	3	4	5	6

Hacer Reserva

Cancelar

# Modificar Reserva

## Seleccione una reserva de la lista



Cliente: ID   
 Nombre   
 Usuario   
 e-mail   
 Teléfono   
 Fax

Reservas:

	ID	Hotel	TipoHab	Check In	Check Out	Estado	Hab
<a href="#">Seleccionar</a>	2	Centro	King	18/07/2003	20/07/2003	Pendiente	
<a href="#">Seleccionar</a>	5	Centro	Queen	16/07/2003	19/07/2003	EnCurso	
<a href="#">Seleccionar</a>	9	Punta del Este	Queen	22/07/2003	24/07/2003	Pendiente	
<a href="#">Seleccionar</a>	10	Punta del Este	Queen	27/07/2003	27/07/2003	Pendiente	
<a href="#">Seleccionar</a>	13	Punta del Este	Queen	30/07/2003	01/08/2003	Pendiente	

1

# Modificar Reserva

## Ingrese los nuevos datos de la reserva



Cliente: ID   
 Nombre   
 Usuario   
 e-mail   
 Teléfono   
 Fax

Ciudad:  ▼

Hotel:  ▼

Tipo de Habitación:  ▼

Check-in

Check-out

Check-in							Check-out						
Jun	Julio de 2003					Ago	Jun	Julio de 2003					Ago
L	M	M	J	V	S	D	Lun	Mar	Mié	Jue	Vie	Sáb	Dom
30	1	2	3	4	5	6	30	1	2	3	4	5	6
7	8	9	10	11	12	13	7	8	9	10	11	12	13
14	15	16	17	18	19	20	14	15	16	17	18	19	20
21	22	23	24	25	26	27	21	22	23	24	25	26	27
28	29	30	31	1	2	3	28	29	30	31	1	2	3
4	5	6	7	8	9	10	4	5	6	7	8	9	10

# 4 Vista de Restricciones

---

En esta vista se presentan las restricciones normativas, de estándares y de tecnológicas, a las cuales está sujeto tanto el proceso de desarrollo como el producto desarrollado, incluidas en las categorías soporte, implementación, interfaces y legalidad de FURPS+.

## 4.1 Normativas

Existen restricciones normativas, dictadas por organizaciones gubernamentales y no-gubernamentales, que determinan algunas decisiones del producto desarrollado.

### Licenciamiento

No existe regulación de licenciamiento para aplicaciones web en el país donde está radicada la cadena hotelera. El licenciamiento del producto pesará totalmente sobre la aplicación back-end. Por esta razón el producto no debe limitar la cantidad de usuarios simultáneos que permite la aplicación.

### Formas de pago

El país donde la cadena hotelera está instalada no permite el pago de servicios por Internet utilizando tarjetas de crédito. De esta forma no puede debitarse de tarjetas de crédito de los clientes los servicios brindados si no es en forma presencial. Por esta razón el mecanismo de pago no será controlado directamente por el Sistema de Gestión Hotelera.

### Registro Impositivo

Toda transacción comercial en el país de residencia de la cadena hotelera debe ser registrada y comunicada a la Dirección General Impositiva siguiendo los procedimientos y formatos provista por ésta. Existe un software que lleva adelante este trabajo y por lo tanto será utilizado directamente dentro del Sistema de Gestión Hotelera.

## 4.2 Estándares

### UML

Todo artefacto utilizado para comunicación y documentación, tanto entre miembros del equipo de desarrollo como con los clientes y usuarios, está basado en UML.

### Interfaz Web

La interfaz de usuario debe estar orientada al web. Debe ser posible visualizar el contenido utilizando cualquiera de los browsers más difundidos, a saber, Netscape Navigator y Microsoft Internet Explorer.

### Web Services

La interoperabilidad con los sistemas de las agencias de viajes no debe estar basada en Web Services.

## 4.3 Tecnología

El desarrollo completo del Sistema debe estar realizado en la plataforma Microsoft .NET, basándose fuertemente en las tecnologías involucradas en la plataforma como ser ADO.NET y ASP.NET. El lenguaje de programación es C#.NET.

## **4.4 Sistemas Existentes**

### **Sistema de Facturación**

La cadena hotelera ha adquirido un módulo de software que gestiona las cuentas de clientes, los pagos y el registro de venta de servicios y productos. Este módulo respeta la regulación impositiva presente en el país de residencia.

### **4.5 Soporte**

El Sistema de Gestión Hotelera tendrá mantenimiento evolutivo permanente orientado principalmente al desarrollo de nuevos módulos para cubrir nuevos servicios brindados por los hoteles.

El Subsistema de Reservas tendrá mantenimiento adaptativo, mejorando la interacción usuario-máquina mediante la adaptación de los casos de uso del subsistema.

# 5 Vista QoS

---

Describe los requerimientos no-funcionales del Sistema dentro de las categorías usabilidad, confiabilidad y performance descritas en FURPS+.

## 5.1 Usabilidad

La interfaz de usuario orientada al web para todos los actores humanos del sistema disminuye la necesidad de capacitación de los usuarios. El sitio es simple, orientado a los casos de uso.

La documentación de usuario está anexada a la interfaz propiamente. En cada lugar donde se encuentre el usuario tendrá disponible una opción de ayuda (haciendo clic sobre el ícono que se muestra a la derecha) que le indicará en qué contexto se encuentra, qué información está viendo, qué información debe proveer y cuál será la actividad que realizará el sistema una vez provista dicha información. No se proveerá documentación de usuario impresa.



El Sistema de Gestión Hotelera será utilizado por clientes de todo el mundo. Adicionalmente, la Organización Pro-Turismo exige que para anunciar servicios en su portal, éstos deben ser provistos en español, inglés y portugués. Estos tres idiomas son soportados por el producto desarrollado (el usuario puede alternar entre idiomas usando el ícono a la derecha). El sistema detectará el origen del usuario para proveerle el idioma que mejor se adapte a él.



## 5.2 Confiabilidad

El Subsistema de Reserva no debe fallar en los procesos de Hacer Reserva o Tomar Reserva. Éstos son críticos para el hotel.

El resto de los procesos debe tener baja frecuencia de fallas, siendo más tolerante para los procesos batch Procesar No Presentados (CU5) y Remover Reservas Caducas (CU6).

## 5.3 Performance

El Subsistema de Reservas tiene fuertes restricciones de performance al momento de realizar una reserva (CU1) y de tomar una reserva (CU4).

En el caso de Hacer Reserva (CU1), estando el cliente registrado, el curso típico de eventos debe llevar a lo sumo 5 segundos una vez que el cliente indica los detalles de su reserva.

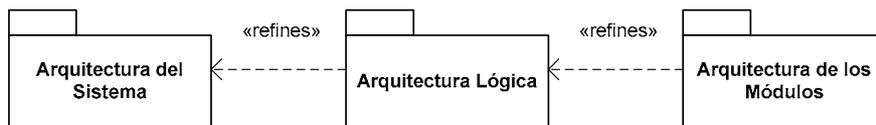
El proceso de check-in (CU4) debe llevar a lo sumo 2 minutos, en el caso que el huésped tenga una reserva. Ese tiempo debe cubrir el caso en que el huésped no conozca los detalles de la reserva.

El resto de los procesos debe poder realizarse en un tiempo razonable.

# 6 Vista Lógica

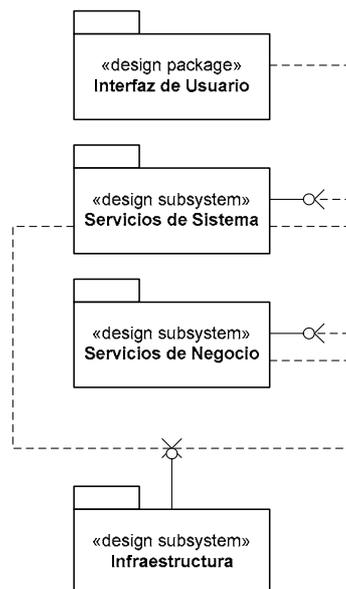
Esta vista presenta tres niveles de arquitectura para el Subsistema de Reservas del Sistema de Gestión Hotelera. Cada nivel corresponde a un refinamiento del nivel anterior. El último nivel es el que presenta mayores detalles; en él, se presentan los módulos participantes de la arquitectura junto a un diagrama. El tipo de diagrama utilizado varía según el módulo en cuestión, lo cual se debe principalmente a que los módulos tienen diferentes roles, según su ubicación en el nivel anterior.

