

CAPÍTULO I
INTRODUCCION

1.1. INTRODUCCIÓN

El incesante y continuo avance de la tecnología y el progresivo aumento de los Sistemas de Información han tenido un gran impacto en nuestra sociedad y en las organizaciones, obligándolas a automatizar y mejorar sus sistemas de información, debido a que la información juega un papel muy importante y se ha convertido en uno de los bienes más preciados que una organización pueda tener.

Las empresas públicas y/o privadas de nuestra región no pueden quedar al margen de dichos avances tecnológicos, debido a que son una herramienta esencial, no sólo para que las empresas mejoren la eficiencia y calidad de sus servicios o productos, sino también para obtener ventajas en un mercado cada vez más competitivo, con el objeto de mejorar la productividad de las empresas o instituciones y lograr alcanzar sus objetivos, mejorando la respuesta de la empresa o institución hacia los requerimientos de sus clientes o socios.

En la actualidad, la tecnología informática está al servicio de las organizaciones para implantar, reajustar o sofisticar sistemas computarizados, que proporcionen información económica, financiera o de otra índole, con la claridad, veracidad y oportunidad deseada, lo cual permite tomar decisiones de diferente índole en forma precisa y oportuna.

La presencia de una organización en Internet es fundamental, no sólo por tener una página web o disponer de un correo electrónico, sino por los innumerables beneficios económicos, comerciales y técnicos que se presentan en este escenario; en este sentido, disponer de un portal en Internet trae beneficios inmediatos debido a la ampliación de la red de contactos, mejora de la calidad del servicio debido a que se puede trabajar desde cualquier lugar y agilizar diferentes procesos.

1.2. ANTECEDENTES

La Cámara Hotelera de Tarija es una institución privada de servicio público cuyo objetivo es agrupar a todas las empresas del ramo hotelero de Tarija (hoteles, hostales, residenciales, alojamientos y posadas). Entre sus objetivos está planificar y promover el mejoramiento y la eficiencia en la prestación de servicios de sus asociados, que contribuyan al desarrollo del turismo y la hotelería de Tarija.

La Cámara Hotelera de Tarija se fundó en febrero de 1972, actualmente cuenta con 44 socios, entre hoteles, hostales, residenciales y alojamientos. El órgano máximo dentro de la Cámara Hotelera es la Asamblea de Socios, quien designa a un Presidente, Vicepresidente, Tesorero y tres Vocales

En el ámbito nacional, se han implementado diversos sistemas web, destinados principalmente a difundir información y promocionar servicios que el ramo hotelero ofrece. Entre los más destacables están los siguientes:

- Cámara Hotelera de La Paz (<http://www.camarahoteleralapaz.org.bo>)
Contiene información explicando los objetivos de la Cámara Hotelera, quiénes fueron sus presidentes, actividades, convocatorias y principalmente muestra sitios y atracciones turísticas, así como también los principales hoteles para el turista.
- Cámara Hotelera de Santa Cruz (<http://www.camarahotelerasantacruz.net>)
Web dedicada a dar información (dirección, teléfono, fax, email) de los distintos hoteles, apart hoteles, residenciales, alojamientos, casas de huéspedes, hostales y posadas de la ciudad de Santa Cruz.
- Cámara Hotelera de Oruro (<http://www.camarahoteleraoruro.com>)
Muestra información de los principales hoteles, residenciales y alojamientos de la ciudad de Oruro.

- Cámara Hotelera del Gran Chaco (<http://yacuiba.com/hotelera/camarahotelera.htm>) Contiene sólo información de la directiva y es parte de <http://www.yacuiba.com>, página dedicada a fomentar el turismo de Yacuiba. Lo mismo ocurre con Sucre, Beni y Pando, que no cuentan con su propio web, sino son mencionadas en otras páginas departamentales dedicadas a fomentar el turismo regional.

Como se puede evidenciar, todos los sistemas existentes de las diferentes cámaras departamentales sólo muestran o difunden información relacionada a promocionar los servicios que ofrece el ramo hotelero a la población en general.

1.3. PLANTEAMIENTO DEL PROBLEMA

El elevado volumen de los datos procedentes de los establecimientos hoteleros y el procesamiento manual de estos, originan problemas en la Cámara Hotelera de Tarija, como ser: retardo en el registro de los datos, información dispersa en varios folders y archivadores, dificultad en la actualización y búsquedas de datos debido a la dispersión y/o pérdida de los mismos, incrementando el tiempo en la elaboración de reportes e informes, afectando las tareas rutinarias que se realizan y la calidad de servicio de la institución hacia sus asociados y entidades externas.

1.4. OBJETIVOS

1.4.1. OBJETIVO GENERAL

“Analizar y diseñar un Sistema de Información Vía Web para la Cámara Hotelera de Tarija”, que permita optimizar las actividades rutinarias y posibilitar el acceso oportuno a la información, mejorando de esta manera la calidad de servicio de la institución.

1.4.2. OBJETIVOS ESPECÍFICOS

- Realizar el análisis y diseño del sistema haciendo uso de la Metodología Orientada a Objetos **RUP** (Rational Unified Process).

- Aplicar estándares y lineamientos para el diseño de interfaz de usuarios para aplicaciones Web.
- Diseñar la base de datos empleando lineamientos de creación y normalización, aplicando el modelo relacional

1.5. JUSTIFICACIÓN

Debido a la situación actual de todas las organizaciones, se hace necesario que cuenten con información de manera rápida y confiable, para ello es necesario el diseño de un sistema de información automatizado, que tenga como finalidad facilitar la ejecución de las operaciones realizadas en la institución, mejorando y optimizando el registro y control de la información.

1.5.1. JUSTIFICACIÓN SOCIAL

La implementación de un sistema informático de estas características, ayudará a mejorar el servicio que la institución presta a la sociedad y por ende a sus asociados, además elevará la capacidad del personal que trabaja en ella, consiguiendo mejorar sus servicios e impulsando la utilización de nuevas tecnologías.

1.5.2. JUSTIFICACIÓN ACADÉMICA

La Carrera de Ingeniería Informática forma profesionales con los conocimientos suficientes para ayudar y apoyar a otras áreas del conocimiento mediante el desarrollo de sistemas o herramientas que faciliten el desenvolvimiento en estos campos del conocimiento. La realización del presente trabajo permite plasmar estos conocimientos adquiridos en el diseño de un sistema que se adecúe a las necesidades de la institución.

1.5.3. JUSTIFICACION TECNOLOGICA

Las nuevas tecnologías de la información son muy dinámicas y cambian día a día, debido a la utilización de nuevas metodologías y tecnologías de desarrollo, lo cual

trae consigo exigencias de hardware cada vez más avanzadas. En nuestro medio se cuenta con la tecnología informática necesaria para el desarrollo del presente trabajo.

1.6. ALCANCES

- Diseñar un entorno de gestión vía Web por medio del cual los usuarios autorizados (establecimientos hoteleros, de acuerdo a los niveles de administración establecidos) podrán registrar partes diarias de forma rápida y sencilla.
- Módulo de gestión de establecimientos hoteleros con sus respectivos datos.
- Módulo de generación de reportes diarios, mensuales y anuales de estadísticas hoteleras.
- Módulo de gestión de socios, aportes mensuales, ingresos y egresos.

1.7. LIMITACIONES

- El presente trabajo no contempla la fase de implementación del sistema.
- El sistema no diseñará módulos para la parte contable.

CAPÍTULO II
MARCO TEÓRICO

2. MARCO TEÓRICO

2.1. MÉTODO ISAC¹

ISAC: Information Systems Work and Analysis of Changes (*Desarrollo de Sistemas de Información y Análisis de Cambio*²) es un método para el desarrollo de sistemas de información orientado al cliente.

ISAC fue desarrollado para asegurar que los negocios (empresas) obtengan el sistema de información que necesitan. Para alcanzar esta meta, ISAC comienza con un análisis del problema organizacional que trata de encontrar soluciones reales a problemas reales. Si una solución implica el desarrollo de un sistema de información, entonces ISAC continúa desarrollando este sistema iniciando un estudio detallado de las actividades del negocio que serán soportadas por el sistema. Para incrementar el ajuste adecuado entre el sistema entregado y las necesidades del negocio, ambos: el análisis del problema y el estudio de actividades, están caracterizados por un alto grado de participación y cooperación de usuarios, desarrolladores y responsables del sistema de información.

ISAC consiste en cuatro etapas, las cuales se concentran en preguntas al usuario y al administrador, como se muestra en el siguiente cuadro.

¹ Requirements Engineering; R. J. Wieringa; Pág. 83.

² Traducción propia.

CUADRO N° 1
Guía de preguntas en cada etapa de ISAC

<p>1. Análisis de Cambio.</p> <ul style="list-style-type: none">• ¿Qué quiere la organización?• ¿Cuán flexible es la organización respecto a los cambios? <p>2. Estudio de Actividad.</p> <ul style="list-style-type: none">• ¿Qué actividades deben reagruparse en sistemas de información?• ¿Qué prioridades tienen los sistemas de información? <p>3. Análisis de Información.</p> <ul style="list-style-type: none">• ¿Qué entradas y salidas tiene cada sistema de información?• ¿Cuáles son los requerimientos cuantitativos de cada sistema de información? <p>4. Implementación.</p> <ul style="list-style-type: none">• ¿Qué tecnología (portadores de información, hardware, software) usamos para los sistemas de información?• ¿Qué actividades de cada sistema de información son manuales, cuáles son automatizadas?

Durante el Análisis de Cambio, los problemas actuales de la organización son analizados, se investigan diferentes soluciones posibles y una solución es elegida. ISAC especifica un modelo de actividades para la solución seleccionada, la cual es, aproximadamente, una especificación de alto nivel de la situación deseada de la organización en términos de sus actividades. **Ésta será la única etapa del Método ISAC que se aplicará en el presente trabajo, debido a que sólo se utilizará para lograr una comprensión del sistema actual y se elegirá una posible solución.** El siguiente cuadro muestra cómo ISAC responde a las preguntas del cuadro anterior.

CUADRO N° 2 EL MÉTODO ISAC

1. Análisis de Cambio
 - Listar problemas.
 - Listar los dueños de problemas.
 - Analizar los problemas.
 - Realizar el modelo de actividad actual.
 - Analizar las metas.
 - Definir las necesidades de cambio.
 - Generar las alternativas de cambio.
 - Realizar el modelo de actividades de las situaciones deseadas.
 - Evaluar las alternativas.
 - Elegir el modelo de actividad.
2. Estudio de Actividad.
 - Descomponer el modelo de actividad elegido en subsistemas de información.
 - Elaborar el modelo de actividad elegido.
 - Identificar los subsistemas de información.
 - Analizar la conveniencia para la automatización.
 - Elaborar modelos de actividad hasta que cada sistema de información sea una actividad.
 - Analizar cada sistema de información.
 - Analizar la contribución de cada sistema de información para el cambio de objetivos.
 - Generar niveles de ambición para el sistema de información.
 - Examinar la factibilidad de los niveles de ambición.
 - Realizar el análisis costo/beneficio de los niveles de ambición.
 - Elegir el nivel de ambición.
 - Coordinar los sistemas de información.
 - Definir las interfaces entre los sistemas de información.
 - Asignar una prioridad a cada sistema de información.
3. Análisis de Información. Para cada sistema de información:
 - Especificar las relaciones de entrada/salida para el sistema de información.
 - Especificar las estructuras de datos para el sistema de información
4. Implementación.

2.1.1. ANALISIS DE CAMBIO ³

El propósito del Análisis de Cambio es asegurarse que los problemas del negocio a ser solucionados sean identificados y diagnosticados correctamente. Para este fin, realizamos listas de los problemas actuales y los dueños de los problemas correlacionando estas listas entre sí. Las alternativas prometedoras son especificadas realizando un modelo de las actividades de la organización tal como son ejecutadas en la nueva situación. Luego las alternativas son evaluadas en sus posibles soluciones del problema y una de ellas es elegida para la implementación.

a) Realizar una Lista de Problemas

El propósito de esta tarea es llegar a un acuerdo entre los dueños del problema, los desarrolladores y los responsables del proceso de desarrollo acerca de qué problemas son los que el proceso de desarrollo debería resolver. Un **problema** es una insatisfacción con la situación actual, para la cual existe una posibilidad de mejora. El **dueño del problema** es una persona, grupo o institución que no está satisfecha con la situación actual. Los dueños de problemas son llamados **grupos de interés** en ISAC.

b) Listar Dueños del Problema

Una vez identificados los problemas a resolver por ISAC, podemos identificar grupos de personas para los cuales estos problemas existen. Para obtener un buen entendimiento de la situación ISAC recomienda realizar una matriz de problemas contra dueños del problema.

³ Requirements Engineering; R. J. Wieringa; Pág. 84.

c) **Analizar los Problemas**

Los problemas identificados hasta aquí son ahora sujetos a un análisis causa–efecto. El análisis causa–efecto es realizado por **especialistas del dominio**, gente que tiene conocimiento experto de los problemas bajo estudio (un dueño de problema es un usuario final del sistema requerido, un responsable, un administrador, etc.) El resultado del análisis es representado en un grafo causa–efecto, el cual es un grafo dirigido, donde los nodos representan los problemas y las flechas apuntan desde la causa al efecto. Se debe hacer un **estudio cuantitativo** de los problemas.

d) **Realizar un modelo de actividad del negocio actual**

ISAC ahora ordena realizar un modelo de actividad de la situación actual en el negocio. Un modelo de actividad actual representa las actividades tal como son realizadas actualmente en el negocio y el flujo de información importante entre estas actividades. El propósito de realizar un modelo de actividad actual es para ser capaz de tratar o discutir posibles cambios al negocio en el siguiente paso. El modelo de actividad será la plataforma para generar y tratar soluciones alternativas.

e) **Analizar metas**

Antes de investigar posibles cambios a la situación actual debemos listar las metas del proceso de desarrollo. Una **meta** es simplemente una situación deseada. Debemos especificar metas en esta etapa para saber cómo evaluar la solución propuesta. La especificación de metas nos permite entender porqué los problemas identificados antes son problemas en absoluto y lo que los responsables y dueños de problemas quieren conseguir del proceso de desarrollo.

f) **Definir necesidades de cambio**

Una de las funciones de una lista de metas es ser capaz de dar la razón porqué los problemas son en realidad problemas. Una forma de hacer esto es construir una

matriz de problemas contra metas y tratar de encontrar grupos de problemas similares. Cada grupo define una **necesidad de cambio** que actuará como una meta del proceso de desarrollo.

g) Generar alternativas de cambio

Esto se realiza en una sesión de lluvia de ideas; en la cual cualquier idea es puesta por escrito sin ninguna evaluación o crítica.

h) Realizar el modelo de actividad de las situaciones deseadas

Ahora construimos **paquetes** de una o más alternativas de cambio que merecen ser examinados. Para analizar un paquete, realizamos un modelo de actividad de él. Esto hace del paquete un tema posible de discusión racional y nos permite compararlo con la situación actual.

i) Evaluar alternativas

Para cada uno de los paquetes que se modelan, estimamos cuál es el poder de resolución para cada problema. Esto se realiza construyendo, para cada paquete, una matriz de problemas contra dueños de problemas.

j) Elegir una alternativa

El responsable se presenta con un reporte de la evaluación de alternativas y entonces elige una de ellas, no es necesario que la alternativa elegida contenga el desarrollo de un sistema de información.

2.2. PARADIGMA DE LA ORIENTACIÓN A OBJETOS

Es una nueva manera de ver y expresar el mundo, de pensar acerca de los problemas para encontrar una representación adecuada a nivel de software. El software es organizado como una colección de unidades atómicas (objetos) constituidas por datos y funciones, que interactúan entre sí.

2.2.1. BENEFICIOS DE LA TECNOLOGÍA O-O

- Promueve la reusabilidad.
- Reduce la complejidad del mantenimiento (extensibilidad y facilidad de cambios).
- Riqueza semántica.
- Disminuye la brecha semántica entre la visión interna y la visión externa del sistema.
- Facilita la construcción de prototipos.
- Diseño de software rápido.

2.3. EL PROCESO UNIFICADO RATIONAL (RUP)

Es un proceso de ingeniería de software que provee un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización desarrolladora. El objetivo del Proceso Unificado Rational (RUP) es permitir la producción de un software de la mayor calidad que satisfaga las necesidades de los usuarios finales, dentro de planificaciones y presupuestos predecibles.⁴ RUP es una instancia específica y detallada de un proceso más genérico conocido como el Proceso Unificado de Desarrollo de Software.⁵ RUP utiliza el Lenguaje

⁴ El Lenguaje Unificado de Modelado, G. Booch, J. Rumbaugh, I.Pág, 399.

⁵ Rational Unified Process: Best Practices for Software development Teams, pág. 17, <http://www.ibm.com/developerworks/rational/library/253.html> , 10/03/2010.

Unificado de Modelado (UML) para preparar todos los esquemas de un sistema software, permitiendo utilizar UML de una manera efectiva.

Se elige RUP como metodología de análisis y diseño debido a que se sostiene en los siguientes fundamentos principales:

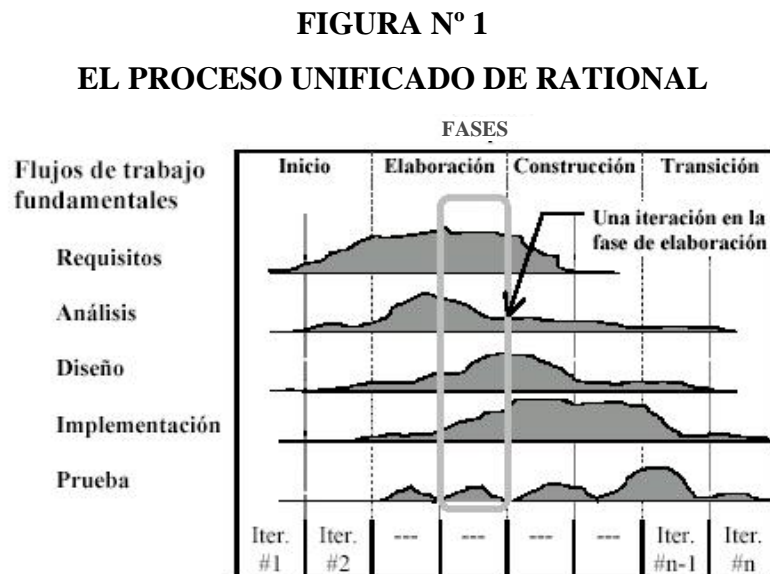
- **Dirigido por Casos de Uso.-** Los casos de uso se utilizan como artefacto principal para definir el comportamiento deseado para el sistema y para comunicar este comportamiento entre las personas involucradas en el sistema. También indica que los casos de uso son la entrada principal para el análisis, diseño, implementación y prueba del sistema, incluyendo la creación, verificación y validación del sistema.
- **Centrado en la Arquitectura.-** La arquitectura es una vista de diseño completo con las características más importantes resaltadas, dejando los detalles de lado. Debido a que lo que es significativo depende en parte de una valoración, que a su vez, se adquiere con la experiencia, el valor de una arquitectura depende de las personas que se hayan responsabilizado de su creación. No obstante, el proceso ayuda al arquitecto a centrarse en los objetivos adecuados, como la comprensión de la capacidad de adaptación al cambio y la reutilización.
- **Iterativo e Incremental.-** Es práctico dividir el trabajo en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo y los incrementos, al crecimiento del producto. Para una efectividad máxima, las iteraciones deben estar controladas; deben seleccionarse y ejecutarse de una forma planificada.

Algunas de las metas que persigue RUP son:⁶

- Asegurar la producción de un software de alta calidad que reúna las necesidades de los usuarios finales dentro de un plan y un presupuesto predecible.
- Proveer un enfoque disciplinario para asignar tareas y responsabilidades dentro del desarrollo del sistema.
- Proveer un camino metódico, sistemático para desarrollar, diseñar y validar una arquitectura.
- Reducir en gran medida el riesgo que representa la construcción de sistemas complejos, porque evoluciona de forma incremental partiendo de sistemas más pequeños en los que ya se tiene confianza.

El proceso puede ser descrito en dos dimensiones a través de dos ejes, donde:

- El eje horizontal representa el tiempo y muestra el aspecto dinámico del proceso, además es expresado en términos de ciclos, fases, iteraciones e hitos.
- El eje vertical representa el aspecto estático del proceso, descrito en términos de actividades, artefactos y flujos de trabajo.



⁶ Sistemas de Información II, pág. 17, http://prof.usb.ve/Imendoza/Documentos/PS-6116/Teoría_PS6116_O-O_y_RUP.pdf, 01/03/2010

2.3.1. FASES

Una **Fase** es el intervalo de tiempo entre dos hitos importantes del proceso de desarrollo, durante el cual se cumple un conjunto bien definido de objetivos. RUP divide un ciclo de desarrollo en cuatro fases consecutivas:

- **Fase de Inicio:** Se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos. Se define el alcance del proyecto.
- **Fase de Elaboración:** Se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- **Fase de Construcción:** Se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual del usuario.
- **Fase de Transición:** Proporcionar el sistema a sus usuarios finales, lo cual incluye: manufactura, envío, entrenamiento, soporte y mantenimiento del producto, hasta que el cliente esté satisfecho.⁷

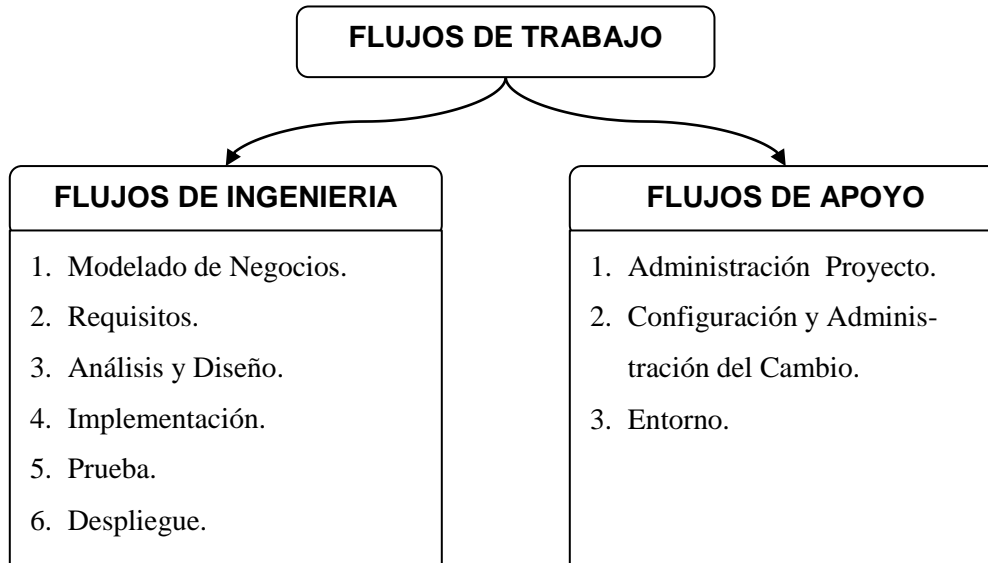
Dentro de cada fase hay varias iteraciones. Una **Iteración** representa un ciclo de desarrollo completo: desde la captura de requisitos hasta la implementación y pruebas, que además produce una versión (interna o externa) de un producto ejecutable.

2.3.2. FLUJOS DE TRABAJO

Son una secuencia de actividades que produce un resultado de valor observable. En UML un Flujo de Trabajo puede ser expresado como:

⁷ Sistemas de Información II, http://prof.usb.ve/Imendoza/Documentos/PS-6116/Teoría_PS6116_O-O_y_RUP.pdf, pág. 21, 03/03/2007

FIGURA N° 2
FLUJOS DE TRABAJO DE RUP



Fuente: Best Practices for Software development Teams ⁸

2.3.2.1. MODELADO DEL NEGOCIO

- Permite familiarizarse con el funcionamiento de la empresa, conocer sus procesos.
- Entender la estructura y la dinámica de la organización para la cual el sistema va a ser desarrollado.
- Entender el problema actual en la organización objetivo e identificar potenciales mejoras.
- Asegurar que clientes, usuarios finales y desarrolladores tengan un entendimiento común de la organización objetivo.

⁸ Rational Unified Process: Best Practices for Software development Teams, pág. 10, <http://www.ibm.com/developerworks/rational/library/253.html> , 10/03/2010.

2.3.2.2. REQUISITOS

El objetivo es describir **qué** debe hacer el sistema y permitir a los desarrolladores y clientes estar de acuerdo con esa descripción.

- Identificar **Actores** (usuarios) y otros sistemas que interactúan con el sistema.
- Identificar **casos de uso** representando el comportamiento del sistema.
- Describir detalladamente cada caso de uso, mostrando cómo el sistema interactúa paso a paso con los actores y lo que hace.

2.3.2.3. ANÁLISIS Y DISEÑO

Se especifican los requerimientos y se describen cómo se van a implementar en el sistema:

- Transformar los requisitos al diseño del sistema.
- Desarrollar una arquitectura para el sistema.
- Adaptar el diseño para que sea consistente con el entorno de implementación.

2.3.2.4. IMPLEMENTACIÓN

- Definir la organización del código, en términos de subsistemas de implementación organizados en capas (layers).
- Implementar clases y objetos en componentes.
- Probar los componentes y unidades.
- Integrar los productos.

2.3.2.5. PRUEBA

- Verificar la interacción entre objetos.
- Verificar la integración correcta de todos los componentes software.
- Verificar que todos los requerimientos hayan sido implementados correctamente.

2.3.2.6. DESPLIEGUE

Tiene como objetivo producir con éxito distribuciones del producto y distribuirlo a los usuarios. Las actividades implicadas incluyen:

- Probar el producto en su entorno de ejecución final.
- Empaquetar el software para su distribución.
- Distribuir el software.
- Instalar el software.
- Proveer asistencia y ayuda a los usuarios.
- Migrar el software existente o convertir la base de datos.

2.4. UML: LENGUAJE UNIFICADO DE MODELADO

Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia basada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML incluye conceptos semánticos, notación y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas iterativas de modelado visual que tengan modeladores de código así como generadores de informes. La especificación de UML no define un proceso estándar pero está pensado para ser útil en un proceso iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos.

UML consta de los siguientes diagramas:

- Diagrama de Casos de Uso ⁹
- Diagrama de Clase (Incluyendo Diagrama de Objetos)
 - Diagrama de Estados
 - Diagrama de Actividad
 - Diagramas de Interacción
 - Diagrama de Secuencia
 - Diagrama de Colaboración
- Diagramas de Implementación
 - Diagrama de Componentes
 - Diagrama de Despliegue

2.4.1. Diagrama de Casos de Uso

Un diagrama de casos de uso es una vista gráfica de algunos o todos los actores, casos de uso y sus interacciones, identificados para un sistema. Cada sistema típicamente tiene un diagrama de Caso de Uso Principal, el cual es la imagen de las fronteras del sistema (actores) y la funcionalidad principal proporcionada por el sistema (casos de uso). Otros diagramas de caso de uso pueden ser creados cuando sea necesario.