Este capítulo se organiza de la siguiente forma:



## 6.1 Arquitectura del Sistema

En el primer nivel se especifica el patrón de arquitectura para el Subsistema de Reservas. El mismo está organizado utilizando el patrón de arquitectura en capas; el mismo es relajado y se conforma de cuatro capas. El siguiente diagrama presenta la Arquitectura del Sistema.



El patrón de arquitectura elegido para el Subsistema de Reservas está alineado con el patrón del Sistema de Gestión Hotelera. Los módulos que conformarán el Subsistema de Reservas convivirán en estas capas con otros módulos del Sistema de Gestión Hotelera.

Cada capa determina un rol para los módulos que residen en ella.

La Interfaz de Usuario tiene como objetivo el manejo de la lógica del usuario. Está conformada por el conjunto de páginas web dinámicas y por módulos que encapsulan la lógica de los casos de uso.

Los Servicios del Sistema representan los servicios básicos que debe proveer el sistema; estos servicios son directamente utilizados por los módulos de la capa superior. Los servicios en esta capa son particulares para cada subsistema del Sistema de Gestión Hotelera. Aquí se

utiliza una aplicación del GRASP Controller, haciendo un híbrido entre use-case controller y facade controller.

Los Servicios de Negocio son servicios de manejo de información del negocio; son servicios aún más básicos que el de la capa superior. Esto permite reutilizar estos módulos en otros subsistemas del Sistema de Gestión Hotelera.

La capa de Infraestructura contiene todos los módulos necesarios para utilizar los servicios de la plataforma. En esta capa residen principalmente adaptadores de los servicios brindados por la plataforma.

## 6.2 Arquitectura Lógica

La Arquitectura Lógica presenta un refinamiento de la Arquitectura del Sistema. La dimensión Requerimientos, principalmente la Vista de Casos de uso, va a verse realizada por los módulos aquí presentados. Se analizará los módulos presentes en cada capa de la Arquitectura del Sistema, presentando finalmente la Arquitectura Lógica.

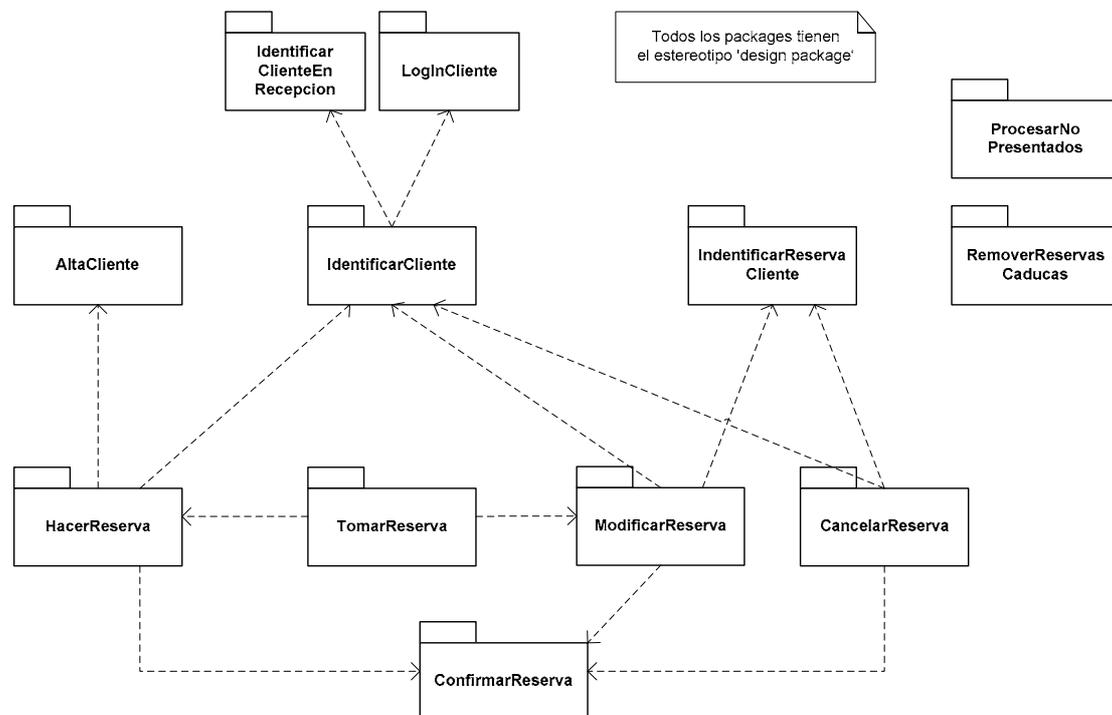
### Interfaz de Usuario

La Vista de Casos de Uso muestra el front-end del sistema. El mismo es generado dinámicamente utilizando tecnología de contenido web dinámico. Desde el punto de vista del back-end se tiene un conjunto de páginas dinámicas generadas a partir de los procesos llevados a cabo por el sistema.

Cada caso de uso indicado en dicha vista requiere, en general, más de una interacción con el usuario. Esta secuencia de páginas es en general variable, dependiendo de las acciones del usuario. La versión expandida de cada caso de uso representa la lógica de dicha interacción.

Esta capa consiste de un módulo por cada caso de uso identificado. Cada módulo contiene la lógica que lleva adelante el caso de uso y un conjunto de páginas dinámicas utilizadas por dicha lógica.

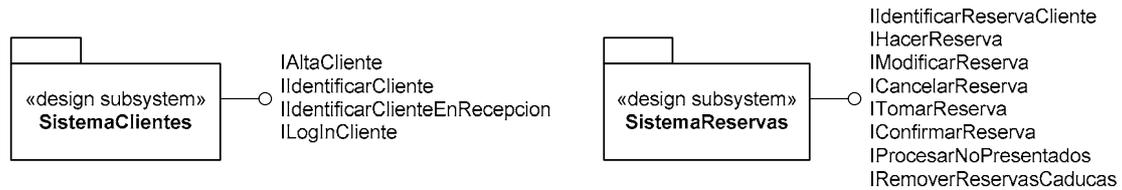
Los módulos identificados y sus interdependencias se presentan en el siguiente diagrama.



## Servicios de Sistema

Cada módulo de la capa superior utiliza servicios de esta capa. Se cuenta con una interfaz por caso de uso; ésta ofrece los servicios que el módulo que maneja la lógica del caso de uso requiere. Exceptuando los módulos Alta Cliente e Identificar Cliente, el resto tiene que ver con el Subsistema de Reservas. Los servicios requeridos por estos casos de uso son cohesivos y por lo tanto pueden ser ofrecidos por el mismo módulo. Por GRASP facade controller restringido a las operaciones de un subsistema, se crea un módulo por subsistema. Este módulo realizará todas las interfaces requeridas por los casos de uso asociados al subsistema.

El siguiente diagrama presenta los módulos identificados y sus interdependencias.

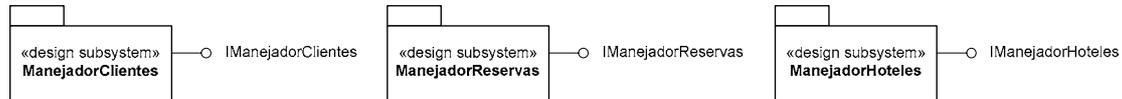


## Servicios de Negocio

Los módulos localizados en esta capa son de interés para el Sistema de Gestión Hotelera y no de exclusividad para el Subsistema de Reservas. En cambio, los servicios provistos por estos módulos son los necesarios para poder llevar adelante las operaciones de los módulos de la capa superior.

Los módulos identificados son Manejador de Clientes, Manejador de Hoteles y Manejador de Reservas. Cada módulo ofrece una única interfaz con los servicios requeridos.

El siguiente diagrama presenta los módulos detectados y sus interdependencias.



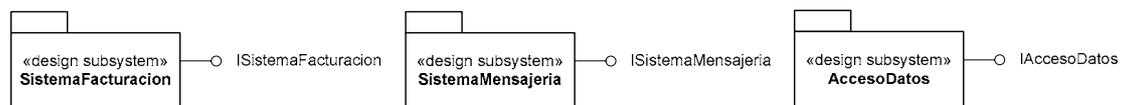
## Infraestructura

Los módulos localizados en esta capa son de dos tipos, adaptadores a sistemas existentes, o servicios generales útiles para cualquier aplicación. Estos módulos están disponibles para todos los módulos en las otras capas, y por ello la arquitectura en capas del Subsistema de Reservas es relajada.

Dentro de la primera categoría se encuentra el Sistema de Facturación. Este módulo encapsula todo lo relativo con la comunicación con dicho sistema. A su vez se cuenta con un Sistema de Mensajería. Este módulo encapsula todo lo relativo con la comunicación con el sistema de envío de mensajes. Desde el punto de vista de [GHJ]+95], ambos son una aplicación de Adapter y Proxy.

En la segunda categoría se encuentra el módulo de Acceso a Datos, el cual encapsula la conexión al motor relacional.

El siguiente diagrama presenta los módulos detectados.



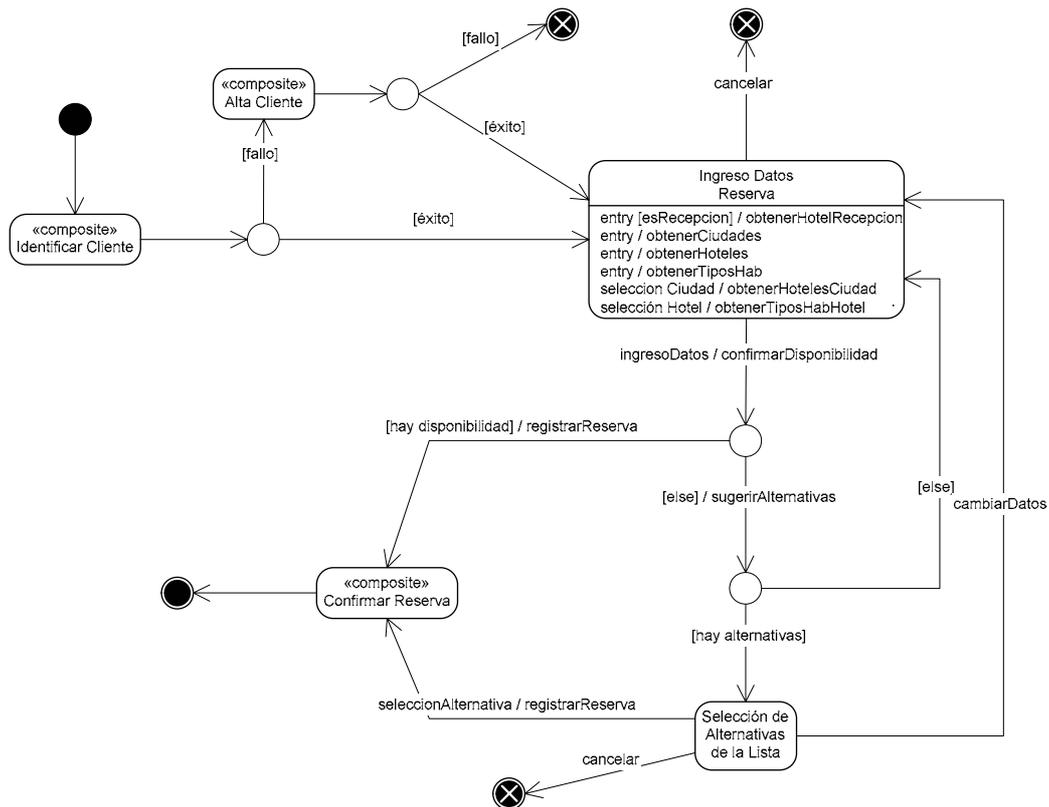
## 6.3 Arquitectura de los Módulos

La Arquitectura de los Módulos presenta un refinamiento de la Arquitectura Lógica. Ésta incluye, para cada módulo, el punto de vista que mejor define su diseño. Para cada tipo de módulo, i.e. para los módulos de cada capa, se utilizará un tipo de diagrama diferente.

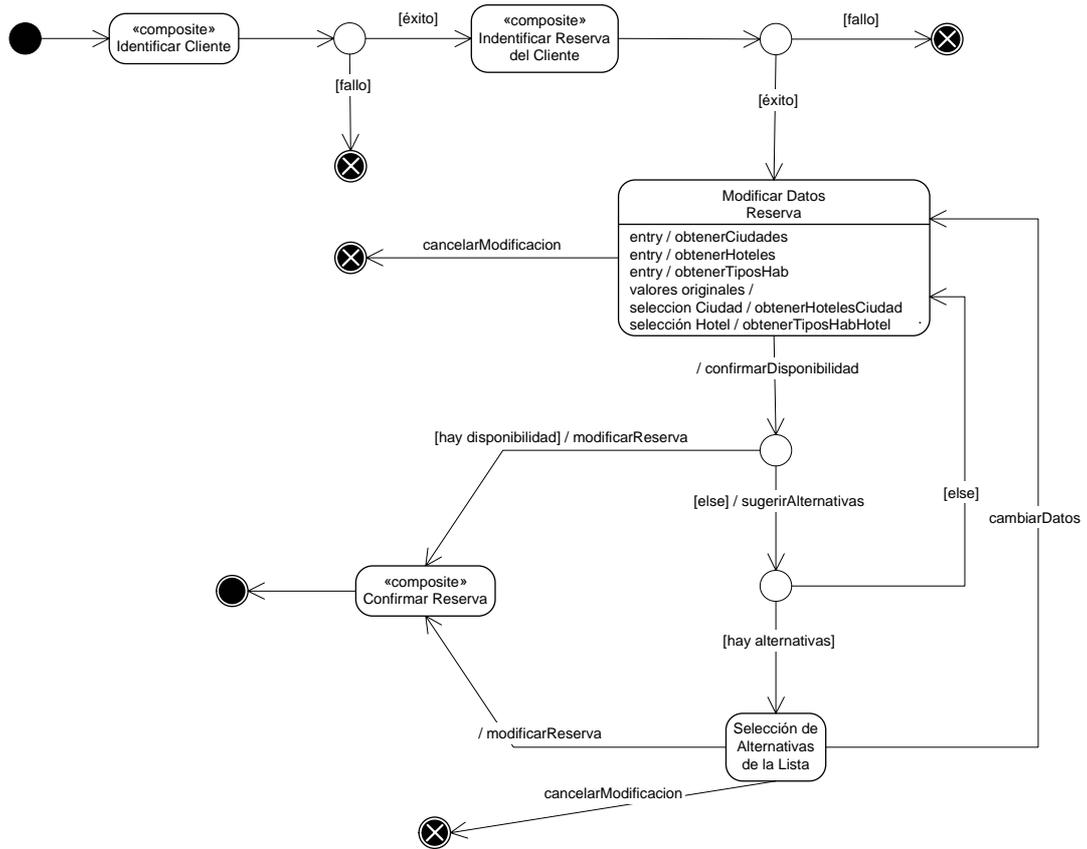
### Interfaz de Usuario

Los módulos en esta capa encapsulan la lógica del caso de uso. Esta lógica puede representarse mediante una máquina de estados que guía la interacción entre el sistema y el usuario. Por esta razón un modelo de estados es el adecuado para describir su diseño interno.

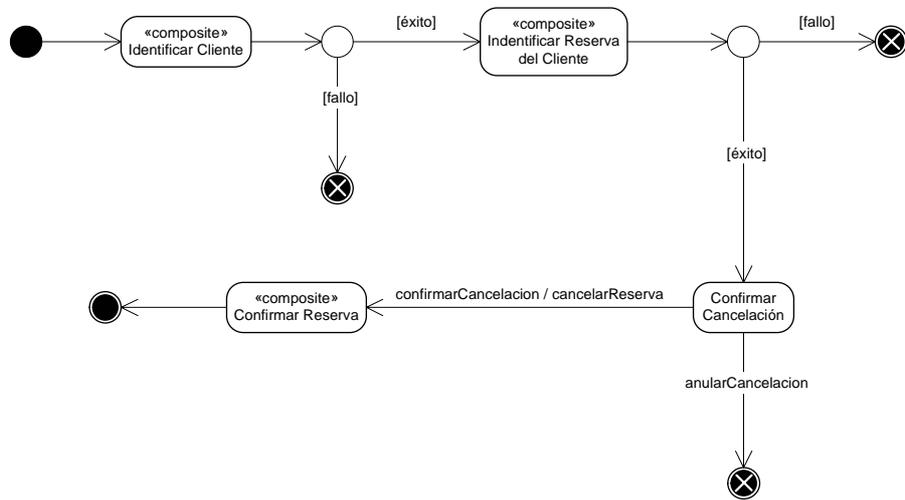
#### Hacer Reserva (CU1)