2.4.2. Diagrama de Clases

Un diagrama de clases es un tipo de modelo estático. Un diagrama de clases describe la vista estática del sistema. Aunque tiene similitudes con un modelo de datos (entidad-relación), las clases no solo muestran la estructura de la información, sino que describen también el comportamiento. Un propósito de los diagramas de clases es definir una base para otros diagramas donde otros aspectos del sistema son mostrados (tales como los estados de los objetos o la colaboración entre ellos mostrados en los

⁹ Es una técnica para capturar información, no pertenece estrictamente al enfoque orientado a objetos.

diagramas dinámicos). Una clase en un diagrama de clase puede ser directamente implementada en un lenguaje de programación orientado a objetos.

A medida que más y más clases son añadidas al modelo, una representación textual de las clases no es suficiente. Los diagramas de clases son creados para proporcionar una imagen o vista de algunas o todas las clases en el modelo.

El diagrama de clases principal en la vista lógica del modelo es típicamente una imagen de los paquetes del sistema (a veces a este diagrama se le llama diagrama de paquetes). Cada paquete también tiene su diagrama de clases principal, que típicamente despliega las clases públicas del paquete. Otros diagramas se crean según sea necesario. Algunos usos típicos de otros diagramas son:

- Vista de todas las clases de implementación en un paquete.
- Vista de la estructura y comportamiento de una o más clases.
- Vista de una jerarquía de herencia.

Los diagramas de clases también pueden ser creados en la vista de casos de uso del modelo. Estos diagramas típicamente son asignados a los casos de uso y contienen una vista de las clases que participan en los casos de uso.

2.4.3. Diagrama de Objetos

Un diagrama de objetos es una variante del diagrama de clases y utiliza casi la misma notación. La diferencia entre los dos es que el diagrama de objetos muestra una serie de objetos (instancias de las clases), en vez de las clases actuales. Un diagrama de objetos es entonces un ejemplo de un diagrama de clases que muestra una posible imagen de la ejecución del sistema (cómo se vería el sistema en algún punto dado de tiempo). Es utilizada la misma notación que para los diagramas de clases, con dos excepciones: los nombres de los objetos son subrayados y son mostradas todas las instancias en una relación.

Los diagramas de objetos no son tan importantes como los diagramas de clases, pero pueden ser utilizados para ejemplificar un diagrama de clases complejo mostrando cómo se verían las instancias actuales y las relaciones. Los diagramas de objetos son también utilizados como parte de los diagramas de colaboración, en los cuales es mostrada la colaboración dinámica entre los objetos.

2.4.4. Diagrama de Estados

Un diagrama de estados es típicamente un complemento de la descripción de una clase. Muestra todos los estados posibles que los objetos de la clase puedan tener, y qué eventos causan un cambio de estado. Un evento puede ser otro objeto que envía un mensaje – por ejemplo, que el tiempo especificado se ha terminado – o que alguna otra condición ha sido cumplida. Un cambio de estado es llamado transición. Una transición puede tener también una acción conectada a él para especificar qué sería hecho en conexión con el estado de transición.

Los diagramas de estados no son dibujados para todas las clases, solamente para aquellas que tienen una serie de estados bien definidos y en donde el comportamiento de la clase es afectado y cambiado por los estados diferentes. Los diagramas de estados pueden también ser dibujados para el sistema en su totalidad.

2.4.5. Diagrama de Secuencia

Un diagrama de secuencia muestra una colaboración dinámica entre una serie de objetos. El aspecto importante de este diagrama es mostrar una secuencia de mensajes enviados entre los objetos.

También son mostradas las interacciones entre los objetos, algo que sucederá en un punto específico de la ejecución de un sistema. Los diagramas consisten en una serie de objetos mostrados con líneas verticales. El tiempo pasa descendentemente en el diagrama, y el

diagrama muestra el intercambio de mensajes entre los objetos a medida que pasa el tiempo en la secuencia o función. Los mensajes son mostrados como líneas con flechas de mensajes entre las líneas verticales de los objetos. Las especificaciones de tiempo y otros comentarios son añadidos en una escritura en el margen del diagrama.

2.4.6. Diagrama de Colaboración

Un diagrama de colaboración muestra una colaboración dinámica, como el diagrama de secuencia. Es a menudo una elección mostrar una colaboración ya sea con un diagrama de secuencia o un diagrama de colaboración. Además de mostrar el intercambio de mensajes (llamado interacción), el diagrama de colaboración muestra los objetos y sus relaciones (a veces referidos como el contexto). A menudo uno puede decidir si utilizar un diagrama de secuencia o un diagrama de colaboración: si el tiempo o la secuencia es el aspecto más importante a enfatizar, escoja un diagrama de secuencia; si es importante enfatizar el contexto, escoja un diagrama de colaboración. La interacción entre los objetos es mostrada en ambos diagramas.

El diagrama de colaboración es dibujado como un diagrama de objetos, donde una serie de objetos son mostrados junto con sus relaciones (utilizando la notación en el diagrama de clases o de objetos). Las flechas de mensajes son dibujadas entre los objetos para mostrar el flujo de mensajes entre los objetos.

Se ponen etiquetas en los mensajes, lo cual entre otras cosas, muestra el orden en el cual son enviados los mensajes. También pueden mostrarse las condiciones, iteraciones, valores de retorno, y así sucesivamente. Cuando está familiarizado con la sintaxis de etiquetas para los mensajes, el desarrollador puede leer la colaboración y seguir el flujo de ejecución y el intercambio de mensajes.

Un diagrama de colaboración puede también contener objetos activos que se ejecutan concurrentemente con otros objetos activos.

2.4.7. Diagrama de Actividades

Un diagrama de actividades muestra el flujo secuencial de las actividades. El diagrama de actividades es utilizado típicamente para describir las actividades realizadas en una operación, aunque puede ser también utilizado para describir otros diagramas, tal como un caso de uso o de interacción. El diagrama de actividades consiste de estados de acción, los cuales contienen una especificación de la actividad que va a ser realizada (una acción). Un estado de acción termina cuando ha sido realizada la acción (un estado en un diagrama de estados necesita un evento explícito aún antes que deje el estado).

Por lo tanto, el control fluye entre los estados, que están conectados entre sí. Las decisiones y las condiciones, así como la ejecución en paralelo de los estados de acción, pueden ser también ser mostrados en el diagrama. El diagrama puede también tener especificaciones de los mensajes que han sido enviados o recibidos como parte de las acciones realizadas.

2.4.8. Diagrama de Componentes

Un diagrama de componentes muestra la estructura física del código en términos de los componentes de código. Un componente puede ser un componente de código fuente, un componente binario, o un componente ejecutable. Un componente contiene información sobre la clase lógica o las clases que implementa, creando un mapeo de la vista lógica a la vista de componentes. Las dependencias entre los componentes son mostradas, haciendo fácil de analizar cómo los otros componentes son afectados por un cambio en uno de los componentes. Los componentes pueden ser mostrados también con cualquiera de las interfaces que exponen, tal como las interfaces OLE/COM y pueden ser agrupados en paquetes. El diagrama de componentes es utilizado en trabajos prácticos de programación.

2.4.9. Diagrama de Despliegue

El diagrama de despliegue muestra la arquitectura física del hardware y el software en el sistema. Se pueden mostrar las computadoras y los dispositivos (nodos), junto con las conexiones que tienen unos con otros; también se puede mostrar el tipo de conexión. Dentro de los nodos, los componentes ejecutables y objetos son localizados para mostrar qué unidades de software son ejecutadas y en qué nodos. Además se muestran las dependencias entre los componentes.

Como se dijo previamente, el diagrama de despliegue muestra la vista de despliegue la cual describe la arquitectura física actual del sistema. Esto está lejos de la descripción funcional en la vista de casos de uso. Sin embargo, con un modelo bien definido, es posible navegar todo el camino desde un nodo en la arquitectura física a sus componentes, a las clases que implementa, a las interacciones de los objetos de la clase en la cual participan y finalmente al caso de uso. Las diferentes vistas del sistema son utilizadas para dar una descripción coherente del sistema como un todo.

2.5. MODELO DE DATOS

2.5.1. Modelo Entidad Relación

El modelo Entidad Relación (E-R) está basado en una percepción del mundo real consistente en objetos básicos llamados entidades y de relación entre estos objetos. Se desarrolló para facilitar el diseño de base de datos permitiendo la especificación de un esquema de la empresa que representa la estructura lógica completa de una base de datos. El modelo E-R es extremadamente útil para hacer corresponder los significados e interacciones de las empresas del mundo real con un esquema conceptual. Debido a esta utilidad, muchas herramientas de diseño de bases de datos se basan en los conceptos del modelo E-R.

2.5.2. Base de Datos Relacionales (El Modelo Relacional)

En una Base de Datos relacional los datos se encuentran organizados y almacenados en tablas llamadas afinidades, cada una de las cuales es implementada como un registro. Cada tabla del modelo relacional es en dos dimensiones: filas y columnas, cada fila contiene datos que pertenecen a alguna cosa o parte de alguna cosa y cada columna contiene un atributo.

El objetivo del diseño de las bases de datos relacionales es la generación de un conjunto de esquemas relacionales que nos permita almacenar la información sin redundancia innecesaria, pero también nos permitan recuperar fácilmente esa información. Un enfoque es el diseño de esquemas que se hallen en unas formas normales deseables.

2.5.3. Aplicaciones de Base de Datos Cliente/Servidor

Es un modelo para el desarrollo de Sistemas de Información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o **recursos**. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

2.5.4. El Modelo Cliente Servidor

Es el modelo de ejecución que siguen todas las aplicaciones de red. Un servidor es un proceso que se está ejecutando en algún nodo de la red, su función es gestionar el acceso a indeterminado recurso. Un cliente es un proceso que se ejecuta en el mismo nodo o en uno diferente y realiza peticiones al servidor.

2.6. PHP

PHP son las siglas "Personal Home Page". Es un lenguaje de programación pensado en el web de forma que es ideal para la creación de páginas dinámicas. PHP es la versión libre del sistema equivalente de Microsoft ASP.

PHP es un lenguaje encapsulado dentro de los documentos html (aunque los CGI hechos con PHP terminan en extensión .php y no .html). De forma que se pueden introducir instrucciones php dentro de las páginas. Gracias a esto el diseñador gráfico del web puede trabajar de forma independiente al programador. PHP es interpretado por el servidor (apache) generando un HTML con el resultado de substituir las secuencias de instrucciones PHP por su salida.

Por lo tanto una web dinámica con PHP contiene una serie de documentos php, que el servidor apache interpreta proporcionando al cliente documentos html con el resultado de las órdenes php.

2.6.1. Tareas principales de PHP

PHP se ha convertido en un lenguaje que permite hacer de todo. En un principio diseñado para realizar poco más que un contador y un libro de visitas, PHP ha experimentado en poco tiempo una verdadera revolución y, a partir de sus funciones, se pueden realizar una multitud de tareas útiles para el desarrollo del Web. Entre sus funciones podemos destacar:

- **Funciones de correo electrónico.** Podemos con una facilidad asombrosa enviar un e-mail a una persona o lista, parametrizando toda una serie de aspectos tales como el e-mail de procedencia, asunto, persona a responder. Otras funciones menos frecuentes pero de indudable utilidad para gestionar correos electrónicos son incluidas en su librería.

- **Gestión de bases de datos.** Resulta difícil concebir un sitio actual, potente y rico en contenido que no es gestionado por una base de datos. El lenguaje PHP ofrece interfaces para el acceso a la mayoría de las bases de datos comerciales y por ODBC a todas las bases de datos posibles en sistemas Microsoft, a partir de las cuales podremos editar el contenido de nuestro sitio con absoluta sencillez.
- **Gestión de archivos.** Crear, borrar, mover, modificar. Es decir, podremos hacer cualquier tipo de operación más o menos razonable que se nos pueda ocurrir a partir de una amplia librería de funciones para la gestión de archivos. También podemos transferir archivos por FTP a partir de sentencias en nuestro código, protocolo para el cual PHP ha previsto también gran cantidad de funciones.

Además de estas funciones, existen otras muchas más pensadas para Internet (tratamiento de cookies, accesos restringidos, comercio electrónico y otras), de propósito general (funciones matemáticas, explotación de cadenas, de fechas, corrección ortográfica, compresión de archivos y otras) o para el tratamiento de ficheros XML. A esta inmensa librería cabe ahora añadir todas las funciones personales que uno va creando por necesidades propias y que luego son reutilizadas en otros sitios y todas aquellas intercambiadas u obtenidas en foros o sitios especializados.

2.6.2. Ventajas de PHP

Entre las ventajas podemos citar:

- Muy sencillo de aprender.
- Similar en sintaxis a C y a PERL
- Soporta en cierta medida la orientación a objeto. Clases y herencia.

- El análisis léxico para recoger las variables que se pasan en la dirección lo hace PHP de forma automática. Librándose el usuario de tener que separar las variables y sus valores.
- Se puede incrustar código PHP con etiquetas HTML.
- Excelente soporte de acceso a base de datos.
- La comprobación de que los parámetros son válidos se hace en el servidor y no en el cliente (como se hace con javascript) de forma que se puede evitar que chequear que no se reciban solicitudes adulteradas. Además PHP viene equipado con un conjunto de funciones de seguridad que previenen la inserción de órdenes dentro de una solicitud de datos.
- Se puede hacer de todo lo que se pueda transmitir por vía HTTP.

2.7. POSTGRES

Los sistemas de mantenimiento de Bases de Datos relacionales tradicionales (DBMSs) soportan un modelo de datos que consisten en una colección de relaciones con nombre, que contienen atributos de un tipo específico. En los sistemas comerciales actuales, los tipos posibles incluyen numéricos de punto flotante, enteros, cadenas de caracteres, cantidades monetarias y fechas. Postgres ofrece una potencia adicional sustancial al incorporar los siguientes cuatro conceptos adicionales básicos en una vía en la que los usuarios pueden extender fácilmente el sistema

- Clases
- Herencia
- Tipos
- Funciones

Otras características aportan potencia y flexibilidad adicional:

- Restricciones (Constraints)
- Disparadores (triggers)
- Reglas (rules)
- Integridad transaccional

Estas características colocan a Postgres en la categoría de las Bases de Datos identificadas como *objeto-relacionales*. Nótese que éstas son diferentes de las referidas como *orientadas a objetos*, que en general no son bien aprovechables para soportar lenguajes de Bases de Datos relacionales tradicionales. Postgres tiene algunas características que son propias del mundo de las bases de datos orientadas a objetos. De hecho, algunas Bases de Datos comerciales han incorporado recientemente características en las que Postgres fue pionera.

Las principales mejoras en PostgreSQL incluyen:

- Los bloqueos de tabla han sido sustituidos por el control de concurrencia multi versión, el cual permite a los accesos de sólo lectura continuar leyendo datos consistentes durante la actualización de registros, y permite copias de seguridad en caliente desde pg dump mientras la base de datos permanece disponible para consultas.
- Se han implementado importantes características del motor de datos, incluyendo subconsultas, valores por defecto, restricciones a valores en los campos (constraints) y disparadores (triggers).
- Se han añadido funcionalidades en línea con el estándar SQL92, incluyendo claves primarias, identificadores entrecomillados, forzado de tipos cadena literales, conversión de tipos y entrada de enteros binarios y hexadecimales.
- Los tipos internos han sido mejorados, incluyendo nuevos tipos de fecha/hora de rango amplio y soporte para tipos geométricos adicionales.

- La velocidad del código del motor de datos ha sido incrementada aproximadamente en un 20-40%, y su tiempo de arranque ha bajado el 80% desde que la versión 6.0 fue lanzada.

2.7.1. PostgreSQL el motor de base de datos libres

Al igual que Linux, Apache y Sendmail son ejemplos de software libre (GPL o BSD) de excelente calidad y más usados que su competencia comercial, existe un Motor de Base de Datos Relacional y Orientado a Objetos libre, PostgreSQL.

PostgreSQL es un Motor de Base de Datos con todas las características y ventajas de un motor comercial. PostgreSQL soporta: SQL92, Transacciones, Triggers, Constrains, Foreigns Keys, Replicación, etc. Además soporta conectividad ODBC para ser utilizado con aplicaciones Windows y JDBC 2.0 para Java.

2.7.2. Arquitectura

PostgreSQL implementa una arquitectura cliente/servidor, lo que facilita su administración y configuración y mejora su desempeño.

En el servidor, PostgreSQL cuenta con un proceso postmaster, que se encarga de recibir las conexiones y cada vez que recibe una, crea una instancia del motor postgres, para procesar todos los requerimientos del cliente.

En el cliente, se cuenta con una interfaz de comandos llamada psql, que le permite al usuario ejecutar cualquier sentencia SQL, así como administrar la Base de Datos y las tablas. Además se cuenta con una interfaz gráfica, pgaccess, que permite realizar las funciones de administración, crear funciones, formas y reportes.

PostgreSQL cuenta con interfaces de programación para todos los lenguajes: C, C++, Perl, PHP, Python y Java.

PostgreSQL permite la integración con aplicaciones Windows, ya que soporta ODBC. Con ODBC se pueden desarrollar aplicaciones cliente Windows (VB, FoxPro, etc), que interactuen con un servidor Linux con PostgreSQL; además desde una base de datos Access se pueden importar tablas desde un servidor PostgreSQL.

2.8. SERVIDOR WEB APACHE

Es imposible recomendar el mejor servidor Web, que dependerá en gran medida de las necesidades concretas de cada caso. Sin embargo, lo más razonable es que en caso de emplear como servidor una máquina Unix la elección sea Apache, y si es una máquina con Windows Server, entonces Internet Information Server.

Hoy en día el servidor Apache es el número uno indiscutible del mercado. Esto se debe a su bajo coste (es gratuito), su excelente rendimiento y su gran flexibilidad.

Apache debe su nombre a su origen: consiste en una versión parcheada «<A .PatCHy sErver») del servidor de la NCSA. Este servidor es el producto más ampliamente conocido del Proyecto Apache que trata de construir un servidor http robusto, potente, disponible en todas las plataformas, gratuito y con disponibilidad de código fuente. Se diseñó originalmente para correr en máquinas Unix y sus múltiples variantes; sin embargo, hoy en día está disponible para prácticamente la totalidad de sistemas operativos.

2.8.1. Arquitectura del servidor Apache

El servidor Apache es un software que está estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías:

- **Módulos Base:** Módulo con las funciones básicas del Apache
- **Módulos Multiproceso:** son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones
- **Módulos Adicionales:** Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiproceso para cada uno de los sistemas operativos, sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código. El resto de funcionalidades del servidor se consiguen por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software.

2.9. HERRAMIENTAS CASE

Las herramientas CASE (Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan en el ciclo de vida de desarrollo del software, en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

En el presente trabajo se utilizó las siguientes herramientas CASE:

- **Enterprise Architect V7** para diseñar los diagramas de UML.
- **DeZign for Databases V6** para el diseño del modelo relacional.

2.9.1. ENTERPRISE ARCHITECT

Enterprise Architect¹⁰ combina el poder de la última especificación UML 2.0 con alto rendimiento, interfaz intuitiva, para traer modelado avanzado al escritorio, y para el equipo completo de desarrollo e implementación. Las características más importantes de Enterprise Architect son las siguientes:

➤ **Alta capacidad**

Enterprise Architect es una herramienta comprensible de diseño y análisis UML, cubriendo el desarrollo de software desde el paso de los requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento. Enterprise Architect es una herramienta multiusuario, basada en Windows, diseñada para ayudar a construir software robusto y fácil de mantener. Ofrece salida de documentación flexible y de alta calidad. El manual de usuario está disponible en línea.

➤ **Velocidad, estabilidad y buen rendimiento**

El Lenguaje Unificado de Modelado provee beneficios significativos para ayudar a construir modelos de sistemas de software rigurosos y donde es posible mantener la trazabilidad de manera consistente. Enterprise Architect soporta este proceso en un ambiente fácil de usar, rápido y flexible. Para una mirada rápida al modelado UML en Enterprise Architect vea nuestro tutorial UML y documentos.

➤ **Trazabilidad de extremo a extremo**

Enterprise Architect provee trazabilidad completa desde el análisis de requerimientos hasta los artefactos de análisis y diseño, a través de la implementación y el despliegue. Combinados con la ubicación de recursos y tareas incorporados, los equipos de Administradores de Proyectos y Calidad están equipados con la información que ellos necesitan para ayudarles a entregar proyectos en tiempo.

¹⁰ <http://www.sparxsystems.com.au/products/ea/index.html>

➤ **Construido sobre las bases de UML 2.**

Las bases de Enterprise Architect están construidas sobre la especificación de UML 2.0 - pero no se detiene ahí! Usa Perfiles UML para extender el dominio de modelado, mientras que la Validación del Modelo asegura integridad. Combina Procesos de Negocio, Información y Flujos de trabajo en un modelo usando extensiones gratuitas.

Además de tiene las siguientes características adicionales:

- Permite crear elementos del modelo UML para un amplio alcance de objetivos.
- Ubica esos elementos en diagramas y paquetes.
- Crea conectores entre elementos.
- Documenta los elementos que ha creado.
- Genera código para el software que se está construyendo en los siguientes manejadores de bases de datos: DB2, Interbase, Informix, Ingres, MS Access, MS SQL Server, Oracle, PostgreSQL, Sybase ASE, ASA y Firebird.
- Realiza ingeniería directa e inversa de código en: ActionScript, C++, C#, Delphi, Java, Python, PHP y VB.NET.

2.9.2. DEZIGN FOR DATABASES

Dezign for Databases¹¹ es una herramienta de diseño y mantenimiento para diseñadores y programadores de bases de datos. Dezign for Databases automatiza y facilita el trabajo a través del ciclo de desarrollo de bases de datos (diseño, documentación y generación de código).

Al emplear Dezign for Databases se puede construir un modelo lógico de los datos del negocio a través del Diagrama Entidad – Relación. Cada entidad está compuesta de atributos que la describen y las entidades están enlazadas por medio de relaciones.

¹¹ <http://www.datanamic.com/dezign/keyfeatures.html>

Además de un soporte completo para realizar el modelado del diagrama Entidad – Relación, DeSign for Databases también permite usar elementos físicos como variables y valores por defecto.

DeSign for Databases suporta las siguientes bases de datos:

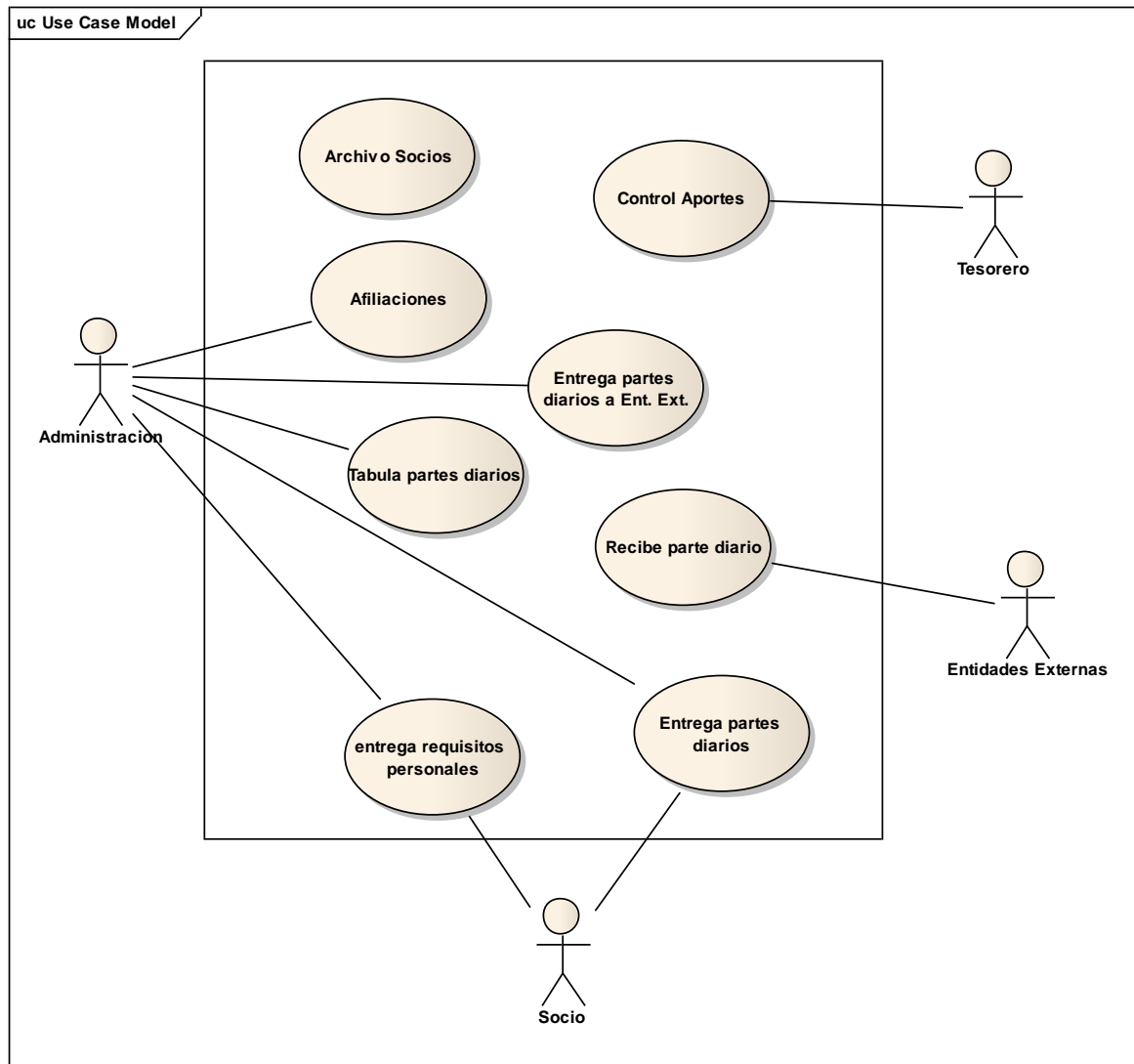
- dBase
- DBISAM 4
- Firebird 1.5, 2
- FoxPro
- IBM DB2 Universal DB 7, 8, 9
- InterBase 6, 7
- Informix 9
- MaxDB 7
- MS Access 97, 2000, 2003, 2007
- MS SQL Server 7, 2000, 2005, 2008
- MySQL 3, 4, 5
- Oracle 8, 9, 10, 11
- Paradox
- Pervasive v8, PSQL v9
- PostgreSQL 7, 8
- Sybase ASE 11, 12

CAPÍTULO III

ANÁLISIS DEL SISTEMA

3.1. CASOS DE USO

3.1.1. DIAGRAMA DE CONTEXTO



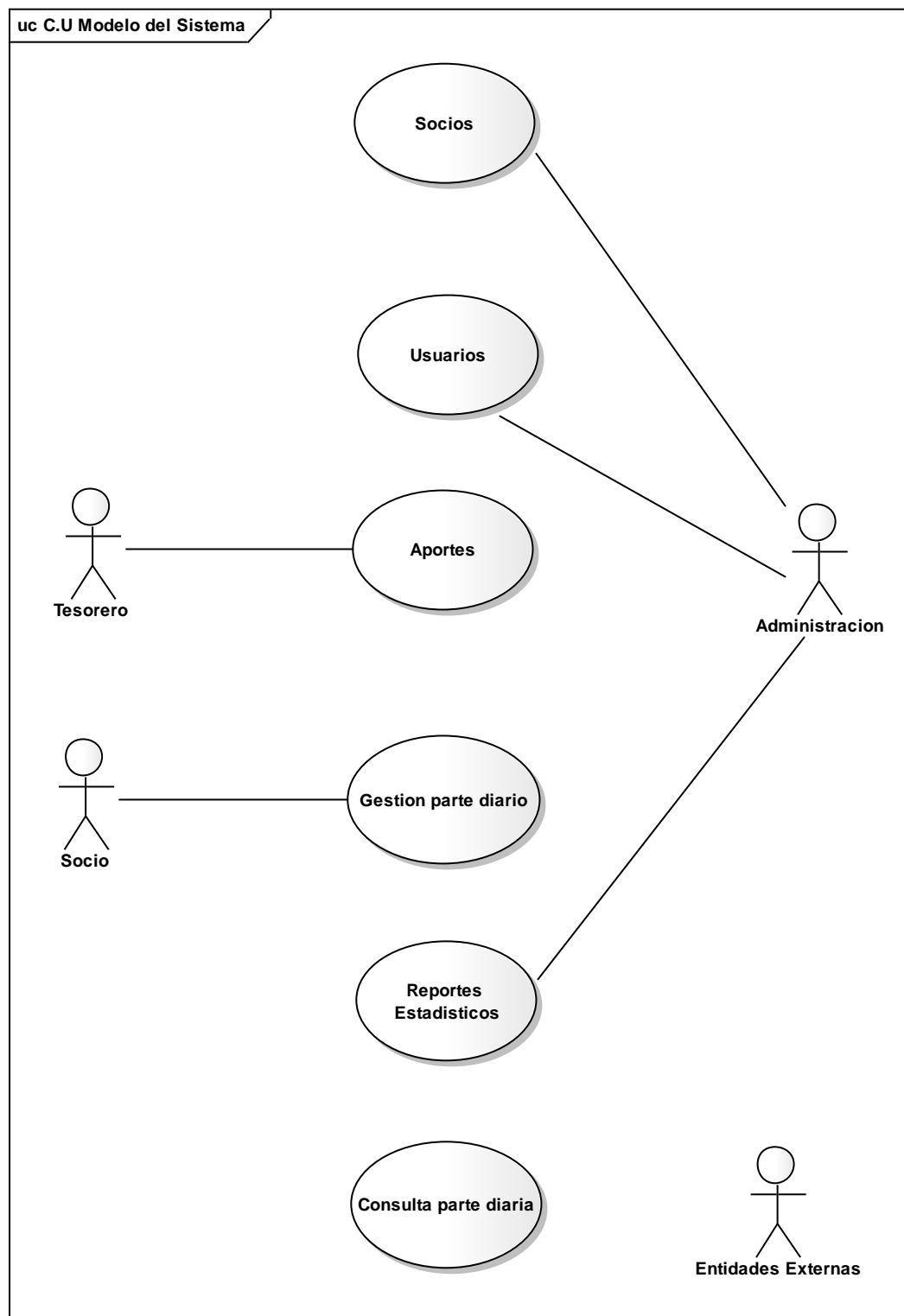
El sistema funciona cuando un establecimiento mediante su representante presenta los papeles de afiliación a la administración, luego la administración es encargada de realizar las afiliaciones, contenida en dos partes, una con datos propios del representante y otra con datos del establecimiento hotelero. Asimismo realiza el manejo de los archivos del socio, como ser ficha de afiliación, modificaciones, actualizaciones y bajas de la institución.

La recepción de partes diarias, previa recepción del informe del socio (partes diarias). El parte diario es un formulario donde se registra el movimiento de pasajeros entrantes, estantes y salientes de los diferentes establecimientos hoteleros.

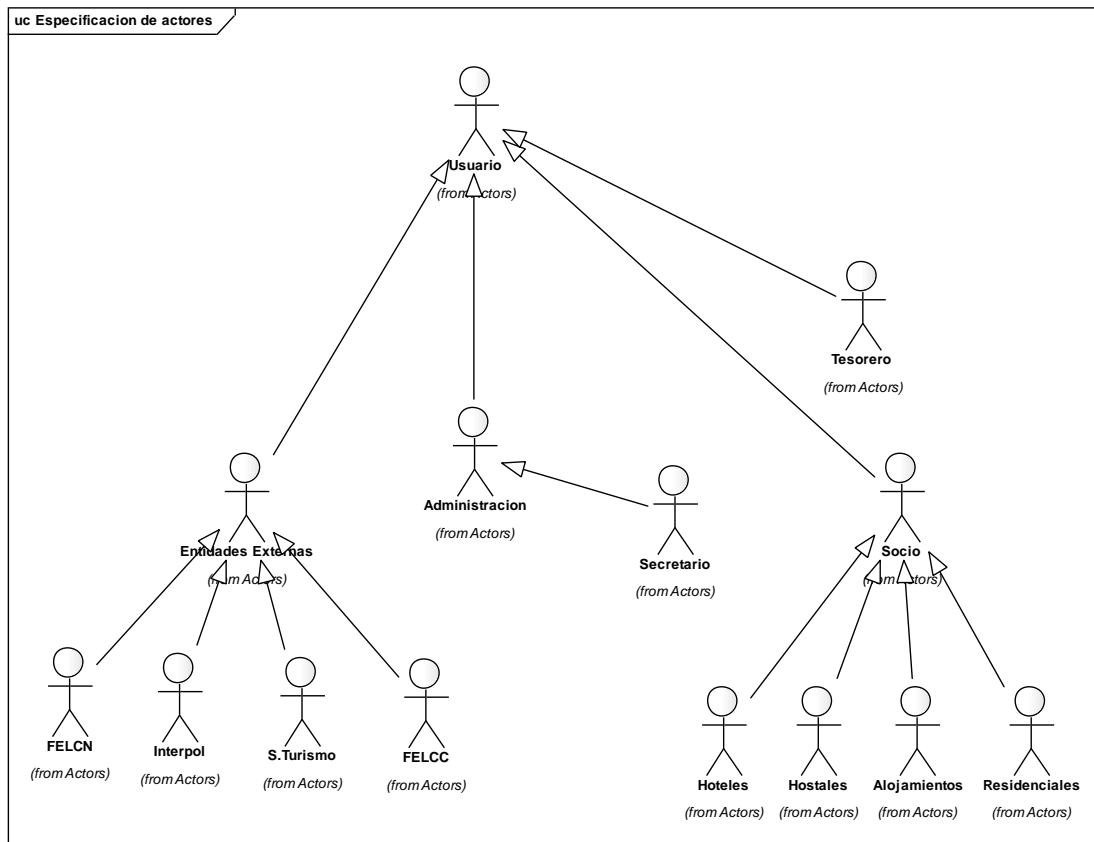
Una vez finalizado el reporte de parte diario se procede a la entrega de estos informes a entidades externas.

El registro del pago de los aportes mensuales se realiza emitiendo un recibo en forma manual, donde el original se entrega al establecimiento hotelero y la copia se guarda en un archivador. Los datos del pago realizado se registran en una planilla de control de cuotas mensuales.

3.1.2. DIAGRAMA DE CASOS DE USO DEL SISTEMA



3.1.3. ESPECIFICACIÓN DE ACTORES



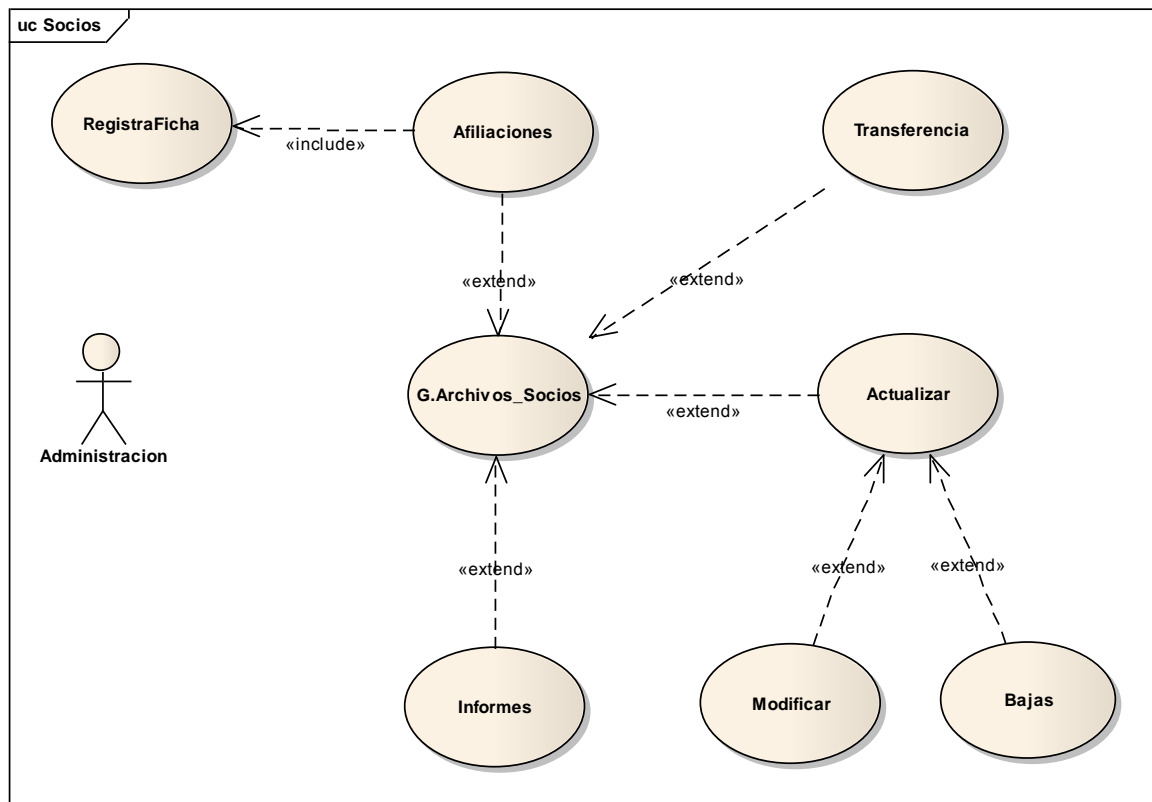
- **Administración.-** Actor que se encarga de gestionar los archivos del socio, como administrar su ficha de afiliación realizando actualizaciones y modificaciones. La administración realiza partes diarios mediante el tabulado de los datos extraídos de los informes que proporciona el asociado.
- **Tesorero.-** Este actor realiza la recepción y registro de los aportes de dinero que el asociado realiza, tiene un libro de control de aporte por categoría. Asimismo entrega un recibo de depósito al socio, realiza el control de aportes de cada socio apoyándose en simples planillas de control.
- **Socio.-** Actor que representa a hoteles, residenciales, hostales, alojamientos, este actor accede al sistema para poder realizar el registro de movimientos de

pasajeros de su institución, este registro se realiza en un formulario proporcionado por la Cámara Hotelera vía web.

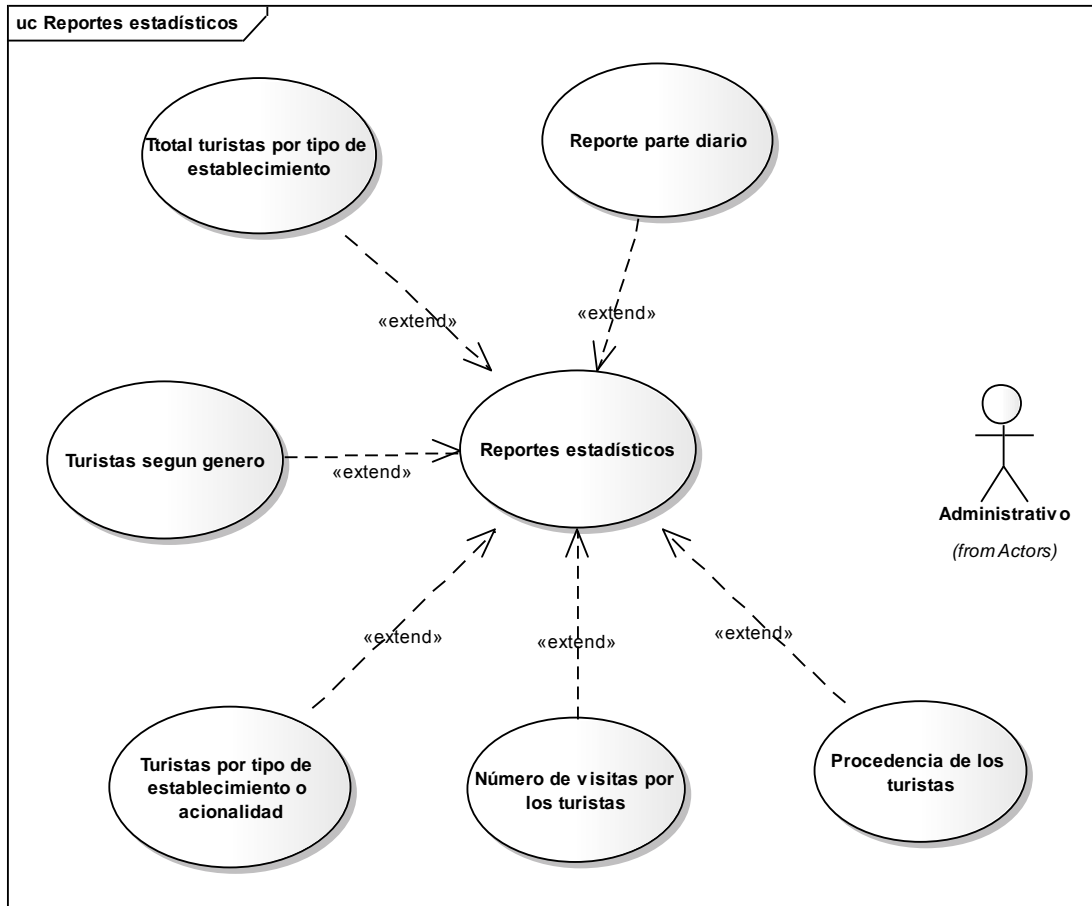
- **Entidades Externas.-** Este tipo de actor interactúa con el sistema accediendo al sistema para adquirir reportes del movimiento de pasajeros de cada uno de los asociados tabulados previamente por la administración. Esta información obtenida mediante reportes es importante para cada una de las instituciones de acuerdo a requerimiento.

3.1.4. DIAGRAMAS DE CASOS DE USO

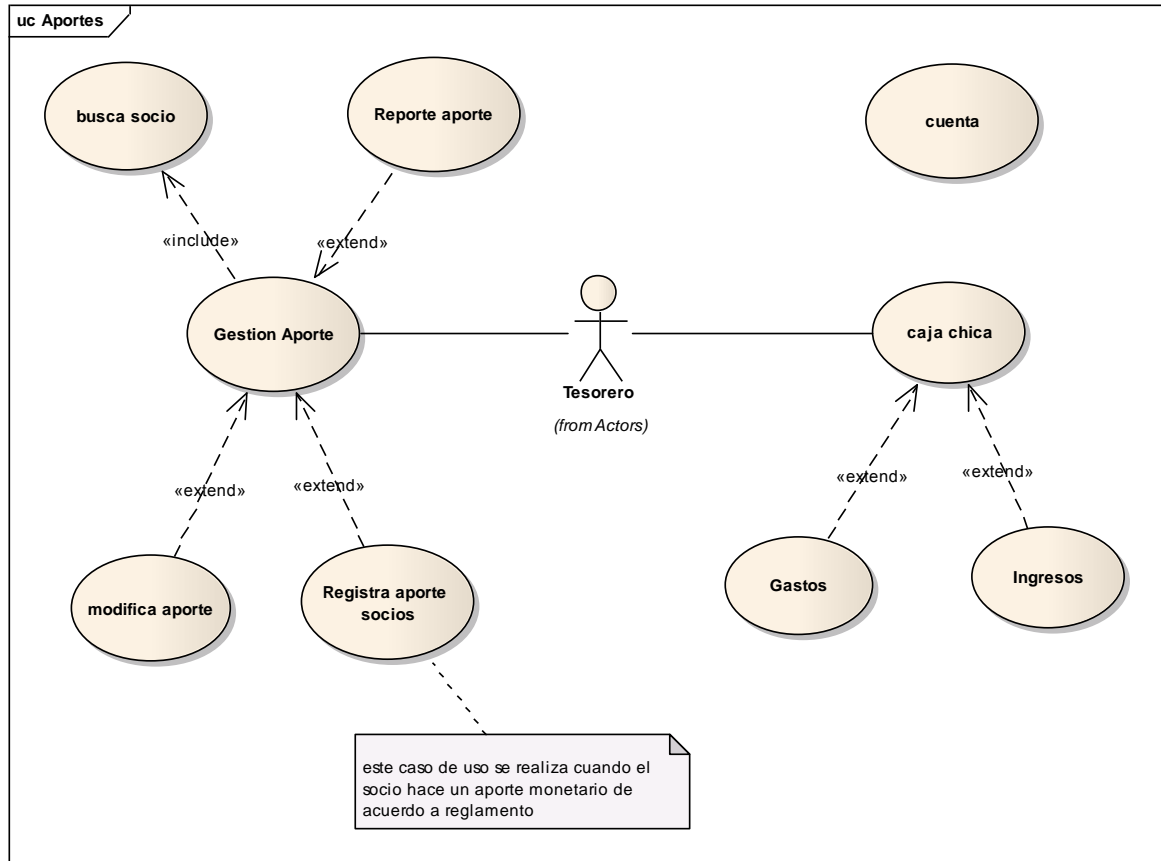
3.1.4.1. CASO DE USO: SOCIOS



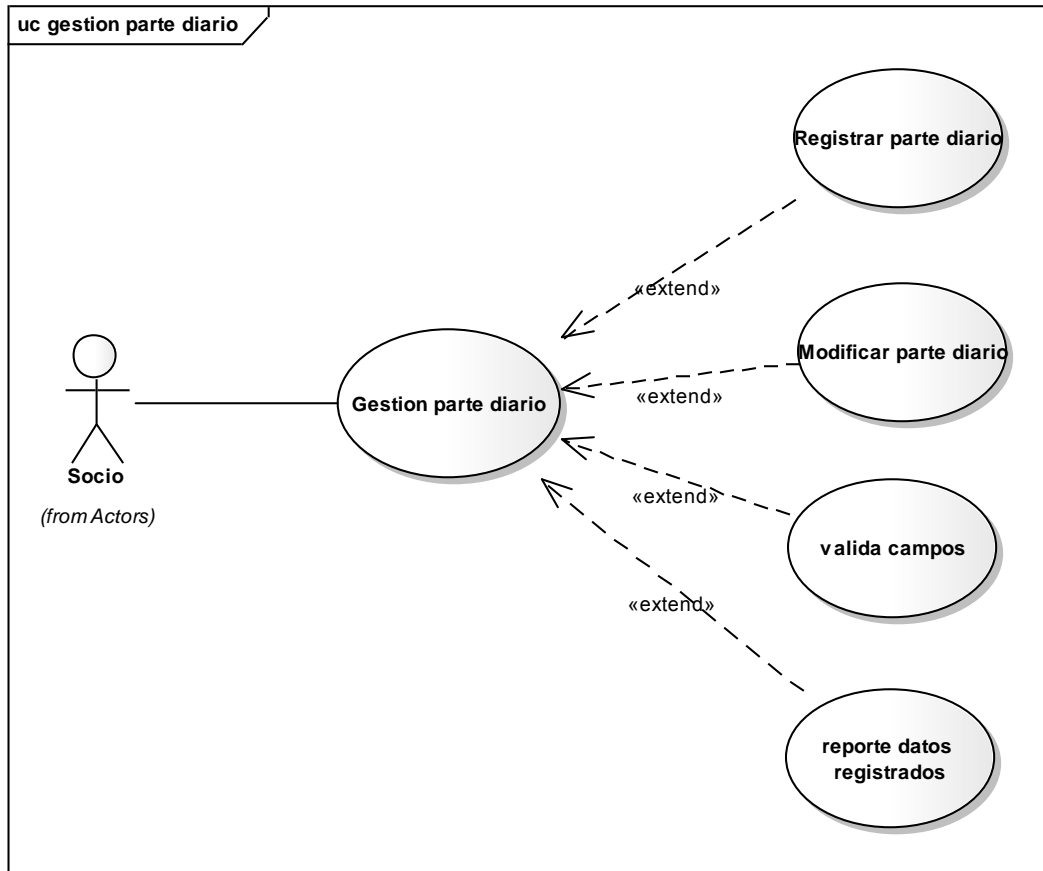
3.1.4.2. CASO DE USO: REPORTES ESTADISTICOS



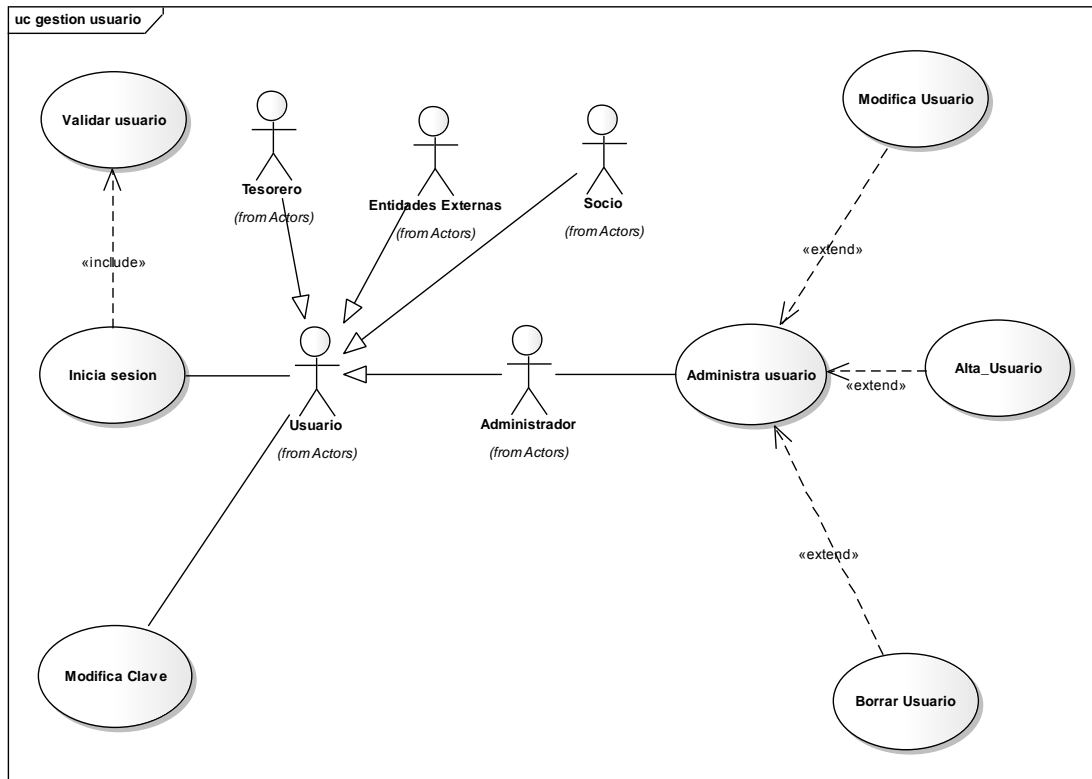
3.1.4.3. CASO DE USO: APORTES



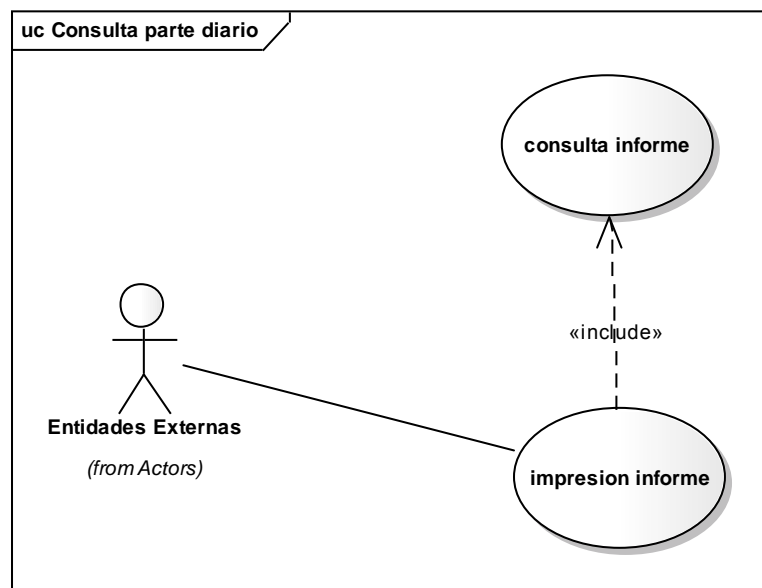
3.1.4.4. CASO DE USO: GESTION PARTE DIARIO



3.1.4.5. CASO DE USO: USUARIOS



3.1.4.6. CASO DE USO: CONSULTA PARTE DIARIA



3.2. ESPECIFICACION DE CASOS DE USO

3.2.1 CASO DE USO: SOCIOS

Caso de Uso:	G. Archivos Socios
Descripción:	Caso de uso donde la administración maneja los archivos de cada socio de acuerdo a requerimiento, como ser: registro de afiliaciones, transferencias, actualizaciones, bajas e informes.
Actores:	Administración, por medio del Presidente de la Cámara Hotelera o un funcionario, como el Secretario.
Precondición:	En caso de nuevas afiliaciones o cuando el socio presente modificaciones a su registro.
Flujo Normal:	<ol style="list-style-type: none"> 1.- El usuario selecciona gestión socios. 2.- El sistema presenta la pantalla de socios. 3.- El usuario elige una de las opciones presentadas: afiliaciones, transferencias, consultas y reportes.
Flujo Alternativo:	
Post condición:	Cierra sesión y realiza el grabado de datos.