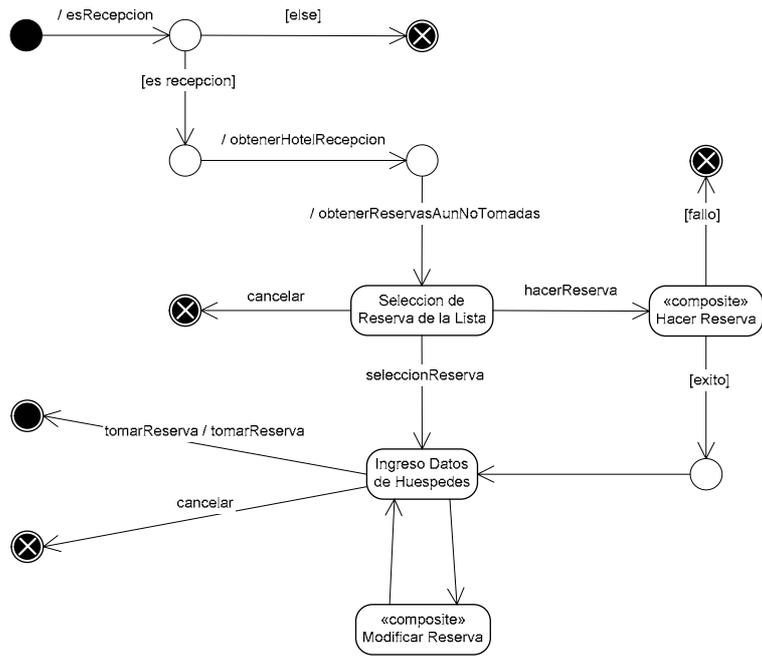
### Modificar Reserva (CU2)



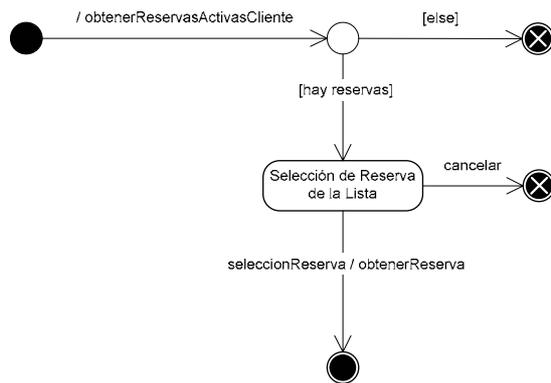
### Cancelar Reserva (CU3)



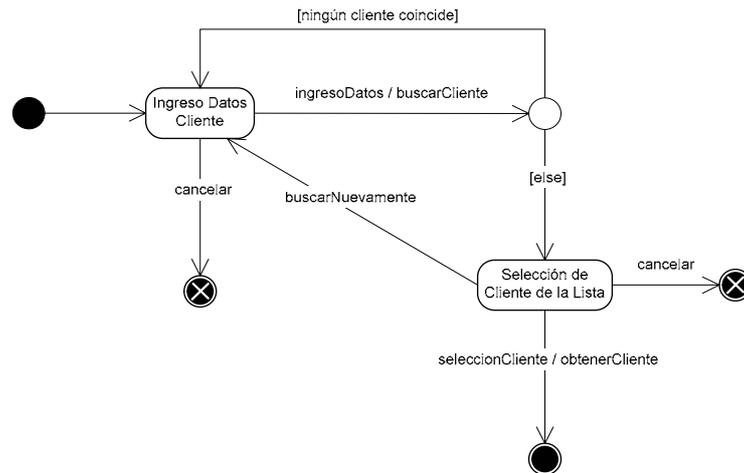
### Tomar Reserva (CU4)



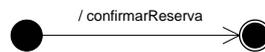
### Identificar Reserva de Cliente (CU7)



## Identificar Cliente en Recepción (CU8)



## Confirmar Reserva (CU10)



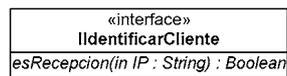
## Servicios de Sistema

Los módulos de esta capa encapsulan la lógica del sistema. Los servicios ofrecidos son realizados utilizando Servicios de Negocio y servicios de Infraestructura; esta realización se hace mediante el envío de mensajes a los módulos ubicados en estas capas. Por esta razón un diagrama de secuencia es el más adecuado para describir su diseño interno.

Para cada módulo se presenta los servicios brindados por cada una de sus interfaces y luego los diagramas de secuencia para las operaciones más complejas.

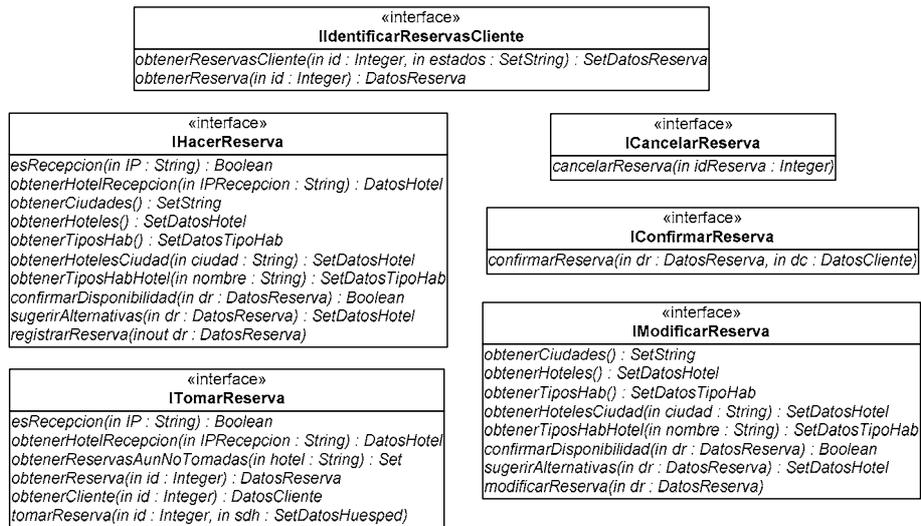
### Subsistema de Clientes

Las interfaces provistas por este módulo se describen a continuación. La realización de las mismas, en una clase llamada Controller, se hace mediante solicitud de los servicios respectivos a los módulos en la capa de servicios de negocio.



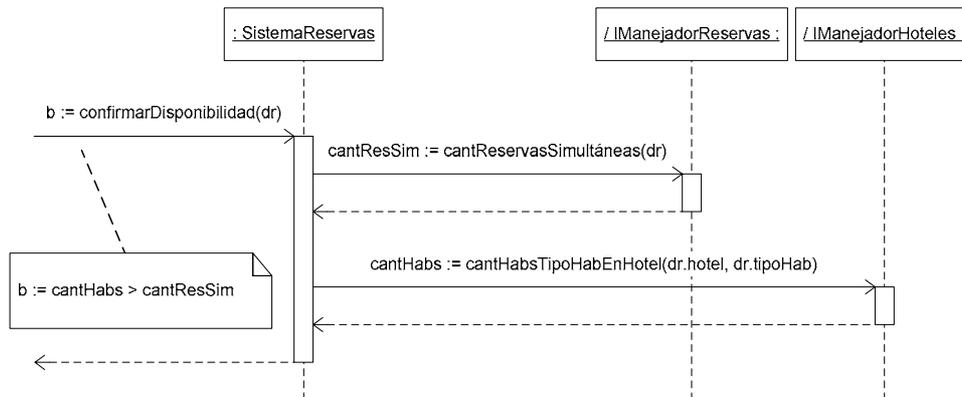
## Subsistema de Reservas

Las interfaces provistas por este módulo se describen a continuación.

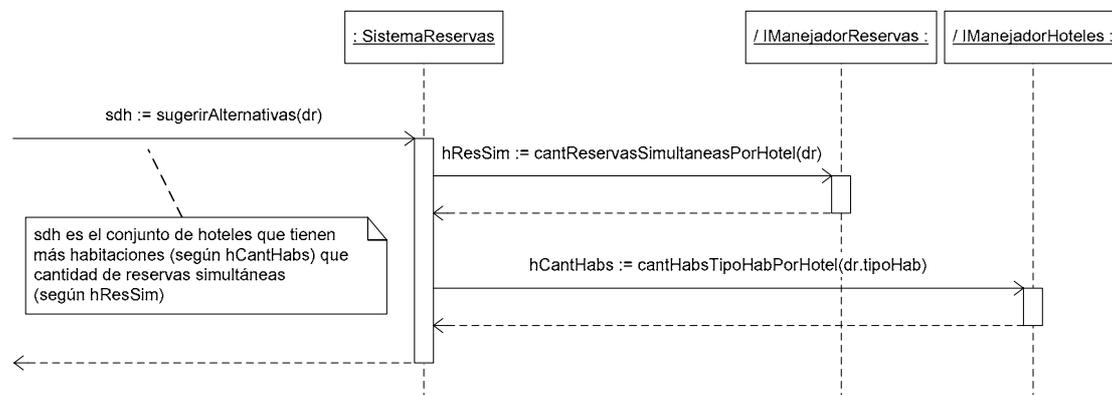


Estas interfaces son realizadas en la clase Controller de este módulo. Las operaciones con la misma firma ubicadas en diferentes interfaces, e.g. esRecepcion, son realizadas por el mismo método. La mayoría de los métodos simplemente delegan la solicitud del servicio respectivo a los módulos en la capa de negocios. Para aquellos métodos que realizan actividades adicionales se presentan a continuación los diagramas de secuencia que muestran su diseño.

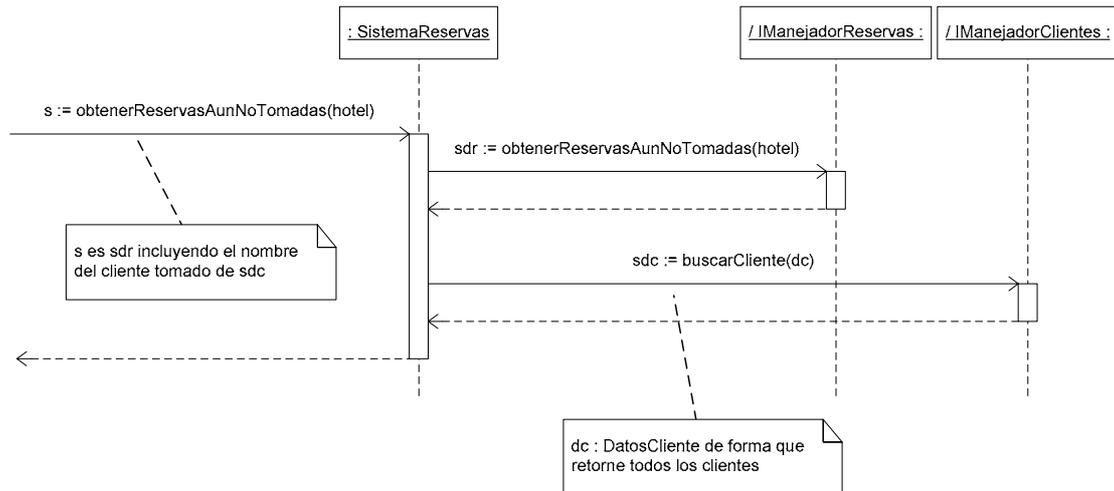
**confirmarDisponibilidad(in dr : DatosReserva) : Boolean**



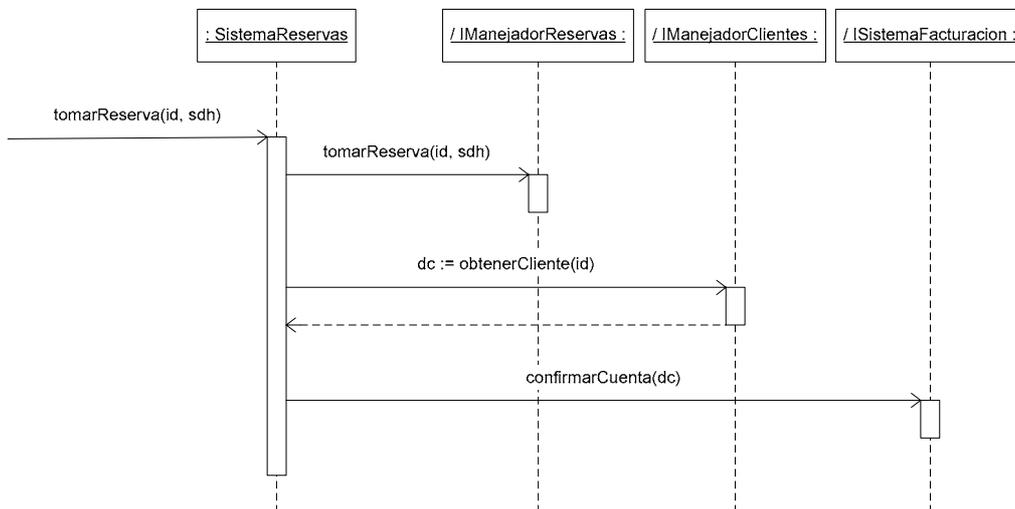
**sugerirAlternativas(in dr : DatosReserva) : SetDatosHotel**



`obtenerReservasAunNoTomadas(in hotel : String) : Set`



`tomarReserva(in id : Integer, in sdh : SetDatosHuesped)`



## Servicios de Negocio

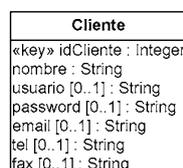
Estos servicios manejan la información del sistema; encapsulan la forma en que los datos están almacenados, la distribución de estos datos y el acceso a ellos. Por esta razón, una interfaz identificando sus servicios y el modelo de información es el adecuado para describir su diseño interno.

### Manejador de Clientes

La interfaz soportada por este módulo es la siguiente.

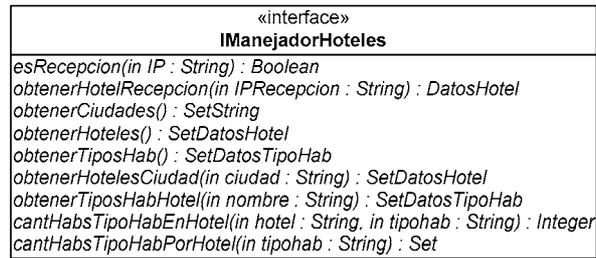


El modelo de información asociado al módulo se presenta en la siguiente figura.

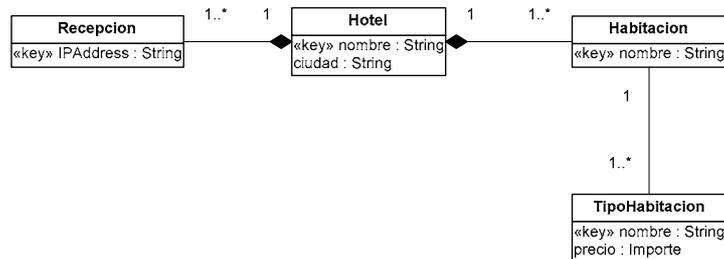


## Manejador de Hoteles

La interfaz soportada por este módulo se presenta a continuación.



El modelo de información asociado al módulo es el siguiente.

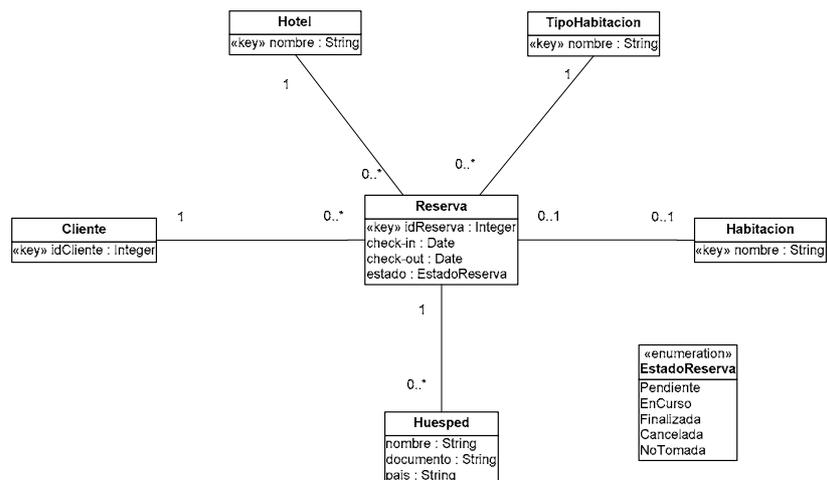


## Manejador de Reservas

La interfaz soportada por este módulo es la siguiente.

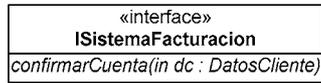
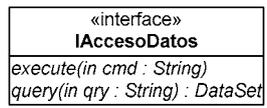


El modelo de información se presenta en la siguiente figura.



## Infraestructura

El refinamiento para los módulos en esta capa consta de la interfaz soportada por los mismos. Se presenta a continuación estas interfaces.



# 7 Vista de Procesos

La Vista de Procesos describe los módulos activos del Subsistema de Reservas; estos son módulos que estarán en ejecución en forma simultánea. Esta vista describe, además, el soporte multi-usuario de la aplicación.