Caso de Uso:	Afiliaciones
Descripción:	Se realiza las afiliaciones a nuevos socios a la Cámara Hotelera.
Actores:	Administración o secretario.
Precondición:	Que el postulante a socio presente todo los documentos requeridos por la institución.
Flujo Normal:	<ol style="list-style-type: none"> 1.- El usuario selecciona gestión socios. 2.- El sistema presenta la pantalla de socios. 3.- El usuario selecciona afiliaciones. 4.- Despliegue de pantalla afiliaciones.
Flujo Alternativo:	
Post condición:	

Caso de Uso:	Registra Ficha
Descripción:	Se registra en un formulario los datos personales del nuevo socio.
Actores:	Administración o secretario.
Precondición:	Que el postulante a socio presente todos los documentos requeridos por la institución.
Flujo Normal:	<ol style="list-style-type: none"> 1.- El usuario selecciona gestión socios. 2.- El sistema presenta la pantalla de socios. 3.- El usuario selecciona afiliaciones. 4.- Despliegue de pantalla afiliaciones. 5.- El usuario introduce datos de afiliación según formulario. 6.- Introduce datos propios del nuevo socio. 7.- Introduce datos de funcionamiento del nuevo socio. 8.- El sistema realiza la validación de los datos introducidos.
Flujo Alternativo:	Datos erróneos despliega pantalla afiliaciones.
Post condición:	Se realiza el grabado de afiliación de nuevo socio.

Caso de Uso:	Actualizar
Descripción:	Actualizamos el archivo del socio en base a su ficha de afiliación.
Actores:	Administración o secretario.
Precondición:	Que exista una solicitud de modificación a su archivo del socio.
Flujo Normal:	<ol style="list-style-type: none"> 1.- El usuario accede a la pantalla ficha socios. 2.- El sistema despliega el formulario de afiliación. 3.- El usuario gestiona en base a los requerimientos del socio.
Flujo Alternativo:	
Post condición:	Datos actualizados el sistema realiza el guardado de los datos almacenados.

Caso de Uso:	Modificar
Descripción:	Modifica los datos de afiliación del socio, datos propios o de funcionamiento.
Actores:	Administración o secretario.
Precondición:	Solicitud de modificación de ficha.
Flujo Normal:	<ol style="list-style-type: none"> 1.- El usuario selecciona gestión socios. 2.- El sistema presenta la pantalla de socios. 3.- El usuario selecciona afiliaciones. 4.- Despliegue de pantalla afiliaciones. 5.- Busca mediante Id al socio a modificar datos. 6.- El sistema despliega ficha del socio. 7.- Modifica datos. 8.- El sistema realiza la validación de los datos introducidos.
Flujo Alternativo:	Datos erróneos despliega pantalla afiliaciones.
Post condición:	Se realiza el grabado de modificación de nuevos datos del socio.

Caso de Uso:	Bajas
Descripción:	Se procede al borrado de registro de afiliación del socio.
Actores:	Administración o secretario.
Precondición:	El socio presenta una solicitud de alejamiento de la institución por un determinado motivo.
Flujo Normal:	<ol style="list-style-type: none"> 1.- El usuario selecciona gestión socios. 2.- El sistema presenta la pantalla de socios. 3.- El usuario selecciona afiliaciones. 4.- Despliegue de pantalla afiliaciones. 5.- El usuario busca mediante id al socio. 6.- El sistema despliega la pantalla ficha socio. 7.- El usuario procede a dar de baja al socio. 8.- El sistema confirma la eliminación de ese registro.
Flujo Alternativo:	Ficha no encontrada despliega la pantalla afiliaciones.
Post condición:	Borrado realizado.

3.2.2. CASO DE USO: APORTES

Caso de Uso:	Gestión Aporte
Descripción:	Manipula los datos referentes al aporte de cada socio.
Actores:	Tesorero
Precondición:	Que exista un aporte por parte del socio.
Flujo Normal:	<ol style="list-style-type: none"> 1.- El socio realiza un aporte de monto de dinero. 2.- El usuario selecciona la opción aportes. 3.- El sistema visualiza la pantalla aportes.
Flujo Alternativo:	
Post condición:	

Caso de Uso:	Registra aporte socios
Descripción:	Realiza el registro del aporte del socio
Actores:	Tesorero
Precondición:	Que exista un aporte por parte del socio
Flujo Normal:	<ol style="list-style-type: none"> 1.- El socio realiza un aporte de monto de dinero. 2.- El usuario selecciona la opción aportes. 3.- El sistema visualiza la pantalla aportes. 4.- El usuario busca el id del socio en su planilla de aportes. 5.- El sistema despliega la pantalla planilla de aportes del socio 6.- El usuario realiza el registro en el formulario. 7.- El sistema valida los datos. 8.- El sistema realiza el registro de datos del nuevo aporte. 9.- El sistema actualiza la ficha de control de aporte.
Flujo Alternativo:	Datos incorrectos al flujo 5.
Post condición:	Registro realizado datos nuevos

Caso de Uso:	Caja chica
Descripción:	Registra las operaciones realizadas en la institución como ser ingresos, los gastos realizados y otros.
Actores:	Tesorero
Precondición:	Que exista una operación ya sea ingreso o gasto.
Flujo Normal:	<ol style="list-style-type: none"> 1.- La administración realiza alguna operación económica como un egreso o un ingreso. 2.- El usuario selecciona la opción registro operaciones. 3.- El sistema visualiza la pantalla caja chica. 4.- El usuario realiza el registro de dicha operación.
Flujo Alternativo:	No exista ninguna operación.
Post condición:	

Caso de Uso:	Gastos.
Descripción:	Registra todos los gastos realizados por la institución.
Actores:	Tesorero.
Precondición:	Que exista un movimiento económico que represente un gasto para la institución.
Flujo Normal:	<ol style="list-style-type: none"> 1.- El usuario selecciona la opción registro de operaciones. 2.- El sistema despliega la pantalla registro de operaciones. 3.- El usuario selecciona registrar gastos. 4.- El sistema despliega la pantalla registrar gasto. 5.- El usuario introduce los datos de la factura representada en gasto.
Flujo Alternativo:	
Post condición:	Registro realizado

Caso de Uso:	Ingresos.
Descripción:	Registra todos los ingresos realizados por la institución.
Actores:	Tesorero.
Precondición:	Que exista un movimiento económico que represente un ingreso para la institución.
Flujo Normal:	<ol style="list-style-type: none"> 1.- El usuario selecciona la opción registro de operaciones. 2.- El sistema despliega la pantalla registro de operaciones. 3.- El usuario selecciona registrar ingresos. 4.- El sistema despliega la pantalla registrar ingresos. 5.- El usuario introduce los datos de la factura representada en un ingreso.
Flujo Alternativo:	
Post condición:	Registro realizado.

Caso de Uso:	Reporte de aportes.
Descripción:	Realiza reportes del movimiento económico de la institución.
Actores:	Tesorero.
Precondición:	Que exista datos almacenados en las tablas de la base de datos.
Flujo Normal:	<ol style="list-style-type: none"> 1.- El usuario selecciona la opción reportes. 2.- El sistema despliega la pantalla selección tipo de reporte. 3.- El usuario selecciona un tipo de reporte. 4.- El sistema genera el reporte. 5.- El sistema presenta la opción de impresión de reporte.
Flujo Alternativo:	Si no existen datos para un determinado reporte el sistema envía un mensaje de datos no encontrados.
Post condición:	Opción de impresión por parte del usuario.

3.2.3. CASO DE USO: REPORTES ESTADISTICOS

Caso de Uso:	Reporte Cuadros Estadísticos
Descripción:	Se procede a la generación de reportes de tipo estadístico presentando graficas demostrativas al usuario del movimiento de pasajeros de los establecimientos.
Actores:	Administración
Precondición:	Que exista los datos actualizados en una base de datos para poder realizar las consultas y generar los reportes.
Flujo Normal:	<ol style="list-style-type: none"> 1.- El usuario selecciona pantalla reportes estadísticos 2.- El sistema despliega la pantalla reportes estadísticos 3.- El usuario procede a realizar el parametrizado de los datos 4.- El sistema la consulta mediante los parámetros introducido por el usuario. 5.- El usuario selecciona generación de partes diarios. 6.- El sistema realiza la generación internamente. 7.- El sistema presenta la tabla de parte diario. 8.- El usuario realiza la carga del parte diario al sistema.
Flujo Alternativo:	
Post condición:	Carga del parte diario al sistema.

Caso de Uso:	Reportes parte diario
Descripción:	Se procede a la generación de reportes de los partes diarios generales de los establecimientos, mostrando al usuario del movimiento de pasajeros de los establecimientos.
Actores:	Administración
Precondición:	Que exista los datos actualizados en una base de datos para poder realizar las consultas y generar los reportes.
Flujo Normal:	<ol style="list-style-type: none"> 1.- El usuario selecciona pantalla reportes estadísticos 2.- El sistema despliega la pantalla reportes estadísticos 3.- El usuario procede a realizar el parametrizado de los datos 4.- El sistema la consulta mediante los parámetros introducido

	<p>por el usuario.</p> <p>5.- El usuario selecciona generación de partes diarios.</p> <p>6.- El sistema realiza la generación internamente.</p> <p>7.- El sistema presenta la tabla de parte diario.</p> <p>8.- El usuario realiza la carga del parte diario al sistema.</p>
Flujo Alternativo:	
Post condición:	Visualiza Muestra del parte diario.

3.2.4. CASO DE USO: USUARIOS

Caso de Uso:	Iniciar sesión
Descripción:	Este caso de uso permite al usuario ingresar su nombre de usuario y su contraseña para que de esta manera pueda empezar a operar el sistema de acuerdo a los privilegios que tenga asignado.
Actores:	Administración, tesorero, entidades externas y socio.
Precondición:	Que el sistema visualice la pantalla de acceso.
Flujo Normal:	<ol style="list-style-type: none"> 1. Se inicia cuando el actor introduce el nombre de usuario y contraseña. 2. El actor selecciona la opción Iniciar Sesión. 3. El sistema autentifica al usuario validando los datos introducidos (nombre de usuario y contraseña). 4. El sistema despliega la Pantalla Principal de la sesión del usuario.
Flujo Alternativo:	Si se ha desplegado un mensaje de error el actor selecciona Aceptar y vuelve al flujo 1 (en caso de introducir los datos del nombre de usuario o contraseña incorrectos).
Post condición:	El usuario ha ingresado al sistema, puede operar el sistema o elegir cerrar Sesión para salir del sistema.

Caso de Uso:	Administra usuario
Descripción:	Permite administrar los datos de los usuarios, realizar altas, bajas y modificaciones de usuarios.
Actores:	Administración.
Precondición:	Que el usuario haya iniciado sesión en el sistema.
Flujo Normal:	<ol style="list-style-type: none"> 1. El Usuario selecciona la opción administrar usuarios. 2. El sistema despliega la pantalla de administrar usuarios. 3. El usuario puede visualizar, registrar o modificar los datos de administrar personas. 4. En caso de que los datos fuesen incorrectos el sistema muestra un mensaje de error y regresa al flujo 2.
Flujo Alternativo:	
Post condición:	Datos nuevos o modificados.

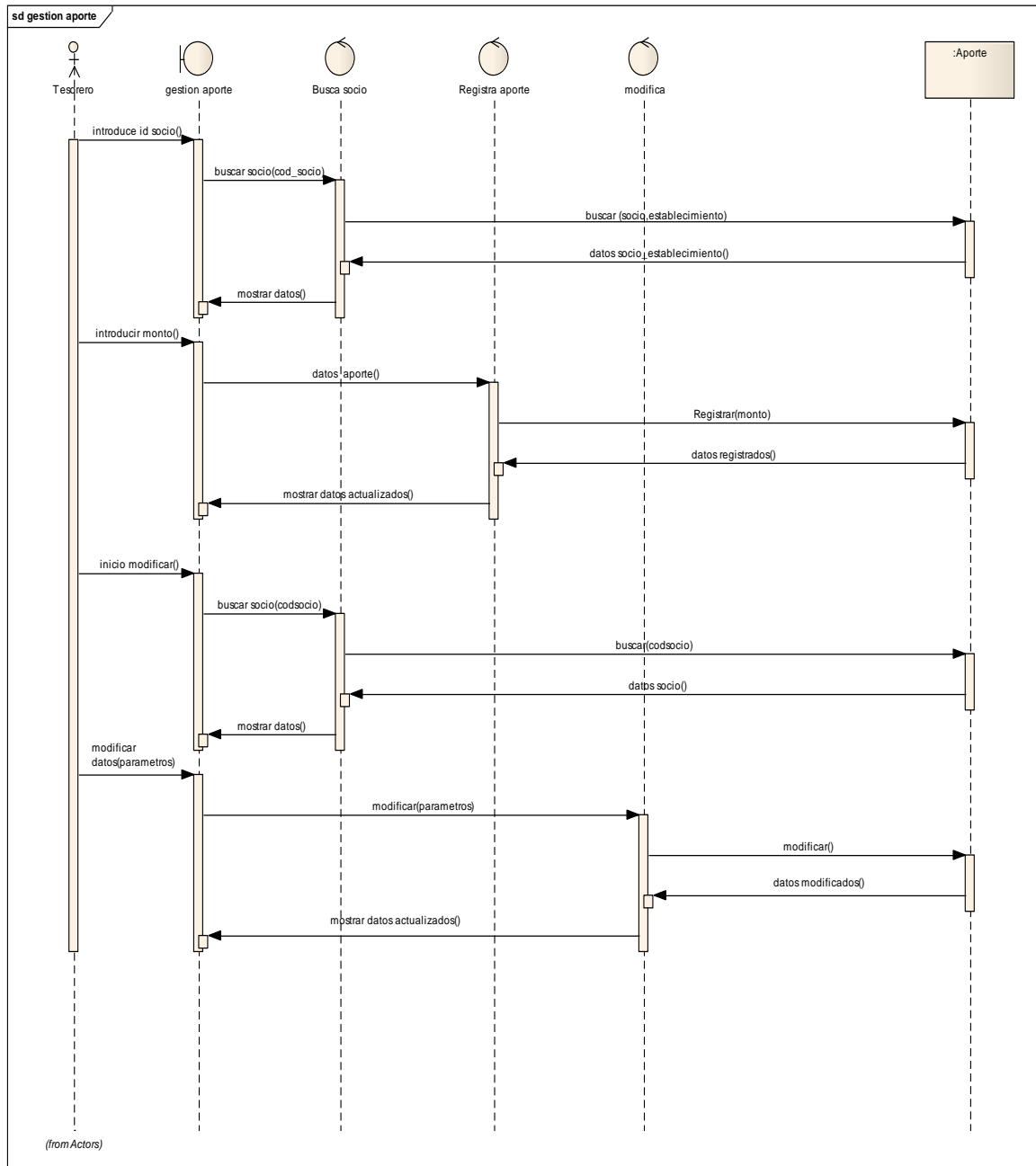
Caso de Uso:	Modificar clave
Descripción:	Este caso de uso permite cambiar la contraseña del usuario.
Actores:	Administración, tesorero, entidades externas y socio.
Precondición:	Que se haya proporcionado los datos del usuario y clave previamente.
Flujo Normal:	<ol style="list-style-type: none"> 1. Se inicia cuando el actor introduce la nueva contraseña. 2. El sistema verifica que los campos requeridos contengan datos y sean mayores a 6 caracteres. 3. El sistema verifica que no exista nombre y contraseña repetidos. 4. El sistema despliega un mensaje de confirmación.
Flujo Alternativo:	Si los campos están vacíos el sistema no permite ingresar al usuario
Post condición:	El nombre de usuario y contraseña han sido validados.

3.2.5. CASO DE USO: CONSULTA PARTE DIARIA

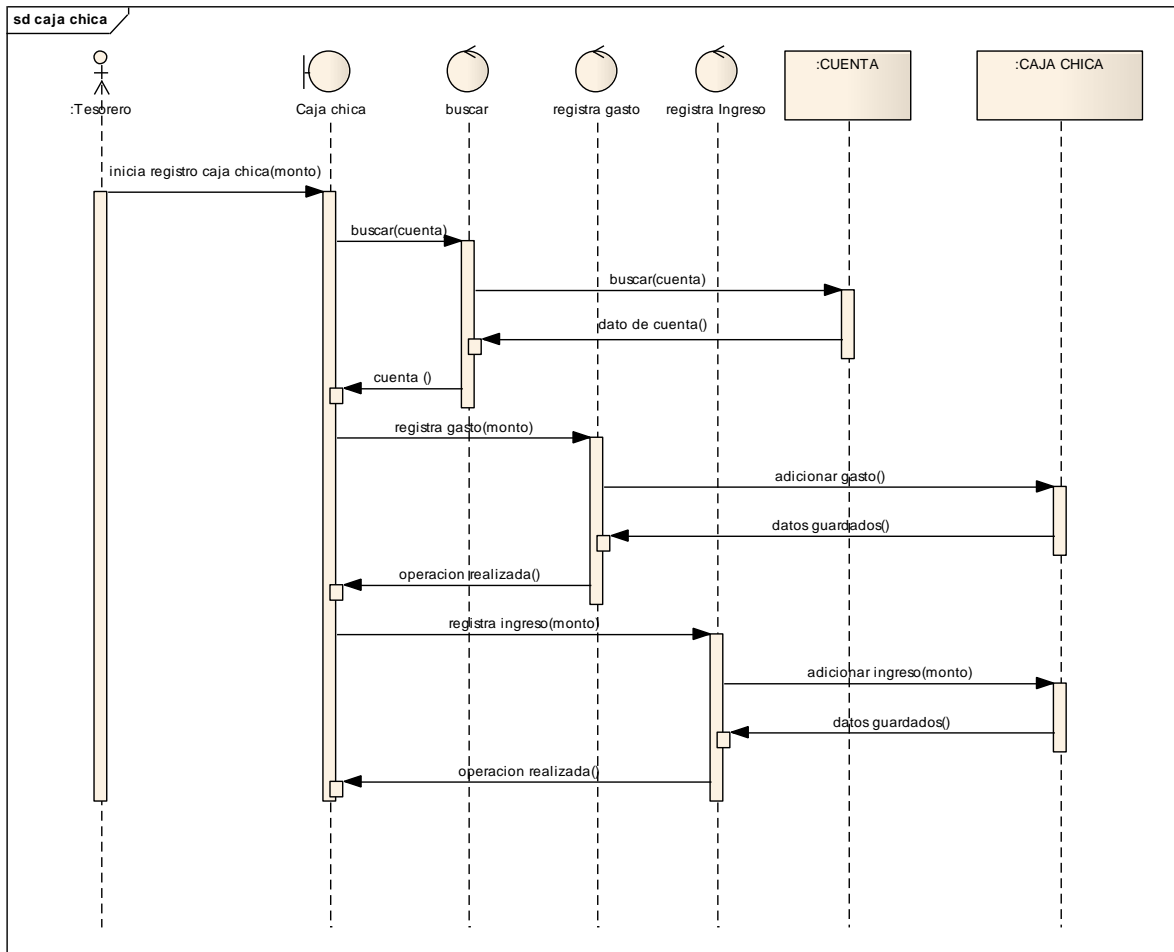
Caso de Uso:	Consulta informe
Descripción:	Este caso de uso permite a las entidades externas realizar una consulta específica.
Actores:	Entidades externas.
Precondición:	Que el usuario elija la pantalla Consulta de diarios.
Flujo Normal:	<ol style="list-style-type: none"> 1.- El usuario selecciona pantalla de consultas de parte diario entidades externas. 2.- El sistema despliega la pantalla. 3.- El usuario procede a realizar el parametrizado de los datos 4.- El sistema la consulta mediante los parámetros introducido por el usuario. 5.- El usuario selecciona la generación de partes diarios. 6.- El sistema realiza la generación internamente. 7.- El sistema visualiza los campos de la tabla de parte diario solicitado.
Flujo Alternativo:	Si se ha desplegado un mensaje de error el actor selecciona Aceptar y vuelve al flujo.
Post condición:	La consulta es visualizada.