## 7.1 Procesos Distribuidos

### Interfaz de Usuario

El Subsistema de Reservas es una aplicación basada en el web. Por esta razón cuenta con un grado de distribución a nivel de interfaz de usuario. A nivel de usuario final, corre en su estación de trabajo una aplicación llamada browser, como ser Netscape Navigator o Microsoft Internet Explorer. Esta aplicación es la encargada de presentar al usuario la interfaz de la aplicación y de enviar al servidor las acciones que el usuario realiza.

Por el lado del servidor se encuentra otra aplicación general, llamado servidor web. El servidor web en uso es Microsoft Internet Information Server, utilizando tecnologías ASP.NET. El servidor web recibe las solicitudes del usuario, utilizando el protocolo HTTP, y redirige dichas solicitudes a las páginas dinámicas ubicadas en los módulos de la capa Interfaz de Usuario. El servidor web asocia a cada usuario información de sesión y es él quién se encarga de gestionar la atención a múltiples clientes en forma simultánea.

### Distribución de las Capas

Los módulos presentes en las diferentes capas que componen a la aplicación no se encuentran distribuidos. Por dicha razón un enfoque de Factory simple fue utilizado para la localización de los proveedores de servicios de cada uno de los módulos. No se utiliza la tecnología Remoting de .NET.

Las razones de no utilizar distribución a este nivel son por performance de las operaciones marcadas como críticas y para disminuir el riesgo del proyecto en función de las tecnologías.

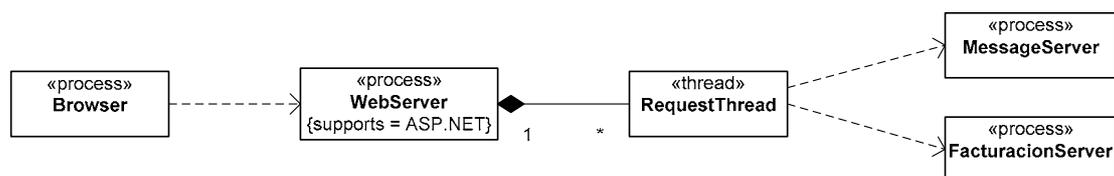
### Servicios de Infraestructura

Los servicios de infraestructura son de dos tipos, aplicaciones legadas que deben ser utilizadas y el propio motor de base de datos. El motor de base de datos corre en forma independiente, atendiendo incluso solicitudes de otras aplicaciones.

Por otro lado, las aplicaciones legadas como los servicios de mensajería y el sistema de facturación fueron adaptadas de forma que exportan sus servicios mediante web services. Así, los módulos presentes en la capa Infraestructura son un adaptador propio a dichos web services.

## 7.2 Arquitectura de Procesos

En base a las decisiones descritas en la sección anterior, se muestra a continuación la arquitectura de procesos del Subsistema de Reservas.



El WebServer destina un hilo de ejecución (RequestThread en la figura) a cada solicitud del usuario. Estos threads están asociados a una sesión que es preservada entre diferentes requests. Cada RequestThread ejecuta todos los módulos en la aplicación del sistema de reservas que sean necesarios para llevar adelante la solicitud del usuario. Cuando es necesario éste se comunica mediante web services con MessageServer o FacturacionServer (servicios que están corriendo en otro servidor).

Es importante notar que la aplicación desarrollada no codifica en forma explícita esta arquitectura de procesos. La tecnología subyacente brinda servicios de alto nivel que lo encapsulan.

Por último, cuando el servidor web crea un RequestThread y comienza con la ejecución del code-behind de las páginas dinámicas, éstas asocian al RequestThread el contexto transaccional. El módulo de acceso a datos toma del RequestThread el contexto transaccional y lo utiliza al momento de realizar acciones contra el motor de base de datos.

# 8 Vista de Implementación

La vista de implementación presenta los componentes de deployment contruidos para el Subsistema de Reservas.

Bajo la tecnología .NET, la unidad de deployment son los assemblies. Un assembly es una colección de funcionalidades construida, versionada e instalada como una unidad de implementación, y contiene, en general, múltiples archivos. Un assembly es el building block del sistema a nivel de implementación.

## 8.1 Estructura de la Aplicación

Cada capa, es decir cada subsystem en el primer refinamiento de la arquitectura lógica, fue desarrollada en un assembly independiente del resto. Dentro del assembly de cada capa se encuentran los módulos que residen en ella desde el punto de vista lógico.

Los assemblies para las capas de Servicios de Sistema, Servicios de Negocio e Infraestructura son, respectivamente, Sistema.dll, Negocio.dll e Infraestructura.dll.

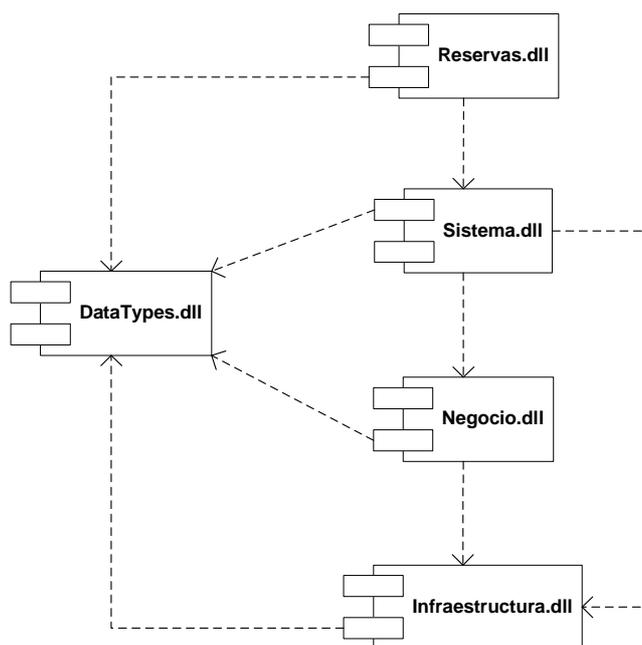
La interfaz de usuario corresponde a una aplicación ASP.NET. La misma genera un assembly encargado del code-behind de las páginas dinámicas, y con un conjunto de packages que contienen las páginas dinámicas propiamente. Así, se tiene el assembly Reservas.dll, y un conjunto de packages con las páginas dinámicas.

Se cuenta con un assembly adicional que contiene la biblioteca de clases utilizada para compartir información entre las capas, a saber, DataTypes.dll.

Es importante notar que el assembly Infraestructura.dll utiliza a los sistemas legados mediante WebServices, por lo cual depende de dichos servicios para su correcto funcionamiento. Esta dependencia puede marcarse hacia la descripción de dichos servicios.

## 8.2 Arquitectura de Implementación

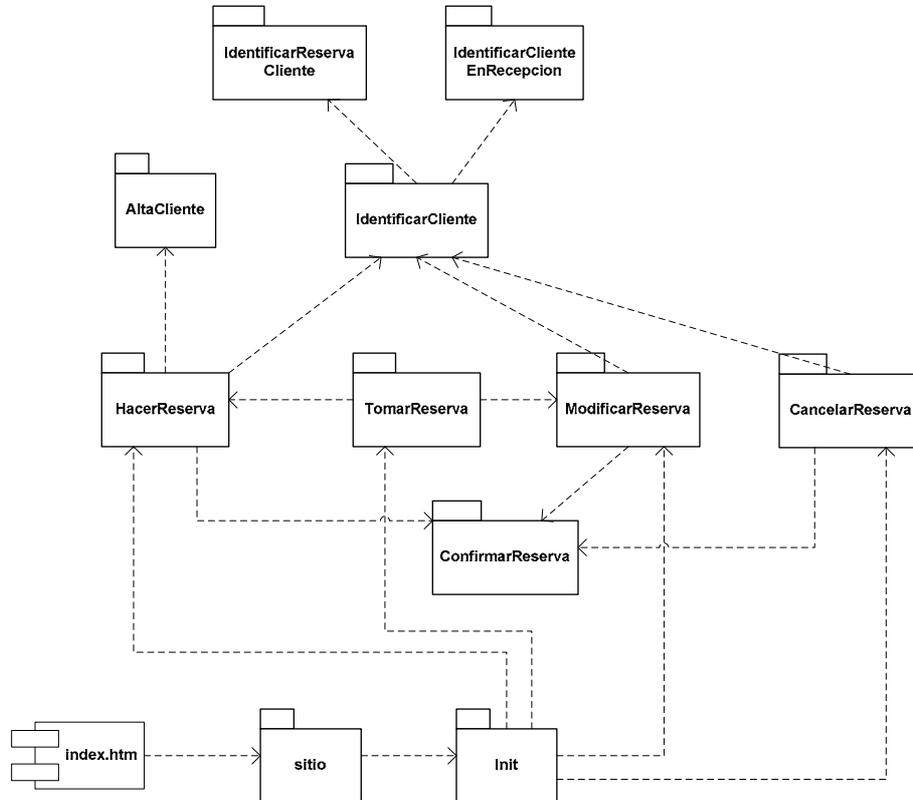
Se presenta a continuación un diagrama de componentes que indica las dependencias entre los assemblies mencionados en la sección anterior.