3.3. DIAGRAMAS DE SECUENCIA

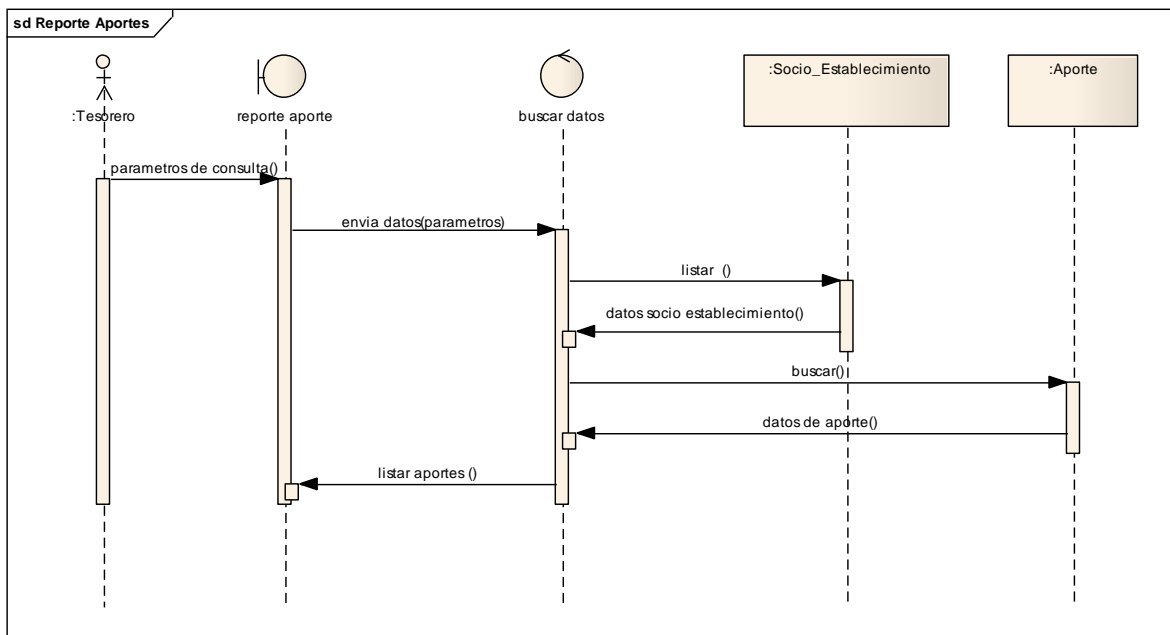
3.3.1. Aportes



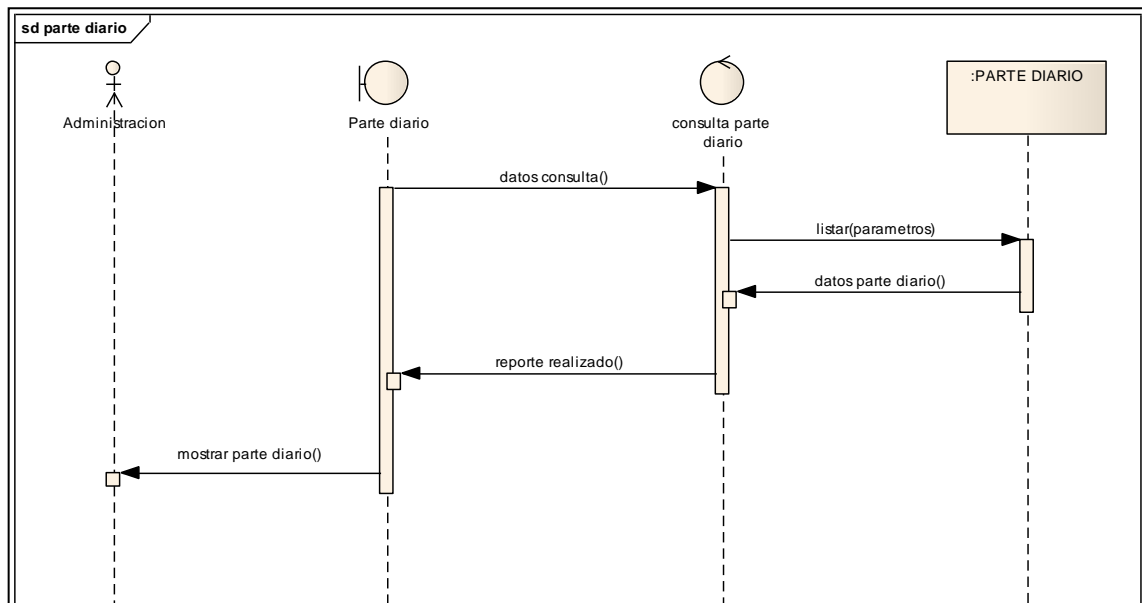
3.3.2. Caja chica



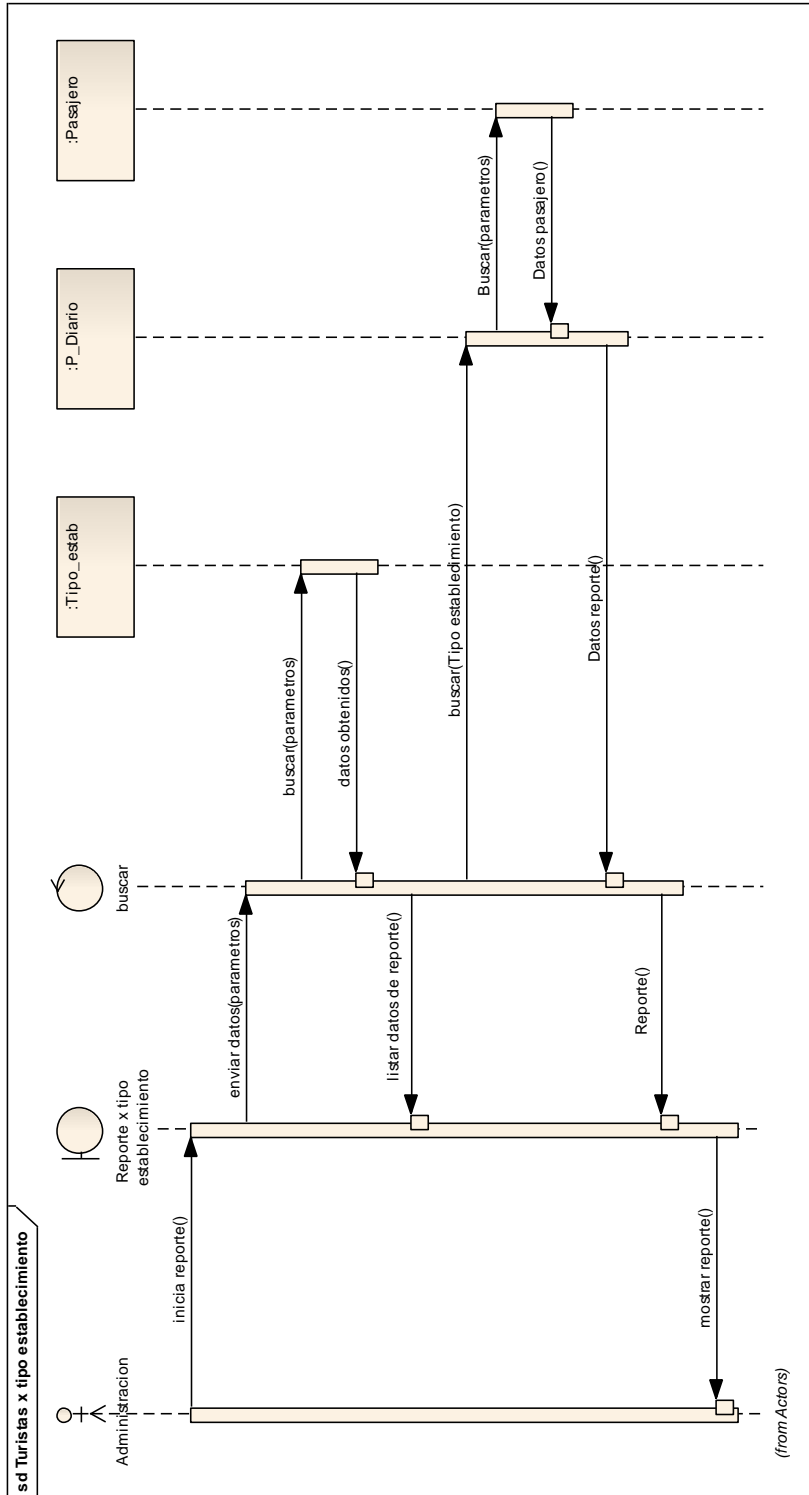
3.3.3. Reporte aporte



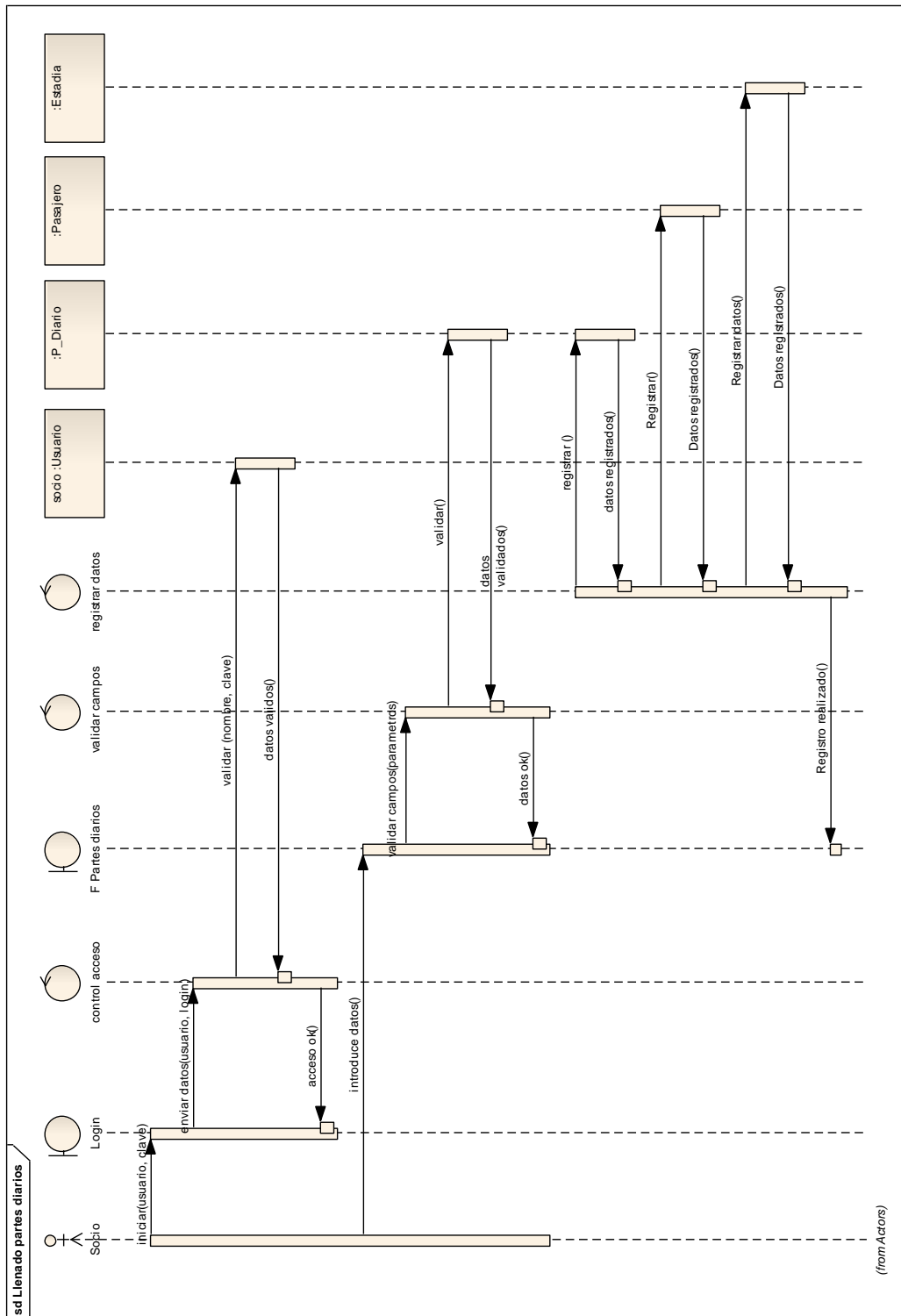
3.3.4. Reporte parte diario



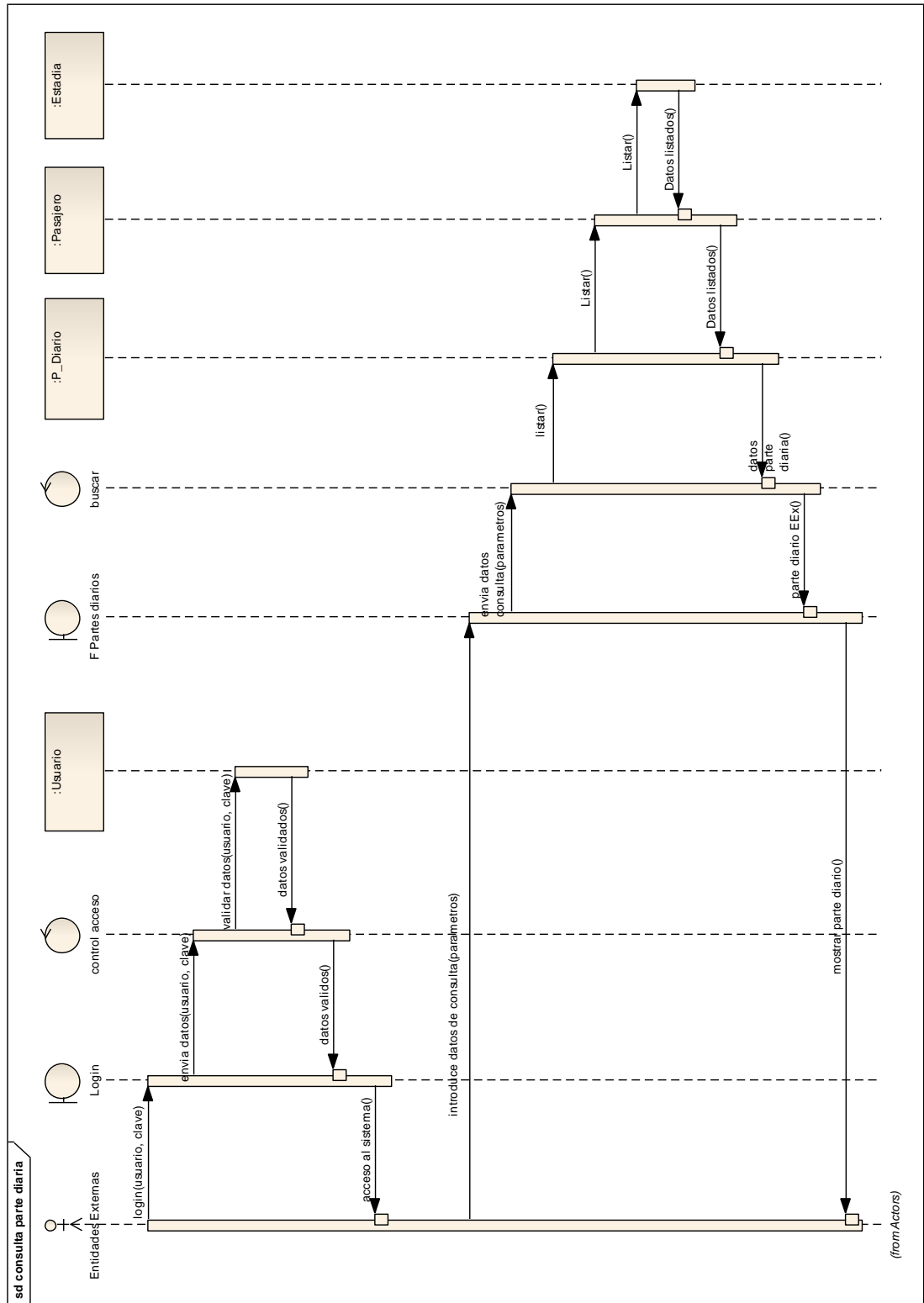
3.3.5. Reporte de turistas por tipo de establecimiento



3.3.6. Gestión parte diario

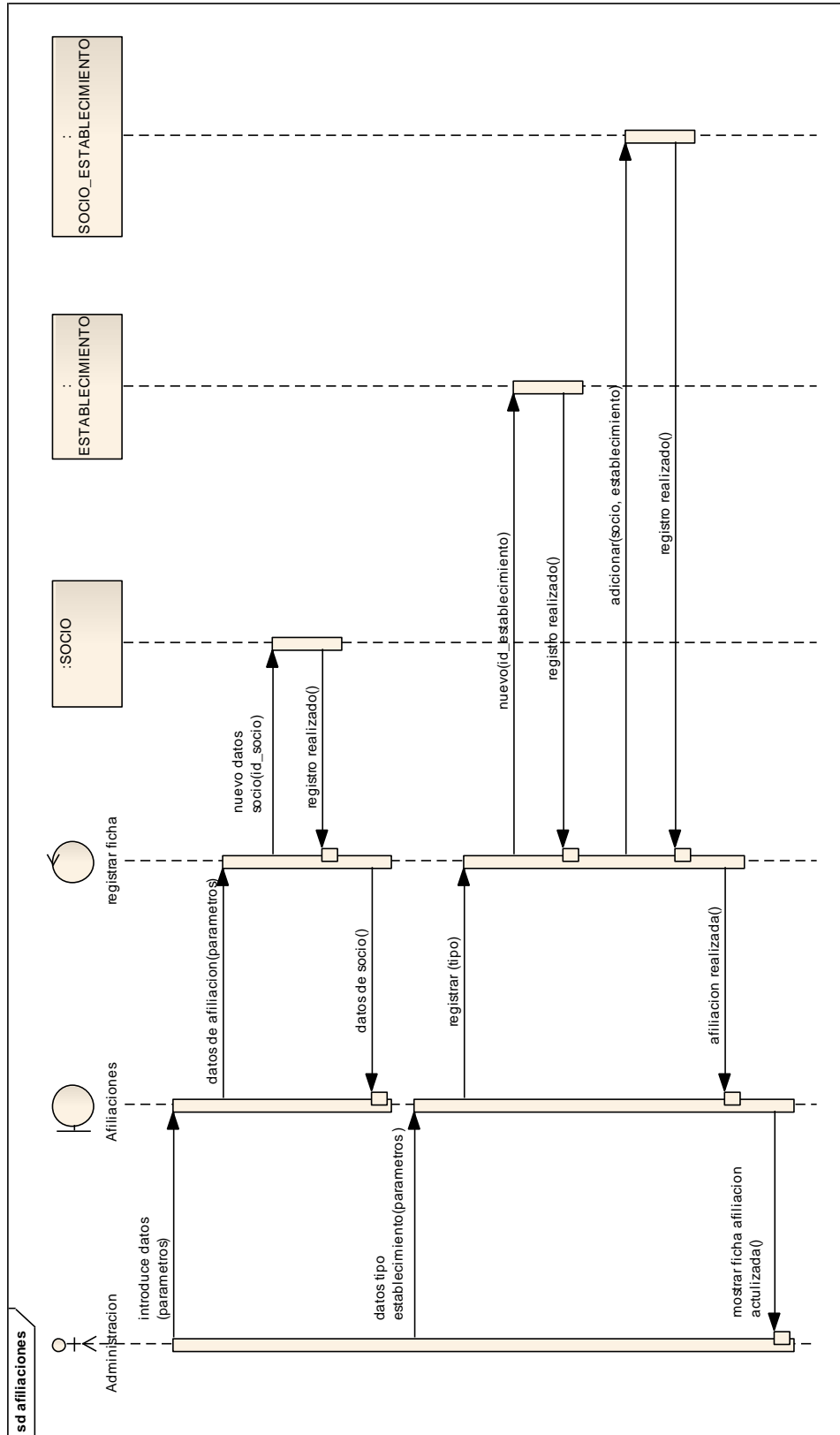


3.3.7. Consulta parte diario

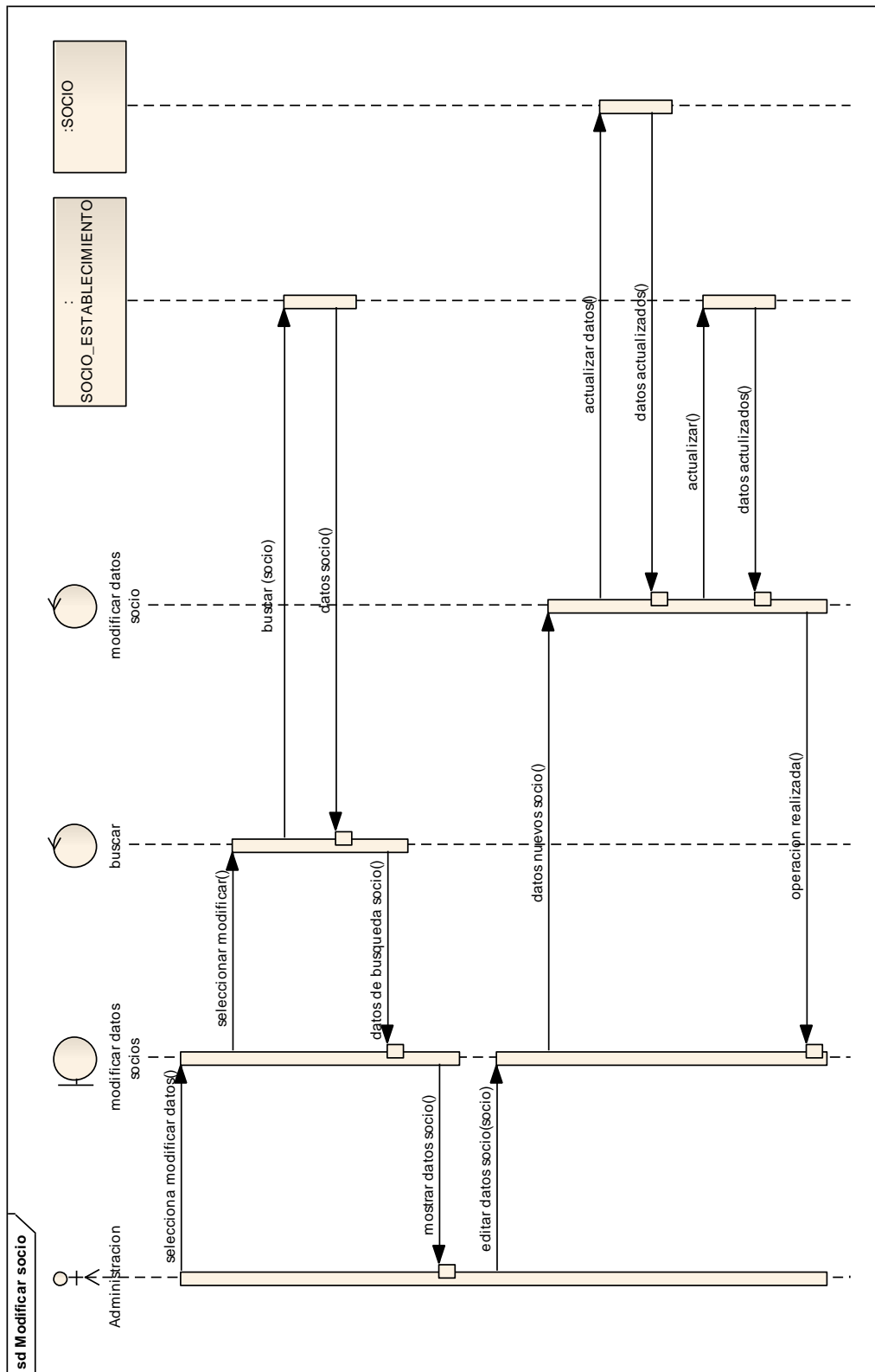


(from Actors)

3.3.8. Afiliaciones

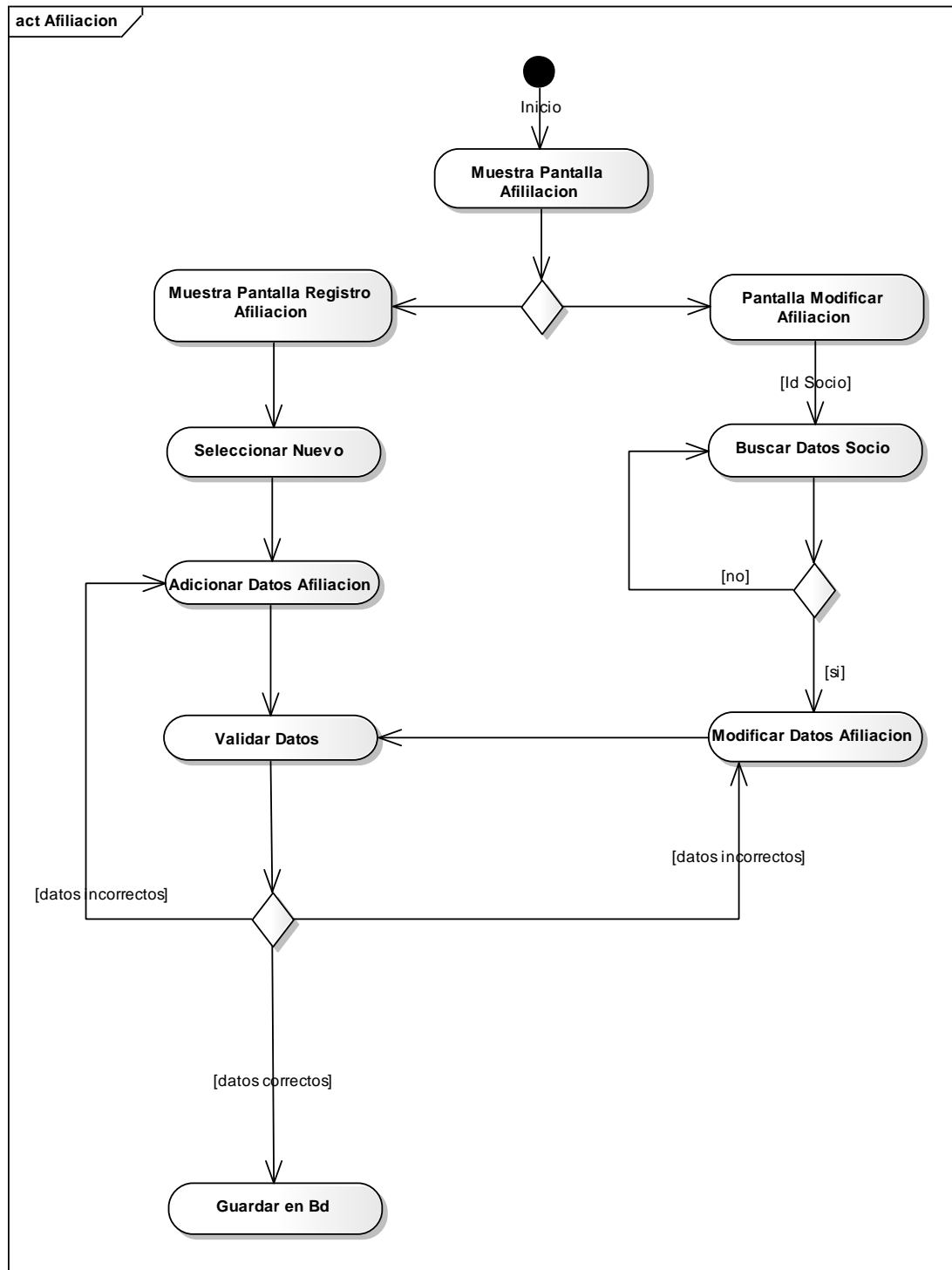


3.3.9. Modificar

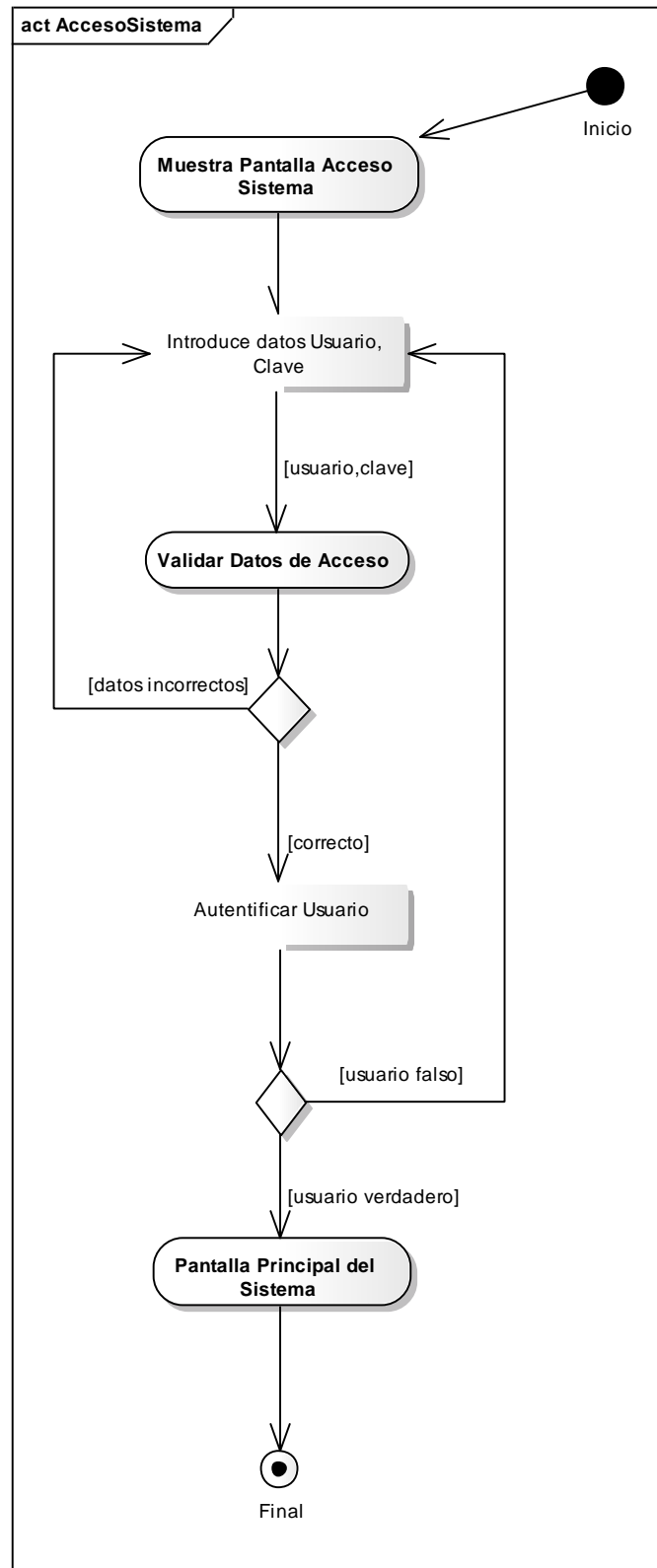


3.4. DIAGRAMAS DE ACTIVIDADES

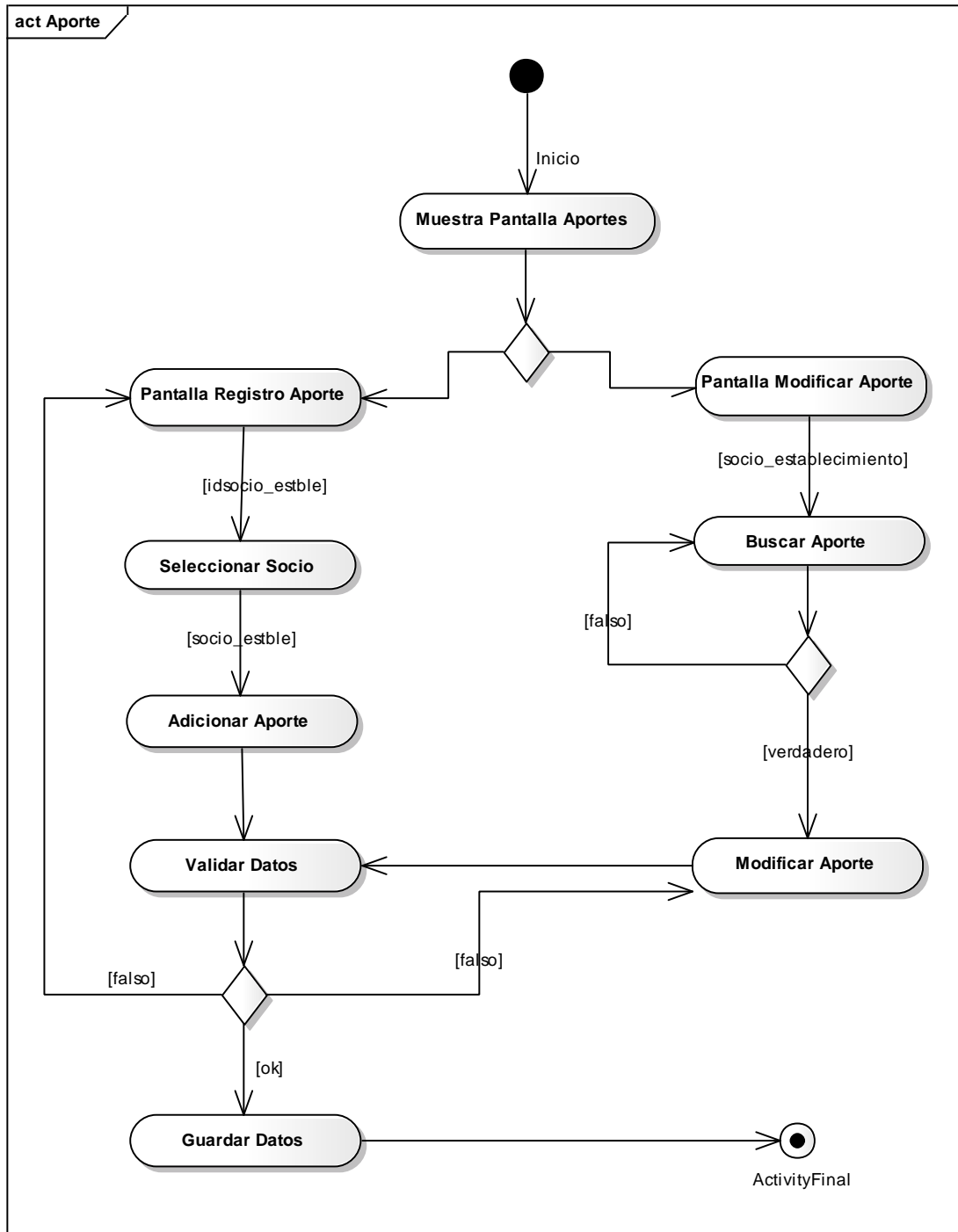
3.4.1. Afiliación



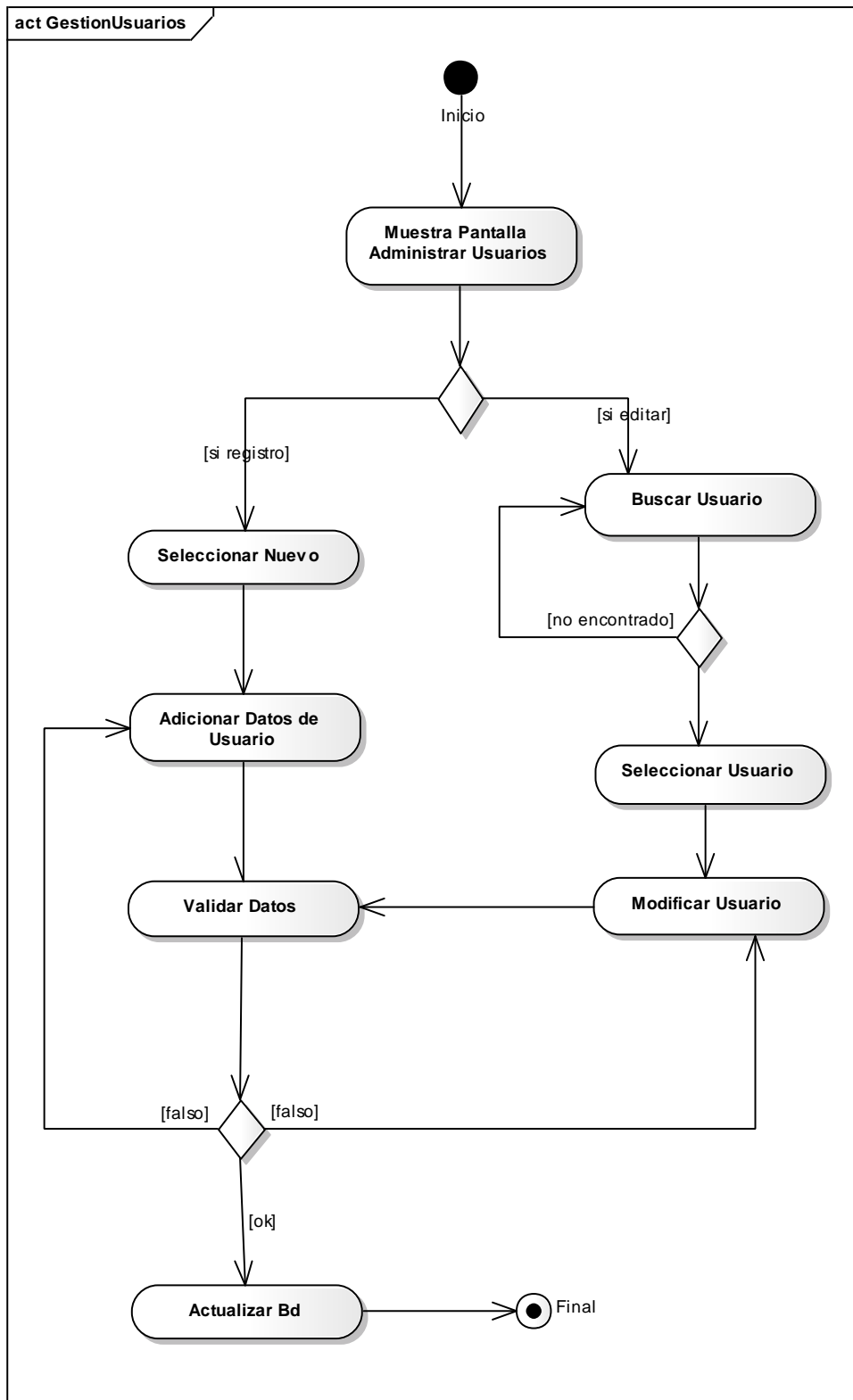
3.4.2. Acceso al sistema



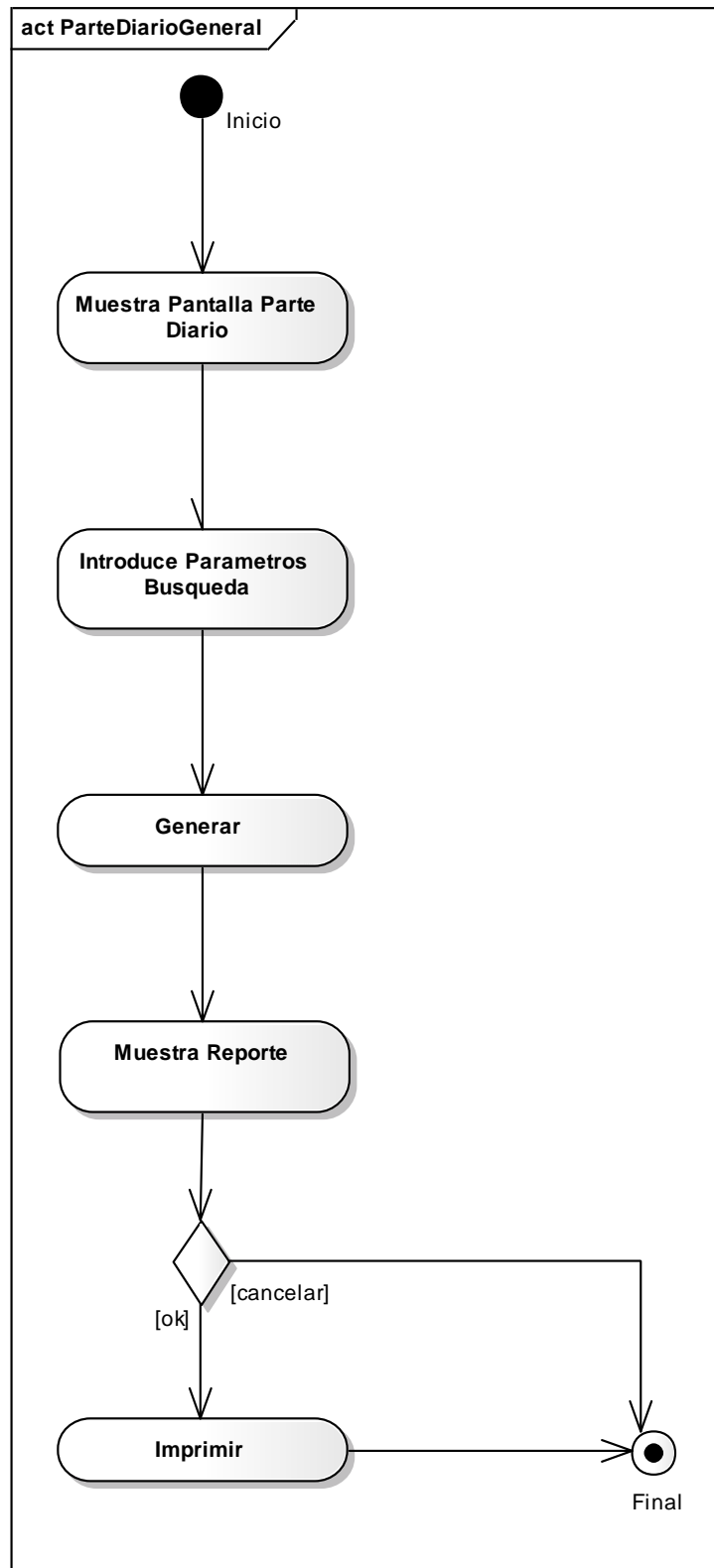
3.4.3. Aporte



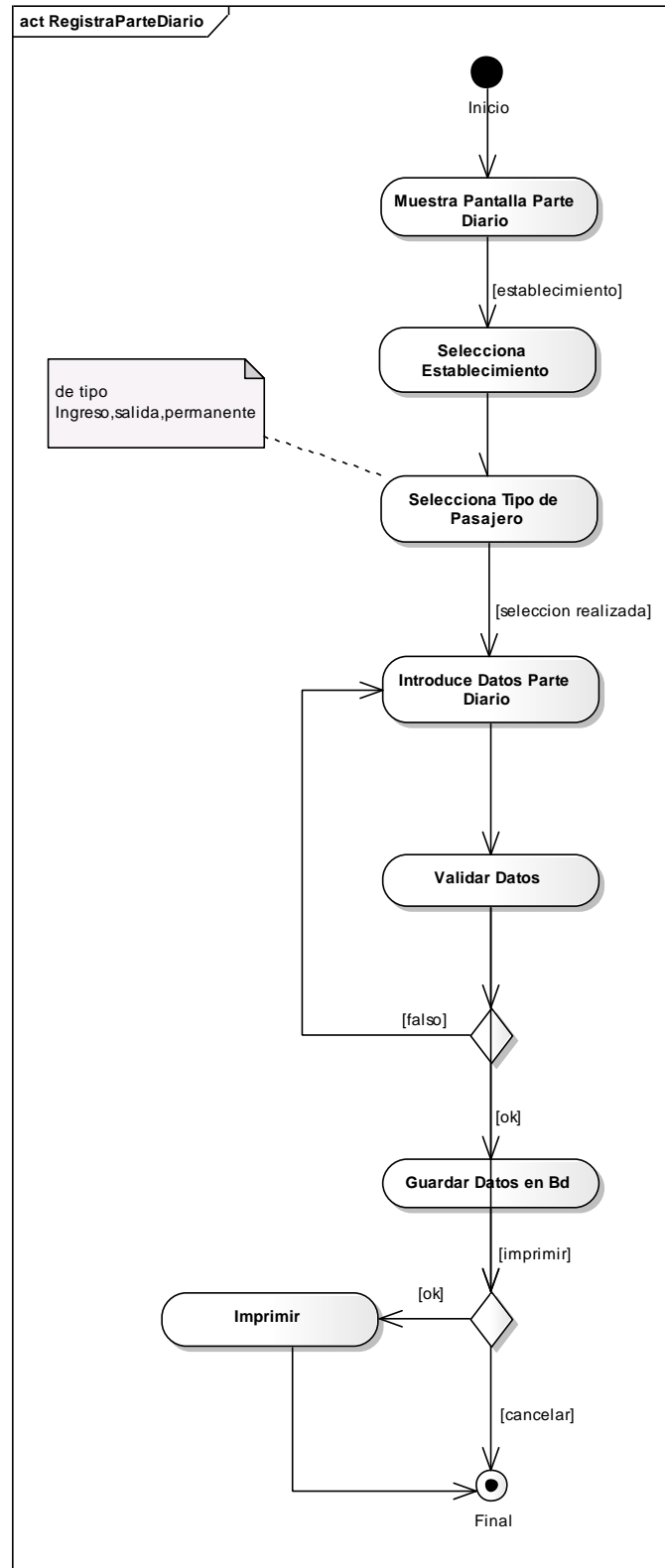
3.4.4. Gestión usuarios



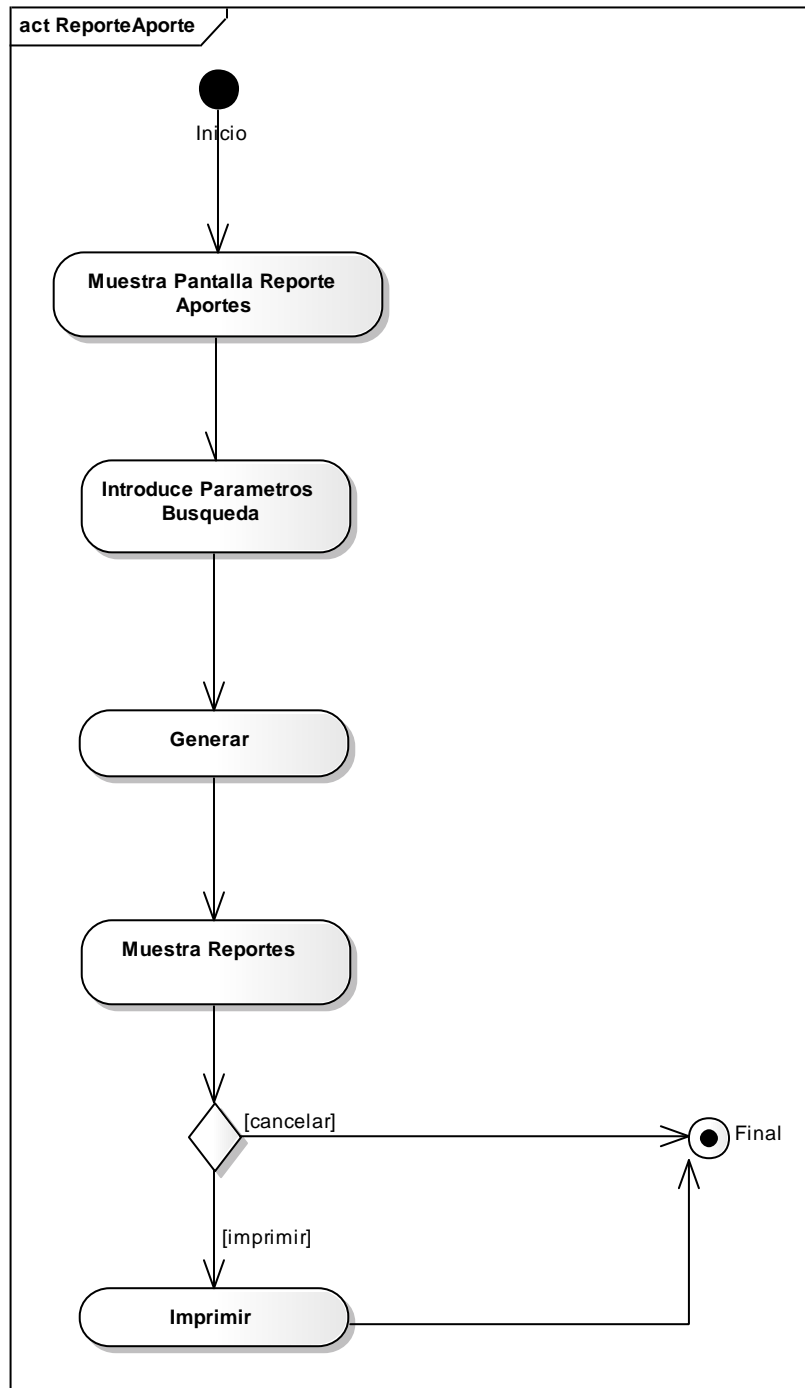
3.4.5. Parte diario general



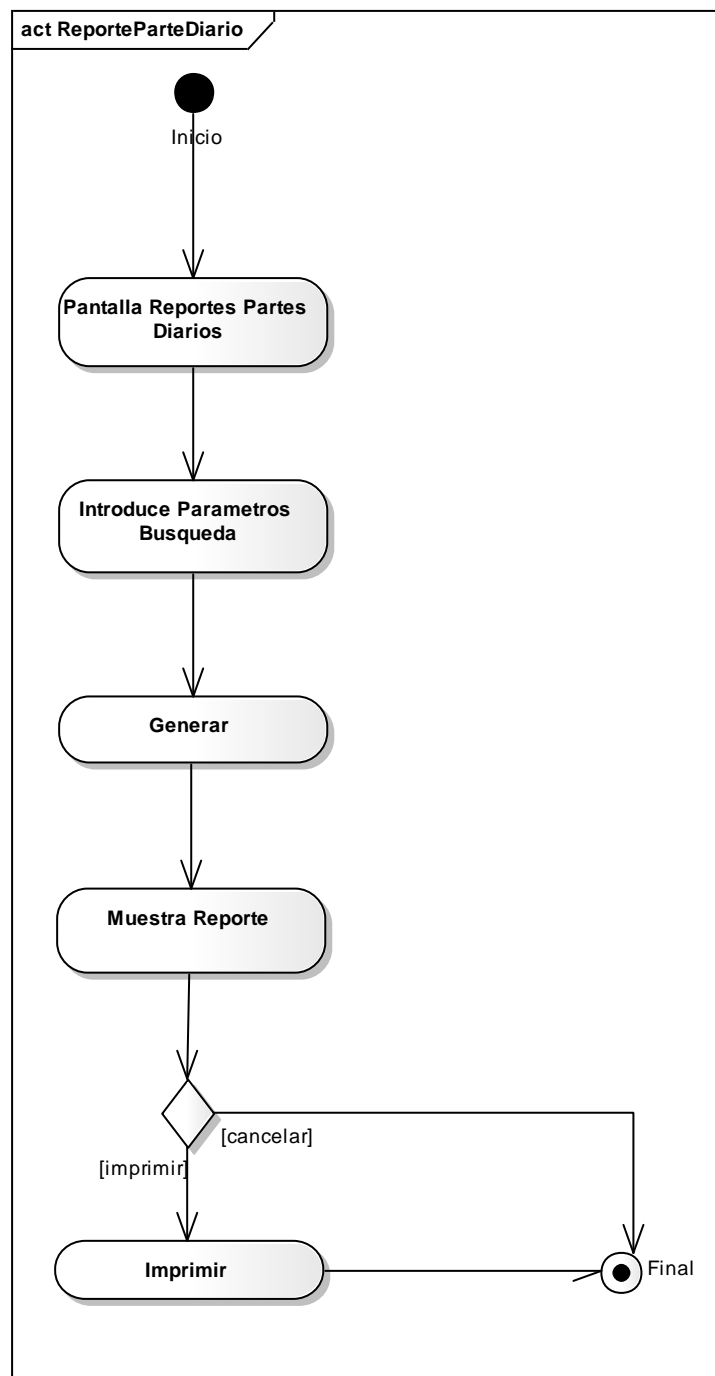
3.4.6. Registra parte diario



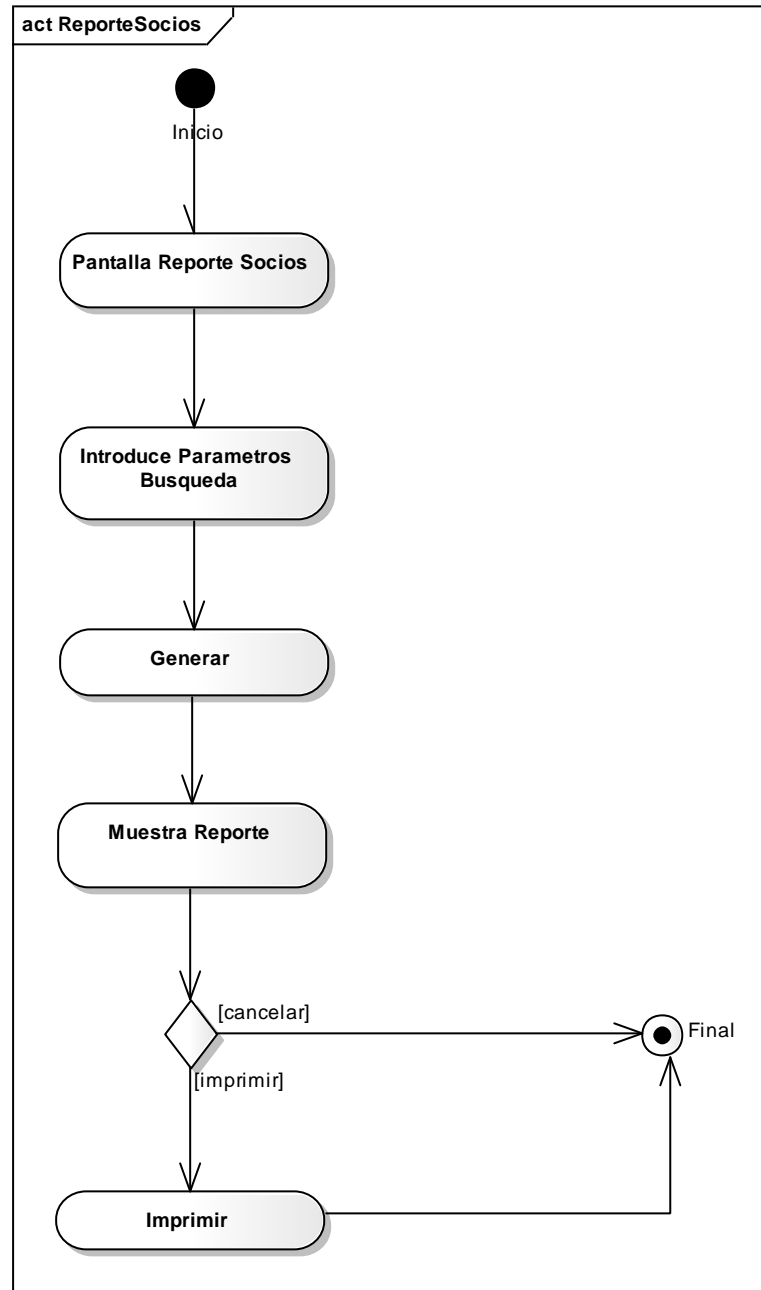
3.4.7. Reporte aporte



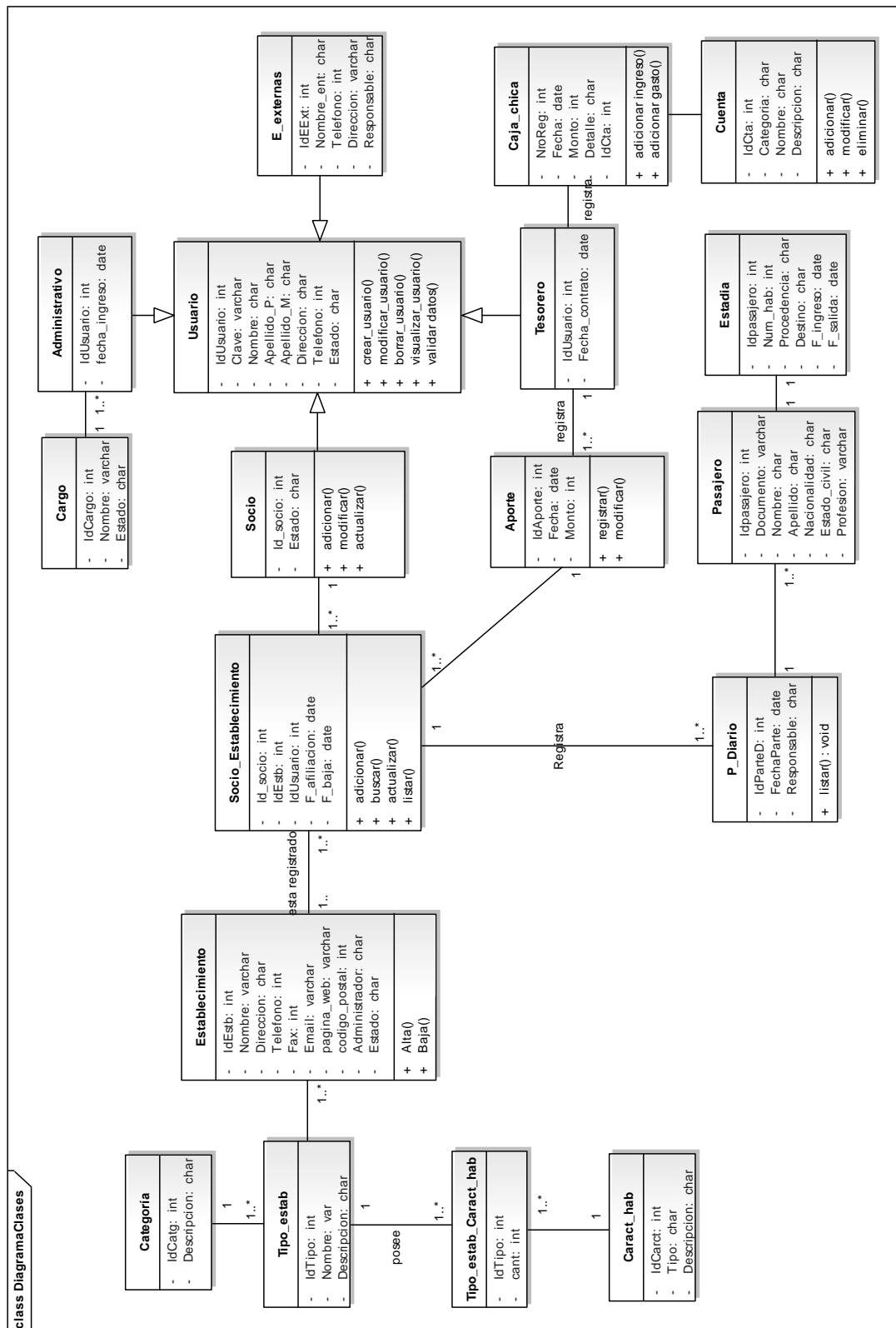
3.4.8. Reporte parte diario



3.4.9. Reporte Socios



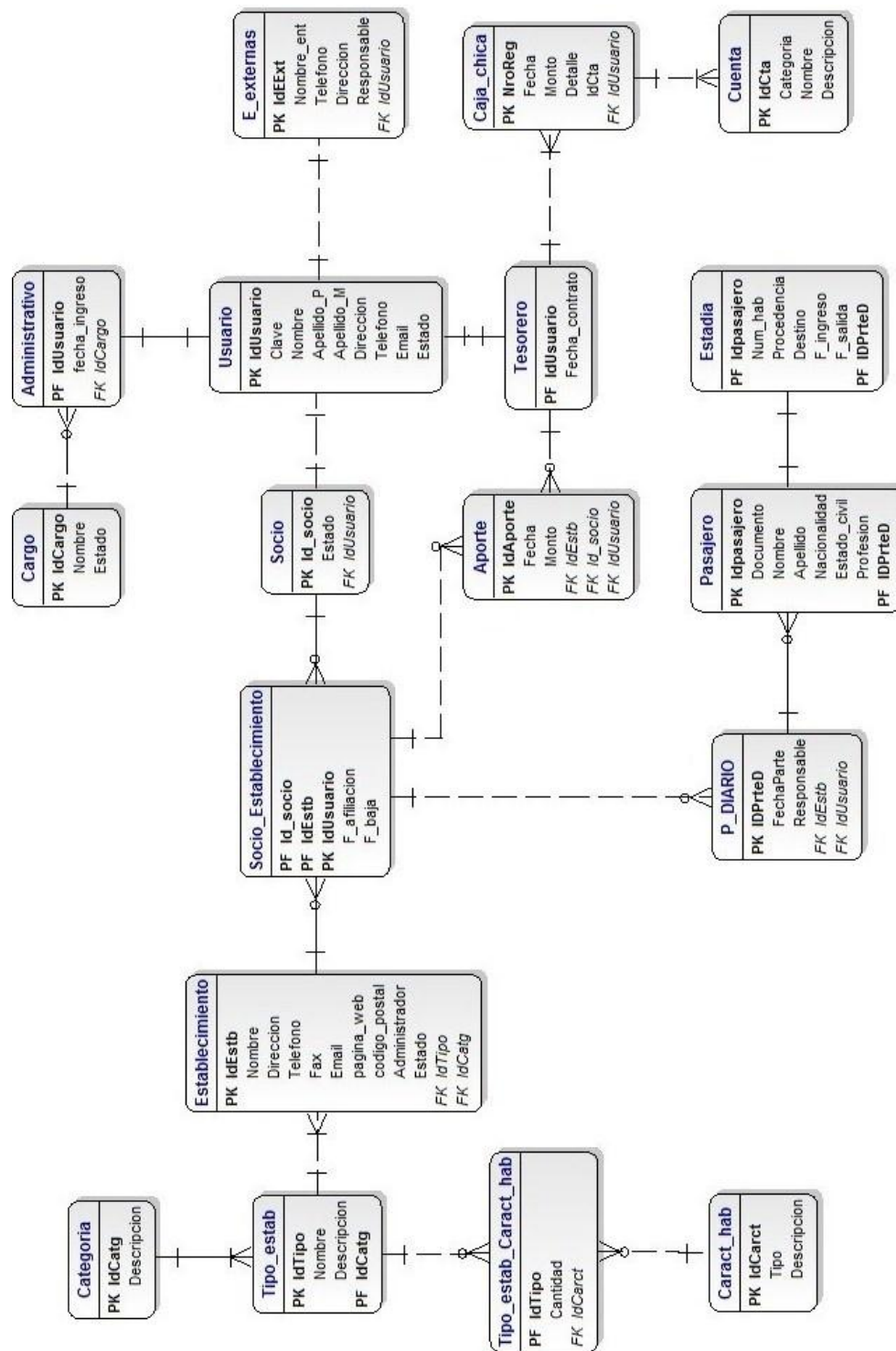
3.5. DIAGRAMA DE CLASES



CAPÍTULO IV
DISEÑO DEL SISTEMA

4.1. DISEÑO DE LA BASE DE DATOS

4.1.1. Diagrama de la Base de Datos



4.1.2. DESCRIPCIÓN DE LAS TABLAS

Tabla: Usuario				
Descripción: registra los datos de los usuarios del sistema.				
CAMPO	LLAVE	TIPO DATO	NO NULO	DESCRIPCIÓN
IdUsuario	PK	INTEGER	Yes	Identificador único de la tabla
Clave		VARCHAR(20)	Yes	Contraseña del usuario que utiliza para ingresar al sistema
Nombre		CHAR(20)	Yes	Nombre del usuario
Apellido_P		CHAR(20)	No	Apellido paterno del usuario
Apellido_M		CHAR(20)	No	Apellido materno del usuario
Direccion		CHAR(20)	Yes	Dirección del usuario
Telefono		INTEGER	Yes	Número de teléfono del usuario
Email		VARCHAR(40)	No	Email del usuario
Estado		CHAR(20)	No	Permite hacer un borrado lógico del usuario (baja)

Tabla: Administrativo				
Descripción: registra los datos correspondientes al personal de la CHT que accede al sistema.				
CAMPO	LLAVE	TIPO DATO	NO NULO	DESCRIPCIÓN
IdUsuario	PF	INTEGER	Yes	Identificador único de la tabla
fecha_ingreso		DATE	Yes	Fecha desde la que trabaja en la institución
IdCargo	FK	INTEGER	Yes	Identificador del cargo del administrativo

Tabla: Aporte				
Descripción: registra los datos de un aporte realizado por los socios.				
CAMPO	LLAVE	TIPO DATO	NO NULO	DESCRIPCIÓN
IdAporte	PK	INTEGER	Yes	Identificador del aporte
Fecha		DATE	Yes	Fecha del aporte
Monto		INTEGER	Yes	Monto del aporte correspondiente
IdEstb	FK	INTEGER	Yes	Identificador del establecimiento
Id_socio	FK	INTEGER	Yes	Identificador del socio
IdUsuario	FK	INTEGER	Yes	Identificador del usuario

Tabla: Caja Chica				
Descripción: registra las transacciones realizadas por ingresos, egresos y otros.				
CAMPO	LLAVE	TIPO DATO	NO NULO	DESCRIPCIÓN
NroReg	PK	INTEGER	Yes	Identificador único de la transacción registrada
Fecha		DATE	Yes	Fecha del registro
Monto		INTEGER	Yes	Monto registrado
Detalle		CHAR(20)	Yes	Detalle de la transaccion realizada
IdCta		INTEGER	Yes	Identificador de la cuenta
IdUsuario	FK	INTEGER	Yes	Identificador del usuario

Tabla: Caract_hab				
Descripción: registra las características de las habitaciones de un establecimiento.				
CAMPO	LLAVE	TIPO DATO	NO NULO	DESCRIPCIÓN
IdCarct	PK	INTEGER	Yes	Identificador de las características de una habitación.
Tipo		CHAR(20)	Yes	Indica el tipo de habitación (simple, doble, suite y otras)
Descripcion		CHAR(30)	Yes	Descripción de las características de la habitación

Tabla: Cargo				
Descripción: registra los cargos existentes en la CHT.				
CAMPO	LLAVE	TIPO DATO	NO NULO	DESCRIPCIÓN
IdCargo	PK	INTEGER	Yes	Identificador único de la tabla
Nombre		VARCHAR(30)	Yes	Nombre del cargo del Administrativo
Estado		CHAR(20)	Yes	Permite hacer un borrado lógico (baja) del cargo.

Tabla: Categoría				
Descripción: Registra la categorización de los establecimientos hoteleros.				
CAMPO	LLAVE	TIPO DATO	NO NULO	DESCRIPCIÓN
IdCatg	PK	INTEGER	Yes	Identificador de la categoría
Descripcion		CHAR(20)	No	Descripción de la categoría del establecimiento (indica el número de estrellas o clases)

Tabla: Cuenta				
Descripción: registra los datos de las cuentas.				
CAMPO	LLAVE	TIPO DATO	NO NULO	DESCRIPCIÓN
IdCta	PK	INTEGER	Yes	Identificador único de la cuenta
Categoria		CHAR(20)	No	Categoría a la que pertenece la cuenta
Nombre		CHAR(20)	No	Nombre de la cuenta
Descripcion		CHAR(20)	No	Descripción o detalle de la cuenta

Tabla: Estadía				
Descripción: registra los datos de la estadía de un hospedado en un establecimiento				
CAMPO	LLAVE	TIPO DATO	NO NULO	DESCRIPCIÓN
Idpasajero	PF	INTEGER	Yes	Identificador del pasajero
Num_hab		INTEGER	Yes	Número de habitación donde se hospeda el pasajero
Procedencia		CHAR(20)	Yes	Procedencia del hospedado
Destino		CHAR(20)	Yes	Destino del hospedado al salir del establecimiento.
F_ingreso		DATE	Yes	Fecha que ingresó al establecimiento.
F_salida		DATE	Yes	Fecha que sale del establecimiento.
IDPrteD	PF	VARCHAR(40)	Yes	Identificador del parte diario

Tabla: Establecimiento				
Descripción: registra los datos de los establecimientos.				
CAMPO	LLAVE	TIPO DATO	NO NULO	DESCRIPCIÓN
IdEstb	PK	INTEGER	Yes	Identificador único de la tabla
Nombre		VARCHAR(30)	Yes	Nombre del establecimiento hotelero
Direccion		CHAR(30)	Yes	Dirección del establecimiento
Telefono		INTEGER	No	Teléfono del establecimiento
Fax		INTEGER	No	Fax del establecimiento
Email		VARCHAR(40)	No	Email del establecimiento
pagina_web		VARCHAR(40)	No	Página web del establecimiento
codigo_postal		INTEGER	No	Código postal del establecimiento
Administrador		CHAR(20)	Yes	Nombre del administrador o encargado del establecimiento
Estado		CHAR(10)	Yes	Indica si el establecimiento está activo, inactivo o dado de baja.
IdTipo	FK	INTEGER	Yes	Identificador del tipo de establecimiento
IdCatg	FK	INTEGER	Yes	Identificador de la categoría del establecimiento

Tabla: E_externas				
Descripción:				
CAMPO	LLAVE	TIPO DATO	NO NULO	DESCRIPCIÓN
IdEExt	PK	INTEGER	Yes	Identificador único de la tabla
Nombre_ent		CHAR(20)	Yes	Nombre de la entidad externa
Telefono		INTEGER	Yes	Teléfono de la Entidad Externa
Direccion		VARCHAR(40)	Yes	Dirección de la Entidad Externa
Responsable		CHAR(30)	Yes	Responsable de la Entidad Externa
IdUsuario	FK	INTEGER	Yes	Identificador del usuario

Tabla: Pasajero				
Descripción: registra los datos de un pasajero hospedado en un establecimiento.				
CAMPO	LLAVE	TIPO DATO	NO NULO	DESCRIPCIÓN
Idpasajero	PK	INTEGER	Yes	Identificador del pasajero
Documento		VARCHAR(20)	Yes	Número de documento del hospedado
Nombre		CHAR(30)	Yes	Nombre de la persona hospedada
Apellido		CHAR(30)	Yes	Apellido de la persona hospedada
Nacionalidad		CHAR(20)	Yes	Nacionalidad del hospedado
Estado_civil		CHAR(10)	Yes	Estado civil del hospedado
Profesion		CHAR(20)	Yes	Profesión del hospedado
IDPrteD	PF	VARCHAR(40)	Yes	Identificador del parte diario

Tabla: P_DIARIO				
Descripción: registra los datos del Parte Diario.				
CAMPO	LLAVE	TIPO DATO	NO NULO	DESCRIPCIÓN
IDPrteD	PK	VARCHAR(40)	Yes	Identificador único del parte diario
FechaParte		DATE	Yes	Fecha del parte diario
Responsable		CHAR(30)	Yes	Responsable del llenado del parte diario
IdEstb	FK	INTEGER	Yes	Identificador del establecimiento