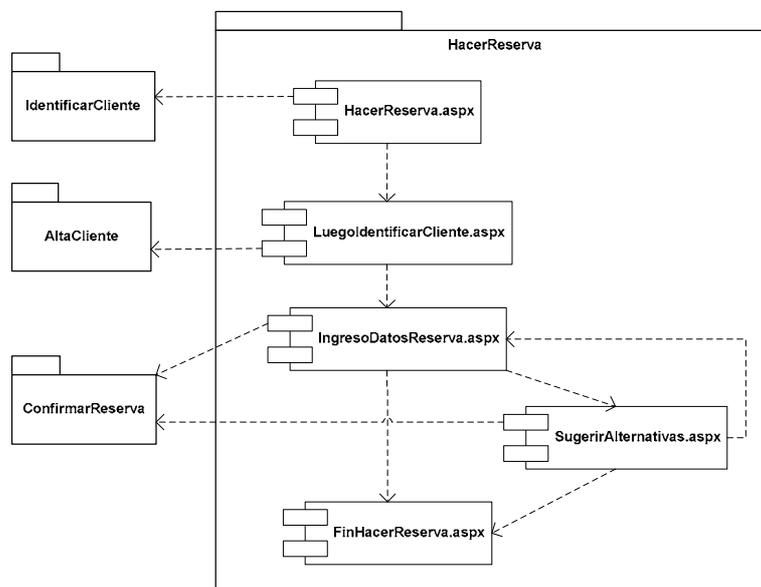
## Front-End

Como se mencionó antes, el front-end de la aplicación tiene dos partes fundamentales, el code-behind, encapsulado en Reservas.dll, y un conjunto de packages con las páginas dinámicas.

El siguiente diagrama presenta los detalles de estas dependencias.

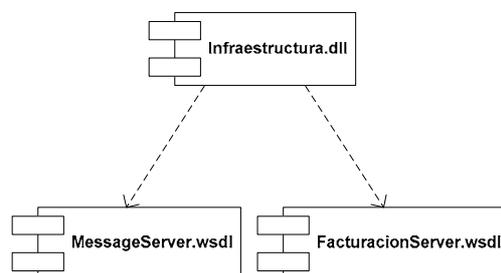


A su vez, cada package contiene un conjunto de páginas dinámicas. Todas ellas dependen del assembly Reservas.dll, lo cual no fue indicado en el diagrama. Además, existe una dependencia de la mayoría de dichos packages hacia el package images, lo cual tampoco fue indicado por claridad del diagrama. Se presenta el contenido del package HacerReserva en la siguiente figura.



## Infraestructura

El assembly Infraestructura.dll contiene referencias a WebServices por los cuales accede a los servicios de mensajería de la empresa, MessageServer, y al Sistema de Facturación, FacturacionServer. El siguiente diagrama presenta estas dependencias.



Se presenta a continuación el fragmento de MessageServer.wsdl que describe las funcionalidades del Webservice.

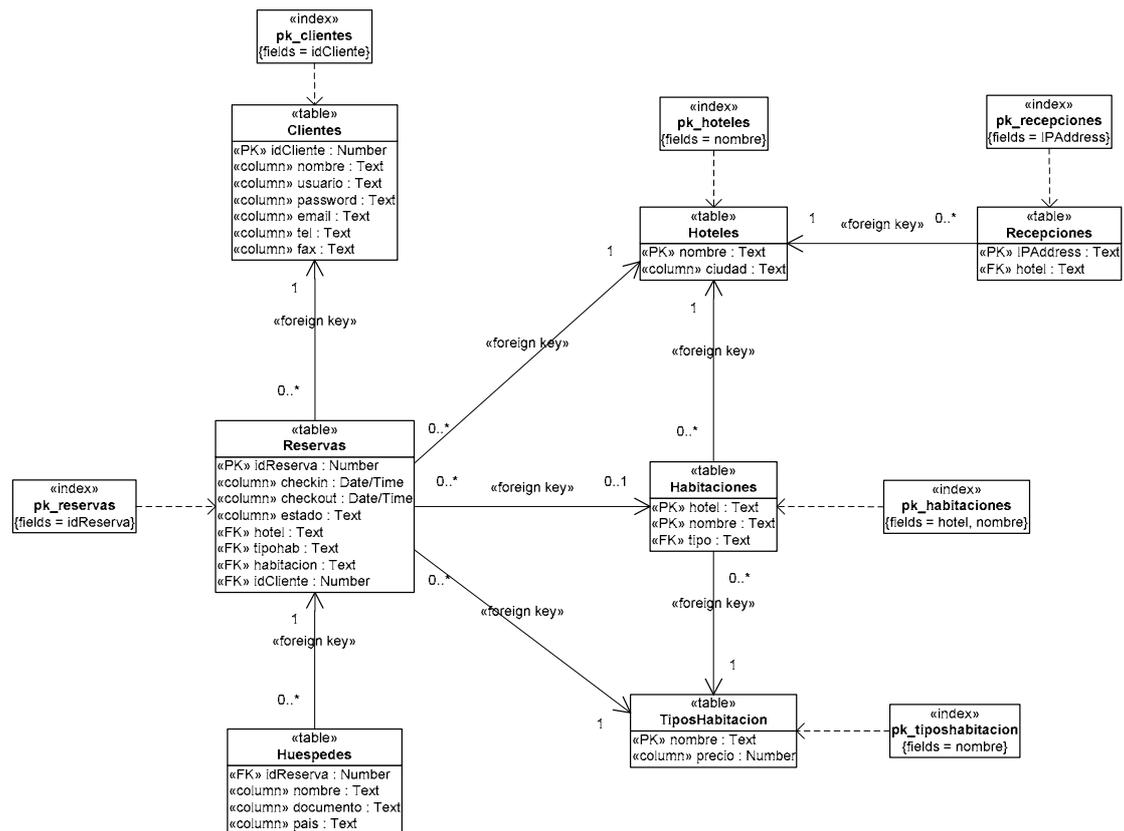
```
<?xml version="1.0" encoding="utf-8"?>
<definitions xmlns:s1="http://tempuri.org/AbstractTypes"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:s0="http://tempuri.org/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  targetNamespace="http://tempuri.org/" xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types>
    ...
  </types>
  ...
  <binding name="MailServiceHttpPost" type="s0:MailServiceHttpPost">
    <http:binding verb="POST" />
    <operation name="sendMail">
      <http:operation location="/sendMail" />
      <input>
        <mime:content type="application/x-www-form-urlencoded" />
      </input>
      <output />
    </operation>
  </binding>
  <service name="MailService">
    <port name="MailServiceSoap" binding="s0:MailServiceSoap">
      <soap:address location="http://localhost/MessageServer/MailService.asmx" />
    </port>
    <port name="MailServiceHttpGet" binding="s0:MailServiceHttpGet">
      <http:address location="http://localhost/MessageServer/MailService.asmx" />
    </port>
    <port name="MailServiceHttpPost" binding="s0:MailServiceHttpPost">
      <http:address location="http://localhost/MessageServer/MailService.asmx" />
    </port>
  </service>
</definitions>
```

# 9 Vista de Datos

En esta vista se presenta el modelo de datos utilizado, su distribución, y una descripción de los servicios de persistencia utilizados.

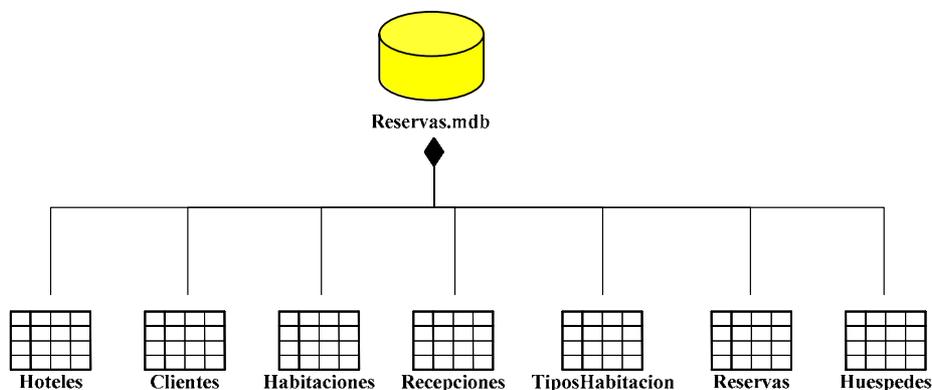
## 9.1 Modelo de Datos

Se utiliza una única base de datos relacional corriendo sobre el motor de base de datos Microsoft Access. La siguiente figura presenta el modelo de datos.

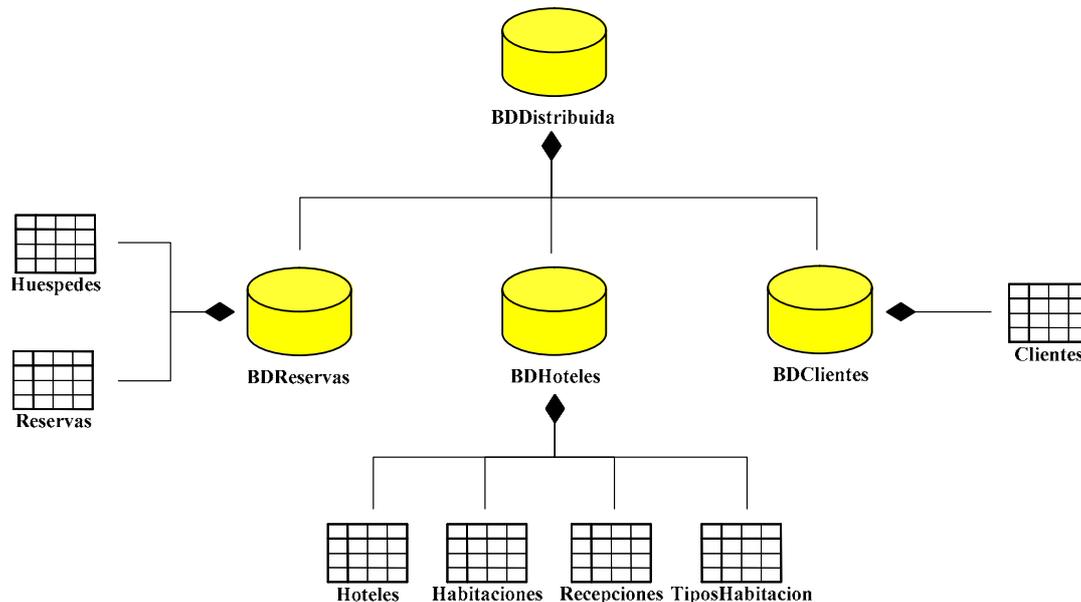


## 9.2 Distribución

No hay distribución a nivel de datos; todas las tablas radican en la misma base de datos.



El diseño de la aplicación permite la distribución vertical de los datos, siguiendo la distribución sugerida por los modelos de información de los módulos de la capa de negocio. La siguiente figura presenta dicha posible distribución.



### 9.3 Servicios de Persistencia

Los servicios de persistencia del Subsistema de Reservas están alineados con los servicios para el Sistema de Gestión Hotelera. Los mismos pueden separarse en cuatro niveles, los cuales serán mencionados a continuación.

En el nivel inferior se encuentra el motor de base de datos relacional, que para el sistema actual se utiliza el producto Microsoft Access.

Por encima de éste se ubica la tecnología ADO.NET provista por el framework .NET. Esta tecnología brinda servicios de alto nivel, e independientes del producto, para acceder a fuentes de datos.

La tecnología ADO.NET es utilizada directamente por el módulo AccesoDatos. Éste utiliza un OLE DB Provider con el string de conexión adecuado para el motor de Microsoft Access. El módulo AccesoDatos, ubicado en la capa Infraestructura, provee los servicios presentados en la siguiente figura.



La interfaz ofrece un servicio de ejecución de comandos, *execute*, para ejecución de las sentencias SQL UPDATE, INSERT y DELETE. El otro servicio, *query*, destinado a sentencias SQL basadas en SELECT. El tipo de retorno de este último es un DataSet, clase miembro de ADO.NET.

Por último, sobre el módulo AccesoDatos, se ubican los módulos de la capa Servicios de Negocio. Los servicios brindados por éstos utilizan el módulo AccesoDatos para ejecutar sus sentencias y consultas SQL. Para el caso de las consultas, el DataSet es encapsulado en otros tipos de datos dentro del Subsistema de Reservas; los mismos se encuentran en el package DataTypes.

## 9.4 Servicios de Transaccionalidad

El poder transaccional utilizado es el brindado por ADO.NET. En una sesión de usuario corre, por vez, un módulo de la interfaz de usuario. Cada ejecución de un caso de uso completo, dirigida por uno de estos módulos, está enmarcada en una transacción. Al iniciar un caso de uso da comienzo una transacción, y al terminar el mismo, se realiza el commit. Cuando el caso de uso falla se realiza un rollback.

El enmarcado de las transacciones es realizado en la biblioteca Manager ubicado en la Interfaz de Usuario. Al comienzo de un caso de uso, el mismo solicita el comienzo de una transacción, y al final de éste se realiza un commit o rollback según corresponda. La propagación del contexto transaccional se realiza junto al hilo de ejecución. No es necesario ningún mecanismo de propagación de contexto transaccional especial dado que toda la aplicación corre en forma local dentro de un único servidor.

# 10 Vista de Deployment

La vista de deployment presenta la infraestructura necesaria para instalar el Subsistema de Reservas.

Se presenta aquí la arquitectura técnica de la aplicación indicando los nodos presentes en la infraestructura tecnológica esperada, y la localización de los componentes en dichos nodos.

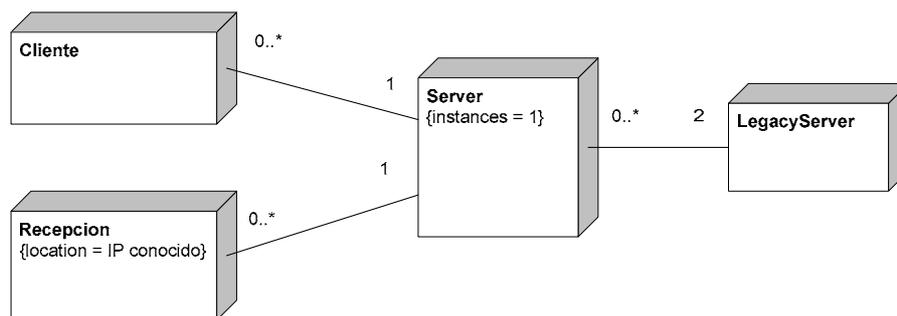
## 10.1 Arquitectura Técnica

Considerando la distribución de la aplicación desde el punto de vista de los procesos, es posible identificar cuatro tipos de nodos, a saber, Cliente, Recepcion, Server, LegacyServer.

El primer nodo representa a las estaciones de trabajo de los usuarios finales. El nodo Recepcion corresponde a la estación de trabajo en Recepción. Estos nodos son distinguidos dado que es necesario conocer su dirección IP, ya que en base a ésta se identifica la forma en que los casos de uso deben ser llevados adelante.

El nodo Server representa a aquel equipo en donde correrá el servidor web y la aplicación del Subsistema de Reservas.

El último nodo, LegacyServer, representa a aquella infraestructura informática necesaria para correr los sistemas legados.



## 10.2 Tecnología requerida

Las estaciones de trabajo de los clientes (Cliente) deben contar con un browser, ya sea Netscape Navigator o Microsoft Internet Explorer, con la habilitación de ejecución de JavaScript. Este tipo de nodos comprende también a las terminales que están a disposición en el hall de los hoteles.

Las estaciones de trabajo de las recepciones de los hoteles (Recepcion) debe correr Microsoft Internet Explorer, en modo Kiosk (iexplore -k). Estas estaciones de trabajo deben tener un número IP conocido y fijo ya que de eso depende el correcto funcionamiento de la aplicación.

Hay únicamente una instancia del nodo Server, la cual centraliza todos los requerimientos de los clientes. Este nodo debe correr Internet Information Server con soporte del framework .NET.

Los nodos LegacyServer representan a aquellos equipos que corren los sistemas legados. El requerimiento tecnológico exigido es el soporte de WebServices.

## 10.3 Deployment

Los assemblies presentados en la Vista de Implementación residen en el nodo de tipo Server presentado en las secciones anteriores.

La configuración de las estaciones de trabajo de tipo Recepcion, en particular su dirección IP, debe estar indicada en la base de datos.

Por último, los nodos LegacyServer deben tener instalado los servicios que ofrecen los WebServices.