IdUsuario	FK	INTEGER	Yes	Identificador del usuario

Tabla: Socio				
Descripción: registra los datos de un socio.				
CAMPO	LLAVE	TIPO DATO	NO NULO	DESCRIPCIÓN
Id_socio	PK	INTEGER	Yes	Identificador del socio
Estado		CHAR(10)	No	Indica el estado de un socio, si está activo o dado de baja.
IdUsuario	FK	INTEGER	Yes	Identificador del usuario

Tabla: Socio_Establecimiento				
Descripción: registra los datos de la relación socio y establecimiento.				
CAMPO	LLAVE	TIPO DATO	NO NULO	DESCRIPCIÓN
Id_socio	PF	INTEGER	Yes	Identificador del socio
IdEstb	PF	INTEGER	Yes	Identificador del establecimiento
IdUsuario	PK	INTEGER	Yes	Identificador del usuario
F_afiliacion		DATE	Yes	Fecha de afiliación a la CHT del establecimiento
F_baja		DATE	No	Fecha en la que deja de ser miembro de la CHT

Tabla: Tesorero				
Descripción: registra los datos del Tesorero de la institución.				
CAMPO	LLAVE	TIPO DATO	NO NULO	DESCRIPCIÓN
IdUsuario	PF	INTEGER	Yes	Identificador del usuario
Fecha_contrato		DATE	Yes	Fecha desde la que trabaja en la institución

Tabla: Tipo_estab				
Descripción: registra los datos de los tipos de establecimiento.				
CAMPO	LLAVE	TIPO DATO	NO NULO	DESCRIPCIÓN
IdTipo	PK	INTEGER	Yes	Identificador del tipo de establecimiento
Nombre		CHAR(20)	Yes	Nombre del tipo de establecimiento (hotel, hostel, alojamiento, residencial)
Descripcion		CHAR(20)	No	Descripción del tipo de establecimiento
IdCatg	PF	INTEGER	Yes	Identificador de la categoría

Tabla: Tipo_estab_Caract_hab				
Descripción: registra los datos de la cantidad de habitaciones por característica.				
CAMPO	LLAVE	TIPO DATO	NO NULO	DESCRIPCIÓN
IdTipo	PF	INTEGER	Yes	Identificador del tipo de establecimiento
Cantidad		INTEGER	Yes	Indica la cantidad de habitaciones con que cuenta un establecimiento
IdCarct	FK	INTEGER	Yes	Identificador de las características de una habitación

4.1.3. CODIGO SQL

TABLA APORTE

```
CREATE TABLE APORTE (  
    IDAPORTE INTEGER NOT NULL,  
    FECHA DATE NOT NULL,  
    MONTO INTEGER NOT NULL,  
    IDESTB INTEGER NOT NULL,  
    ID_SOCIO INTEGER NOT NULL,  
    IDUSUARIO INTEGER NOT NULL,  
    CONSTRAINT PK_APORTE PRIMARY KEY (IDAPORTE)  
);
```

TABLA ESTABLECIMIENTO

```
CREATE TABLE ESTABLECIMIENTO (  
    IDESTB INTEGER NOT NULL,  
    NOMBRE VARCHAR(30) NOT NULL,  
    DIRECCION CHAR(30) NOT NULL,  
    TELEFONO INTEGER,  
    FAX INTEGER,  
    EMAIL VARCHAR(40),  
    PAGINA_WEB VARCHAR(40),  
    CODIGO_POSTAL INTEGER,  
    ADMINISTRADOR CHAR(20) NOT NULL,  
    ESTADO CHAR(10) NOT NULL,  
    IDTIPO INTEGER NOT NULL,  
    IDCATG INTEGER NOT NULL,  
    CONSTRAINT PK_ESTABLECIMIENTO PRIMARY KEY (IDESTB)  
);
```

TABLA P_DIARIO

```
CREATE TABLE P_DIARIO (  
    IDPRTED VARCHAR(40) NOT NULL,  
    FECHAPARTE DATE NOT NULL,  
    RESPONSABLE CHAR(30) NOT NULL,  
    IDESTB INTEGER NOT NULL,  
    IDUSUARIO INTEGER NOT NULL,  
    CONSTRAINT PK_P_DIARIO PRIMARY KEY (IDPRTED)  
);
```

TABLA CAJA_CHICA

```
CREATE TABLE CAJA_CHICA (  
    NROREG INTEGER NOT NULL,  
    FECHA DATE NOT NULL,  
    MONTO INTEGER NOT NULL,  
    DETALLE CHAR(20) NOT NULL,  
    IDCTA INTEGER NOT NULL,  
    IDUSUARIO INTEGER NOT NULL,  
    CONSTRAINT PK_CAJA_CHICA PRIMARY KEY (NROREG)  
);
```

TABLA CAJA_CHICA

```
CREATE TABLE CUENTA (  
    IDCTA VARCHAR(40) NOT NULL,  
    CATEGORIA CHAR(20),  
    NOMBRE CHAR(20),  
    DESCRIPCION CHAR(20),  
    CONSTRAINT PK_CUENTA PRIMARY KEY (IDCTA)  
);
```

TABLA USUARIO

```
CREATE TABLE USUARIO (  
    IDUSUARIO INTEGER NOT NULL,  
    CLAVE VARCHAR(20) NOT NULL,  
    NOMBRE CHAR(20) NOT NULL,  
    APELLIDO_P CHAR(20),  
    APELLIDO_M CHAR(20),  
    DIRECCION CHAR(20) NOT NULL,  
    TELEFONO INTEGER NOT NULL,  
    EMAIL VARCHAR(40),  
    ESTADO CHAR(20),  
    PRIMARY KEY (IDUSUARIO)  
);
```

TABLA TESORERO

```
CREATE TABLE TESORERO (  
    IDUSUARIO INTEGER NOT NULL,  
    FECHA_CONTRATO DATE NOT NULL,  
    CONSTRAINT PK_TESORERO PRIMARY KEY (IDUSUARIO)  
);
```

TABLA E_EXTERNAS

```
CREATE TABLE E_EXTERNAS (  
    IDEEXT INTEGER NOT NULL,  
    NOMBRE_ENT CHAR(20) NOT NULL,  
    TELEFONO INTEGER NOT NULL,  
    DIRECCION VARCHAR(40) NOT NULL,  
    RESPONSABLE CHAR(30) NOT NULL,  
    IDUSUARIO INTEGER NOT NULL,  
    CONSTRAINT PK_E_EXTERNAS PRIMARY KEY (IDEEXT)  
);
```

TABLA TIPO_ESTAB

```
CREATE TABLE TIPO_ESTAB (  
    IDTIPO INTEGER NOT NULL,  
    NOMBRE CHAR(20) NOT NULL,  
    DESCRIPCION CHAR(20),  
    IDCATG INTEGER NOT NULL,  
    CONSTRAINT PK_TIPO_ESTAB PRIMARY KEY (IDTIPO, IDCATG)  
);
```

TABLA CARACT_HAB

```
CREATE TABLE CARACT_HAB (  
    IDCARCT INTEGER NOT NULL,  
    TIPO CHAR(20) NOT NULL,  
    DESCRIPCION CHAR(30) NOT NULL,  
    CONSTRAINT PK_CARACT_HAB PRIMARY KEY (IDCARCT)  
);
```

TABLA CATEGORIA

```
CREATE TABLE CATEGORIA (  
    IDCATG INTEGER NOT NULL,  
    DESCRIPCION CHAR(20),  
    CONSTRAINT PK_CATEGORIA PRIMARY KEY (IDCATG)  
);
```

TABLA TIPO_ESTAB_CARACT_HAB

```
CREATE TABLE TIPO_ESTAB_CARACT_HAB (  
    IDTIPO INTEGER NOT NULL,  
    CANTIDAD INTEGER NOT NULL,  
    IDCARCT INTEGER NOT NULL,  
    CONSTRAINT PK_TIPO_ESTAB_CARACT_HAB PRIMARY KEY (IDTIPO)  
);
```

TABLA ADMINISTRATIVO

```
CREATE TABLE ADMINISTRATIVO (  
    IDUSUARIO INTEGER NOT NULL,  
    FECHA_INGRESO DATE NOT NULL,  
    IDCARGO INTEGER NOT NULL,  
    PRIMARY KEY (IDUSUARIO)  
);
```

TABLA CARGO

```
CREATE TABLE CARGO (  
    IDCARGO INTEGER NOT NULL,  
    NOMBRE VARCHAR(30) NOT NULL,  
    ESTADO CHAR(20) NOT NULL,  
    CONSTRAINT PK_CARGO PRIMARY KEY (IDCARGO)  
);
```

TABLA PASAJERO

```
CREATE TABLE PASAJERO (  
    IDPASAJERO INTEGER NOT NULL,  
    DOCUMENTO VARCHAR(20) NOT NULL,  
    NOMBRE CHAR(30) NOT NULL,  
    APELLIDO CHAR(30) NOT NULL,  
    NACIONALIDAD CHAR(20) NOT NULL,  
    ESTADO_CIVIL CHAR(10) NOT NULL,  
    PROFESION CHAR(20) NOT NULL,  
    IDPRTEED VARCHAR(40) NOT NULL,  
    CONSTRAINT PK_PASAJERO PRIMARY KEY (IDPASAJERO, IDPRTEED)  
);
```

TABLA ESTADIA

```
CREATE TABLE ESTADIA (  
    IDPASAJERO INTEGER NOT NULL,  
    NUM_HAB INTEGER NOT NULL,  
    PROCEDENCIA CHAR(20) NOT NULL,  
    DESTINO CHAR(20) NOT NULL,  
    F_INGRESO DATE NOT NULL,  
    F_SALIDA DATE NOT NULL,  
    IDPRTED VARCHAR(40) NOT NULL,  
    CONSTRAINT PK_ESTADIA PRIMARY KEY (IDPASAJERO, IDPRTED)  
);
```

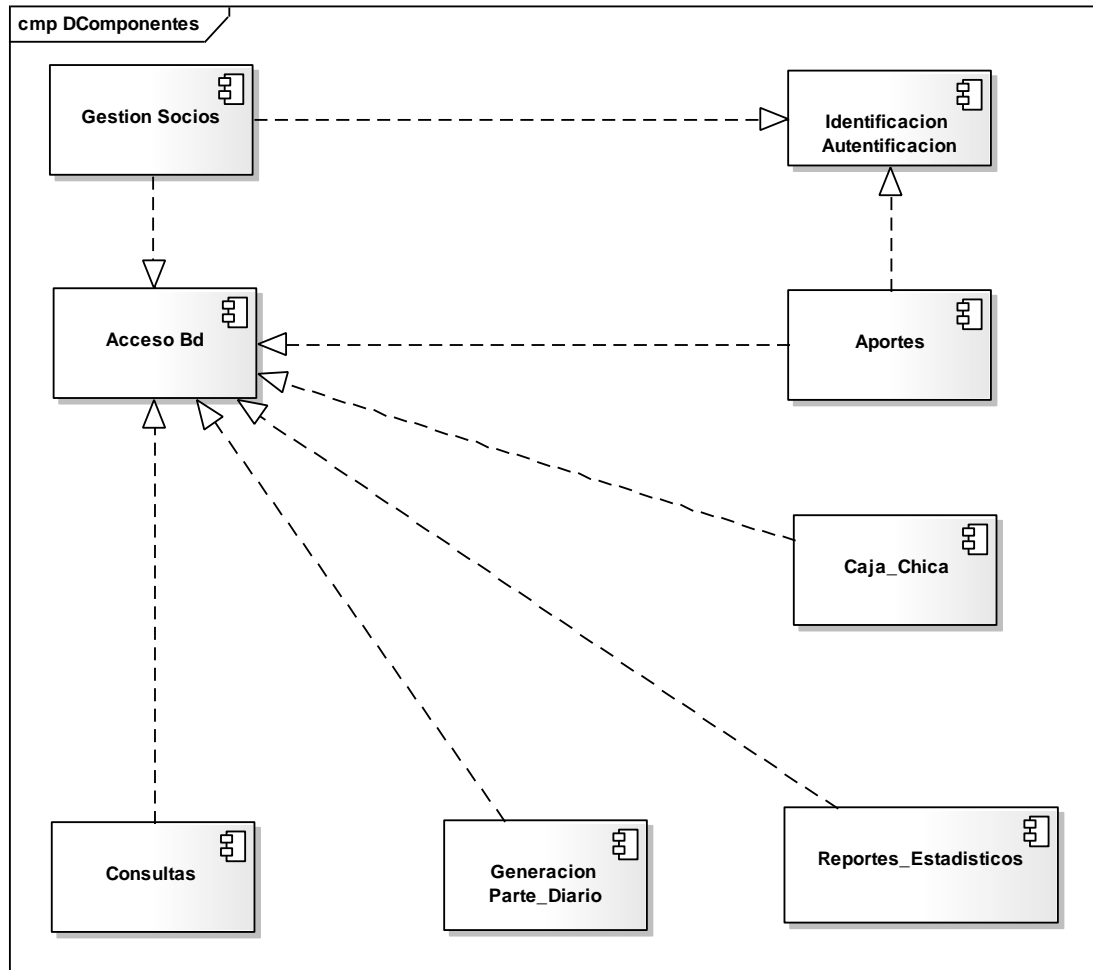
TABLA SOCIO

```
CREATE TABLE SOCIO (  
    ID_SOCIO INTEGER NOT NULL,  
    ESTADO CHAR(10),  
    IDUSUARIO INTEGER NOT NULL,  
    CONSTRAINT PK_SOCIO PRIMARY KEY (ID_SOCIO)  
);
```

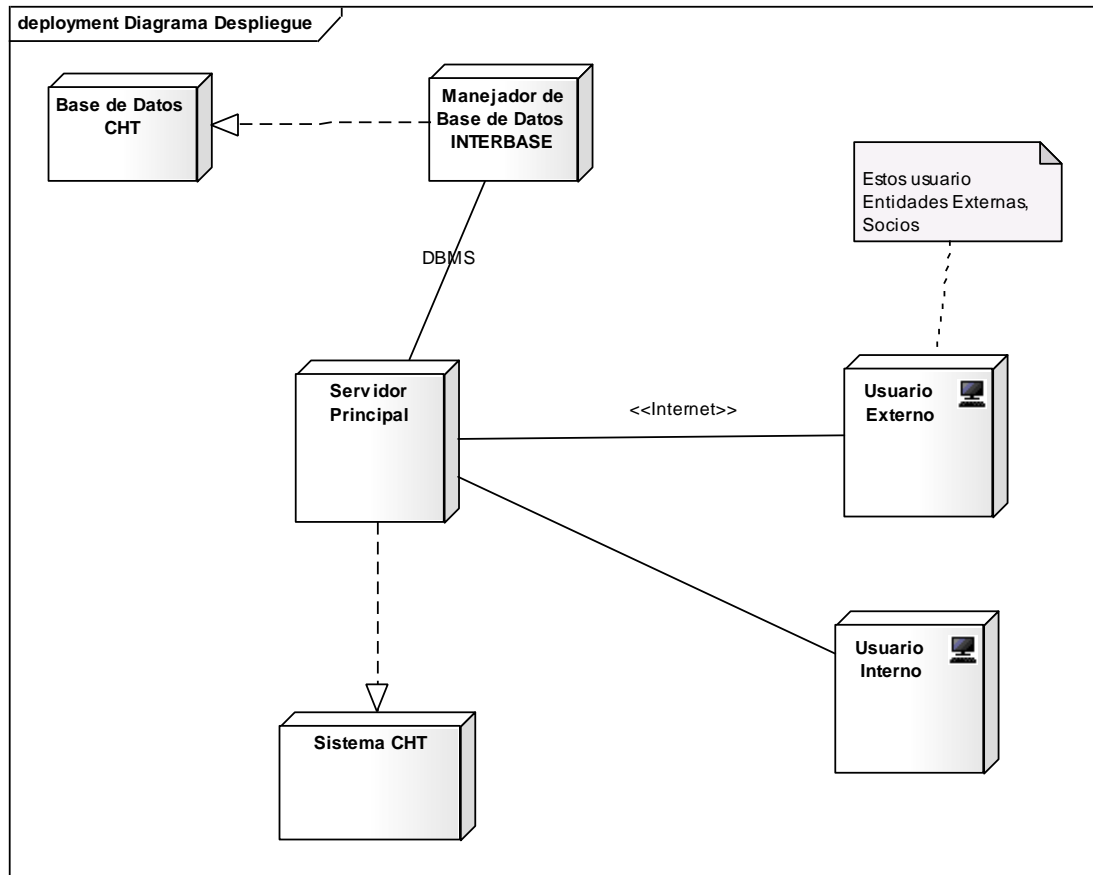
TABLA SOCIO_ESTABLECIMIENTO

```
CREATE TABLE SOCIO_ESTABLECIMIENTO (  
    ID_SOCIO INTEGER NOT NULL,  
    IDESTB INTEGER NOT NULL,  
    IDUSUARIO INTEGER NOT NULL,  
    F_AFILIACION DATE NOT NULL,  
    F_BAJA DATE,  
    CONSTRAINT PK_SOCIO_ESTABLECIMIENTO PRIMARY KEY (ID_SOCIO, IDESTB,  
IDUSUARIO)  
);
```

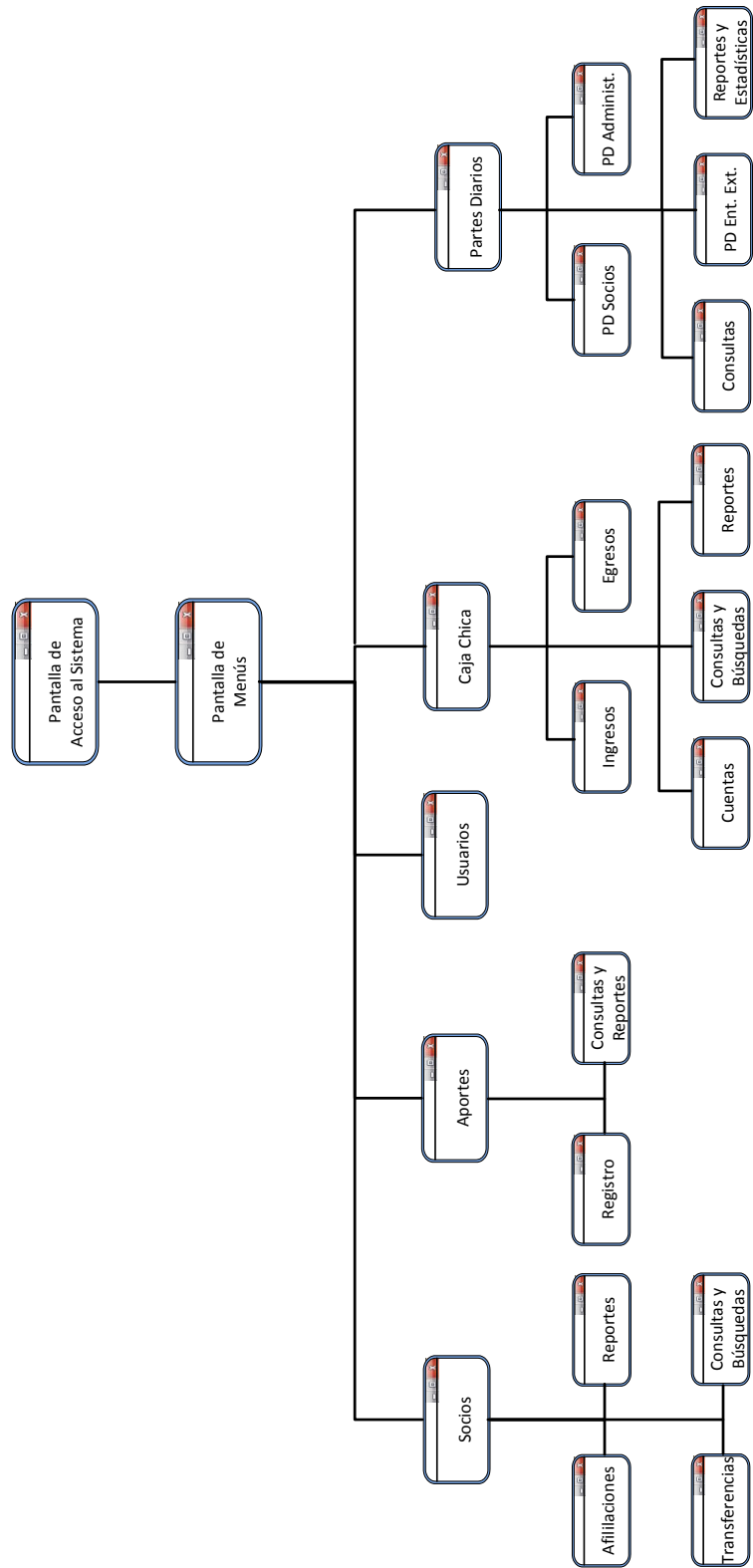
4.2. DIAGRAMA DE COMPONENTES



4.3. DIAGRAMA DE DESPLIEGUE



4.4. MAPA NAVEGACIONAL DE VENTANAS



4.5. DISEÑO DE PROTOTIPOS DE PANTALLAS DEL SISTEMA

Pantalla 1: Acceso al Sistema

The screenshot shows a web browser window titled 'Form2'. The page header for 'Camara Hotelera Tarija' includes the contact number 'La Madrid 00775 Telefon 66-34703'. The central focus is a login form titled 'Control de Acceso' with two input fields: 'Nombre de Usuario' and 'Contraseña', followed by an 'Ingresar' button.

Descripción de Entradas y Salidas			
Nombre de Entrada / Salida: Acceso al Sistema			
Tipo de Forma:	<input checked="" type="checkbox"/> Entrada	<input type="checkbox"/> Salida	
Formato de Entrada / Salida:	<input type="checkbox"/> Reporte	<input checked="" type="checkbox"/> Pantalla	<input type="checkbox"/> Otro
Tipo de Reporte:	<input type="checkbox"/> Interno	<input type="checkbox"/> Externo	
CONTENIDO: formulario de acceso al sistema.			
CAMPOS:			
Usuario: contiene el nombre de usuario que inicia sesión en el sistema.			
Clave: contiene la contraseña o clave de acceso al sistema.			
BOTONES:			
Ingresar: Permite iniciar sesión en el sistema.			
Descripción: pantalla principal del sistema de información automatizado para la gestión de la Cámara Hotelera de Tarija. El sistema presenta cuatro módulos: Gestión Socios, Caja Chica, Aportes y Partes Diarios. Cada uno de estos módulos con sus respectivos reportes y consultas. Para acceder a los módulos el usuario deberá introducir su usuario y contraseña, después de autentificarlos el sistema despliega el menú de opciones.			

Pantalla 2: Pantalla Principal del Sistema

Form1

**Camara
Hotelera Tarija**

La Madrid 00775 Telefon 66-34703

Inicio Salir

SISTEMA CAMARA HOTELERA TARIJA

Gestion Socios

Aportes

Caja Chica

Partes Diarios

Pág. 1 Sec. 1 1/5 A 2,9 cm Lín. 2 Col. 1 GRB MCA EXT SOB Español (Es)

Descripción de Entradas y Salidas			
Nombre de Entrada / Salida: Pantalla Principal Del Sistema			
Tipo de Forma:	<input checked="" type="checkbox"/> Entrada	<input type="checkbox"/> Salida	
Formato de Entrada / Salida:	<input type="checkbox"/> Reporte	<input checked="" type="checkbox"/> Pantalla	<input type="checkbox"/> Otro
Tipo de Reporte:	<input type="checkbox"/> Interno	<input type="checkbox"/> Externo	
CONTENIDO:			
Formulario principal del menú de opciones de acceso a los módulos del sistema.			
CAMPOS:			
BOTONES:			

Gestion Socios: permite Acceder al módulo de gestion socios.

Aportes: permite acceder al módulo Aportes usuario Tesorero.

Caja Chica: acceso al Modulo Caja chica usuario Tesorero

Partes Diarios: modulo accedido desde Internet usuario E.Externas, Socios

Descripción: pantalla que nos muestra una serie de opciones para poder ser accedidas a diferentes módulos del sistema, de acuerdo al usuario identificado previamente.

Comentario:

Pantalla 3: Ficha de Afiliación Socios

Form1
Gestion Socios

Sistema Camara Hotelera Tarija Usuario Nombre

Inicio Salir

Formulario de Afiliación

Datos del Socio

IdSocio 005 CI. 2345434

Nombre Apellido P

Direccion Apellido M

Fecha de Ingreso 01-09-10 Telefono 2345439

Estado Activo

Establecimiento

Tipo Establecimiento Hotel Categoria ****

IdEstble 3454567 Nombre El Sol Direccion La Madrid

Telefono 6623432 Fax 66-343234432 Pagina_Web http-elSol_hotel.com

Características Habitación

Simple 5 Matrimoniales 0

Dobles 3 Presidenciales 0

Triples 3

Otros Detalles Otras Características

Registrar Cancelar

Descripción de Entradas y Salidas			
Nombre de Entrada / Salida: Pantalla Afiliación			
Tipo de Forma:	<input checked="" type="checkbox"/> Entrada	<input type="checkbox"/> Salida	
Formato de Entrada / Salida:	<input type="checkbox"/> Reporte	<input checked="" type="checkbox"/> Pantalla	<input type="checkbox"/> Otro
Tipo de Reporte:	<input type="checkbox"/> Interno	<input type="checkbox"/> Externo	
CONTENIDO:			
Formulario Afiliación			
CAMPOS:			
<u>Campos con datos del Socio</u>			
IdSocio: contiene el identificador del socio			
Nombre: contiene el nombre del socio			
CI: contiene la CI del socio			
Apellido P: contiene el apellido paterno del socio			
Apellido M: campo que contiene el apellido materno del socio			
Dirección: contiene la dirección del socio			
Teléfono: campo que contiene el número de teléfono			
Fecha de ingreso: contiene la fecha de ingreso o fecha de afiliación a la institución			
Estado: contiene el estado del socio habilitado o inhabilitado.			
<u>Campos datos del Establecimiento</u>			
Tipo_Establecimiento: contiene al tipo de establecimiento este puede ser hotel, hostel, residencial o alojamiento.			
IdEstble: identificador del establecimiento.			
Categoría: contiene la categoría del establecimiento.			
Página Web: contiene la Url de la página web del establecimiento.			
Fax: contiene los datos de Fax de la establecimiento.			
Teléfono: contiene el número telefónico del establecimiento.			

Campos características de habitación

Simple: contiene datos tipo numérico del número de habitaciones simples.

Dobles: contiene número de habitaciones dobles.

Tripes: contiene número de habitaciones triples.

Matrimoniales: contiene número de habitaciones matrimoniales.

Presidenciales: contiene número de habitaciones presidenciales.

Otros detalles: contiene detalles adicionales a la habitación.

BOTONES:

Registrar: permite guardar los datos de los campo previa validación de los mismos.

Cancelar: se cancela toda operación de registrado de datos.

Descripción: pantalla que nos permite llenar una serie de campos requeridos para el registro de afiliación, tenemos el grupo de campos de datos del socio, grupo de campos pertenecientes al establecimiento y campos con características de habitaciones.

Comentario:

Pantalla 4: Reporte Afiliados Hoteles

Nr	Nombre	Direccion	Telefono	Categoria	Correo	F.Afiliacion
	Los Parrales	Zona Aranjues	6631248	*****	parrales@entelnet.bo	02/04/2009
12	Ceibos	La Madrid	6648444	****	Ceibo@hotmail.com	02/01/2009
13	Viña del Sur	Miraflores	6632425	****	VSur@entelnet.bo	02/01/2009
15	Victoria Plaza	Calle Sucre	6634430	***	VictPlazar@entelnet.	30/05/2009
16	Hotel Tarja	Calle Sucre	6642684	***	HTarja@hotmail.com	09/07/2009
17	El Sol	Av. Domingo Paz	6665259	***	SolH@hotmail.com	12/09/2009
18	Londres Orellano	C. Cbba	6642600	***	LondresH@hotmail.com	04/06/2009
19	Luz Plaza	Daniel Campos	6642369	***		15/10/2009
20	America	C.Bolivar	6635700	***		10/08/2009

Descripción de Entradas y Salidas			
Nombre de Entrada / Salida: Reporte Afiliados			
Tipo de Forma:	<input type="checkbox"/> Entrada	<input checked="" type="checkbox"/> Salida	
Formato de Entrada / Salida:	<input checked="" type="checkbox"/> Reporte	<input type="checkbox"/> Pantalla	<input type="checkbox"/> Otro
Tipo de Reporte:	<input checked="" type="checkbox"/> Interno	<input type="checkbox"/> Externo	
CONTENIDO:			
Reporte lista Afiliados Hoteles.			
CAMPOS:			
Nombre: Visualiza el nombre del socio(Establecimiento).			
Direccion: Visualiza la dirección.			
Telefono: Visualiza el número telefónico.			
Catg: Visualiza el tipo de categoría.			
Correo: Visualiza el correo electrónico			
F.Afiliacion: Visualiza la fecha de afiliación del establecimiento			
Descripción: Reporte Lista Hoteles Afiliados con sus respectivos datos			
Comentario:			

Pantalla 5: Reportes Afiliados Hostales

Reporte Afiliados						HOSTALES
Nr	Nombre	Direccion	Telefono	Categoria	Correo	F.Afiliacion
	Costanera	C-Cochabamba	6631248	****	Hoscostanera@hotmail	02/04/2009
12	Cristal	Urb. Carmen Aranjues	6648444	****	Cristal@hotmail.com	02/01/2009
13	Loma San Juan	Loma San Juan	6632425	***	Lsjuan@entelnet.bo	02/01/2009
15	Libertador	La Loma	6634430	***	Libertador@entelnet.	30/05/2009
16	Segovia	Calle Sucre	6642684	**	HSego@hotmail.com	09/07/2009
17	Carmen	C sucre	6665259	**	CarmenH@hotmail.com	12/09/2009
18	Gran Buenos Aires	Madrid esq Sucre	6642600	**	GBuenos@hotmail.com	04/06/2009
19	Miraflores	Daniel Campos	6642369	*		15/10/2009
20	Espejos	Domingo Paz	6635700	*		10/08/2009

Descripción de Entradas y Salidas			
Nombre de Entrada / Salida: Reporte Afiliados			
Tipo de Forma:	<input type="checkbox"/> Entrada	<input checked="" type="checkbox"/> Salida	
Formato de Entrada / Salida:	<input checked="" type="checkbox"/> Reporte	<input type="checkbox"/> Pantalla	<input type="checkbox"/> Otro
Tipo de Reporte:	<input checked="" type="checkbox"/> Interno	<input type="checkbox"/> Externo	
CONTENIDO:			
Reporte lista Afiliados Hostales			
CAMPOS:			
Nombre: visualiza el nombre del establecimiento (socio).			
Direccion: visualiza la dirección.			
Telefono: visualiza el número telefónico.			
Categoria: visualiza el tipo de categoría.			
Correo: visualiza el correo electrónico.			
F.Afiliacion: visualiza la fecha de afiliación del establecimiento.			
Descripción: reporte que visualiza una lista con los datos de los afiliados a la CHT.			

Pantalla 6: Reporte Afiliación Alojamiento

Reporte Afiliados						ALOJAMIENTOS
Nr	Nombre	Direccion	Telefono	Categ	Correo	F.Afiliacion
	Casa Grande	C-Cochabamba	6631248	Clase B		
12	Virgen de Lujan	Urb. Carmen Aranjues	6648444	Clase A		02/01/2009
13	Familiar	Zona Mirflores	6632425	Clase A		02/01/2009
15	Panamericano	La Loma	6634430	Clase A		30/05/2009
16	Rivera	Calle Sucre	6642684	Clase A		09/07/2009
17	La Economica	C sucre	6665259	Clase B		12/09/2009
18	El Hogar	Madrid esq Sucre	6642600	Clase B		04/06/2009
19	Imperial	Daniel Campos	6642369	Clase B		15/10/2009
20	Chapaquita	Domingo Paz	6635700	Clase B		10/08/2009

Descripción de Entradas y Salidas

Nombre de Entrada / Salida: Reporte Afiliados

Tipo de Forma:	<input type="checkbox"/> Entrada	<input checked="" type="checkbox"/> Salida
Formato de Entrada / Salida:	<input checked="" type="checkbox"/> Reporte	<input type="checkbox"/> Pantalla
Tipo de Reporte:	<input checked="" type="checkbox"/> Interno	<input type="checkbox"/> Externo

CONTENIDO:

Reporte lista Afiliados Alojamiento

CAMPOS:

Nombre: visualiza el nombre del establecimiento (socio).

Dirección: visualiza la dirección.

Teléfono: visualiza el número telefónico.

Categoría: visualiza el tipo de categoría.

Correo: visualiza el correo electrónico.

F. Afiliación: visualiza la fecha de afiliación del establecimiento.

Descripción: Reporte que lista los datos de los Alojamiento afiliados a la CHT.

Pantalla 7: Reporte Afiliados Características Habitación

Print Preview

Reporte Afiliados							Hoteles
Nombre	Cantidad Habitaciones				Suits		TotalHab
	Simple	Dobles	Triples	Otros	Matrimoniales	Presidencial	
Martinez Tarja	7	12	2	6			27
Los Parrales		24			1	2	27
Los Ceibos	8	14		10			32
Viña de Sur	25	28		14			78
Victoria Plaza	12	2	1	6			42
Salvador	5	10					15
Tarja	9	25	2	6			42
Luz Palace	3	18		2			23
El Sol	8	14	2		2		26
Londres Orellano	2	20	8				19
Totales	79	167	15	44	3	2	331

0% Page 1 of 1

Descripción de Entradas y Salidas

Nombre de Entrada / Salida: Reporte Afiliados

Tipo de Forma:	<input type="checkbox"/> Entrada	<input checked="" type="checkbox"/> Salida
Formato de Entrada / Salida:	<input checked="" type="checkbox"/> Reporte	<input type="checkbox"/> Pantalla
Tipo de Reporte:	<input checked="" type="checkbox"/> Interno	<input type="checkbox"/> Externo

<p>CONTENIDO:</p> <p style="text-align: center;">Reporte Características de Habitación Hoteles</p> <p>CAMPOS:</p> <p>Nombre: visualiza el nombre del establecimiento.</p> <p>Simples: visualiza el número de habitaciones simples.</p> <p>Dobles: visualiza el número de habitaciones dobles.</p> <p>Triples: visualiza el número de habitaciones triples.</p> <p>Matrimoniales: visualiza el número de habitaciones matrimoniales.</p> <p>Presidencial: visualiza el número de habitaciones presidenciales.</p> <p>TotalHab: visualiza el total de habitaciones del establecimiento.</p>
<p>Descripción: reporte que visualiza las características de las habitaciones como ser habitaciones simples, dobles, triples, matrimoniales, residenciales.</p>
<p>Comentario:</p>

Pantalla 8: Caja Chica

Nreg	Fecha	Monto	Cuenta	Detalle
1	02/01/2009	250	inscripcion	monto de la inscripc
2	03/01/2009	50	Gastos Oficina	
3	04/02/2009	300	Cursos	
4	04/02/2009	100	alquiler Oficina	

Descripción de Entradas y Salidas

Nombre de Entrada / Salida: Formulario Caja Chica

- Tipo de Forma: Entrada Salida
- Formato de Entrada / Salida: Reporte Pantalla Otro
- Tipo de Reporte: Interno Externo

CONTENIDO:

Pantalla Caja Chica detalle registros realizados.

CAMPOS:

BOTONES:

Nuevo: al elegir esta opción aparece la pantalla de registro de operaciones.

Modificar: llama a la pantalla modificar registros.

Consulta: llama a la pantalla consulta.

Buscar: llama a la pantalla buscar datos

Descripción: pantalla donde se visualiza el detalle de los ingresos y egresos operaciones de caja chica, al mismo tiempo tenemos botones de acceso a otras pantallas como adicionar un nuevo registro, modificar, buscar o realizar consultas.

Comentario:

Pantalla 9: Registro Caja Chica

The screenshot displays the 'Registro Caja Chica' interface within the 'Sistema Camara Hotelera Tarija' application. The window title is 'Form1'. On the left, there is a navigation menu with options: 'Caja Chica', 'Informes', 'Cuenta', 'Aportes', 'Reportes', and 'Consulta'. The main area features a header with the system name and user information ('Usuario:' and 'Nombre:'). Below the header, the form contains the following fields and controls:

- NroReg: 022
- Fecha: 08/04/2009
- Monto: 300
- Cuenta: (dropdown menu)
- Detalle: (dropdown menu with options: Ingreso, - Ventas, - Alquileres, - eventos)
- Buttons: Registrar (highlighted), Cancelar

Descripción de Entradas y Salidas			
Nombre de Entrada / Salida: Formulario Caja Chica Registro			
Tipo de Forma:	<input checked="" type="checkbox"/> Entrada	<input type="checkbox"/> Salida	
Formato de Entrada / Salida:	<input type="checkbox"/> Reporte	<input checked="" type="checkbox"/> Pantalla	<input type="checkbox"/> Otro
Tipo de Reporte:	<input type="checkbox"/> Interno	<input type="checkbox"/> Externo	
CONTENIDO:			
Pantalla Caja Chica Registro			
CAMPOS:			
NroReg: contiene los datos del número de registro o número de operación.			
Fecha: contiene los datos de la fecha en que se registró la operación.			
Monto: contiene el dato del monto registrado.			
Cuenta: contiene los datos del tipo de cuenta a registrar.			
Detalle: contiene detalles adicionales al registro.			
BOTONES:			
Registrar: realiza el guardado de los datos en las tablas respectivas.			
Cancelar: cancela la operación de registrado.			
Descripción: se realiza el registro de una nueva operación llenando los campos respectivos estos pueden ser ingreso o gasto.			
Comentario:			

Pantalla 10: Aportes

Descripción de Entradas y Salidas			
Nombre de Entrada / Salida: Pantalla Aportes			
Tipo de Forma:	<input checked="" type="checkbox"/> Entrada	<input type="checkbox"/> Salida	
Formato de Entrada / Salida:	<input type="checkbox"/> Reporte	<input checked="" type="checkbox"/> Pantalla	<input type="checkbox"/> Otro
Tipo de Reporte:	<input type="checkbox"/> Interno	<input type="checkbox"/> Externo	
CONTENIDO:			
Pantalla Aportes			
CAMPOS:			
Socio: contiene los datos del identificador del socio.			
Nombre: campo que contienen el nombre del socio.			
Apellido: contiene el apellido del socio.			

Cuenta: contiene los datos del tipo de cuenta a registrar.

NroReg: contiene el número de registro de aporte.

Fecha: contiene la fecha de registro de aporte.

Monto: contiene el monto registrado del aporte.

BOTONES:

Inicio: retorna al formulario de inicio

Salir: cierra la pantalla de aportes

Registrar: realiza el guardado de los datos del aporte en las tablas respectivas

Cancelar: cancela la operación de registrado de aporte

Descripción: se realiza el registro de un nuevo aporte llenando los campos respectivos

Comentario:

Pantalla 11: Consulta Aportes

SISTEMA CAMARA HOTELERA TARIJA

Usuario: _____ Fecha: _____

[Inicio] [Salir]

Resumen de Socios que Aportaron

Establecimiento: Hoteles Fecha: Edt4 [Consultar]

	Establecimiento	Fecha	NroAporte
<input checked="" type="checkbox"/>	Hotel Carmen Zona Central	1-sep	0-1212
<input checked="" type="checkbox"/>	Hotel El Sol La Loma	3-sep	0-1213
<input type="checkbox"/>	Hotel Londres C Cochabamba		
<input checked="" type="checkbox"/>	Hotel El Salvador Calle La paz	8-sep	0-1215
<input checked="" type="checkbox"/>	Hotel Zeballos Domingo Paz	16-sep	0-1216
<input checked="" type="checkbox"/>	Hotel America Bolivar	17-sep	0-1217
<input checked="" type="checkbox"/>	Hotel Los Ceibos Zona Aranjues	17-sep	0-1218
<input checked="" type="checkbox"/>	Hotel Los Pinales Zona Aranjues	20-sep	0-1219
<input checked="" type="checkbox"/>	Hotel America Zona Central	17-sep	0-1217
<input type="checkbox"/>	Hotel Tarija Zona El Molino		
<input type="checkbox"/>	Hotel Max		

Windows Taskbar: Inicio, T-1374.pdf - A..., Delphi 7, Delphi 7 - Modu..., ModuloJotesorero, TrabjDirigidoCH, Ficha de Afiliad..., 10:14 p.m.

Descripción de Entradas y Salidas			
Nombre de Entrada / Salida: Pantalla Consulta Aportes			
Tipo de Forma:	<input type="checkbox"/> Entrada	<input checked="" type="checkbox"/> Salida	
Formato de Entrada / Salida:	<input type="checkbox"/> Reporte	<input checked="" type="checkbox"/> Pantalla	<input type="checkbox"/> Otro
Tipo de Reporte:	<input type="checkbox"/> Interno	<input type="checkbox"/> Externo	
CONTENIDO:			
Pantalla Consulta Aportes			
CAMPOS:			
Establecimiento: contiene los datos del identificador del establecimiento.			
Fecha: contiene la fecha de consulta.			
BOTONES:			
Inicio: retorna al formulario de inicio.			
Salir: cierra la pantalla de consulta de aportes.			
Consultar: realiza la consulta a la base de datos.			
Descripción: se realiza el registro de un nuevo aporte llenando los campos respectivos.			
Comentario:			

Pantalla 12: Parte Diario Socios

Parte Diario Socios
Registro Parte diario

PARTE DIARIO DE PASAJEROS
Usuario: Tesorero
Nombre: Marcelo P

Establecimiento: HOTEL Nombre Establecimiento: El Sol
Direccion: Calle Sucre 782
Tipo de Pasajeros: Permanentes Fecha Parte Diario: 12-05-2009

Nombre	Apellido	NroHab	Nacionalidad	Ciudad	Edad	Profesion	Documento
Carlos	Costas	201	Arg	Salta	45	Abogado	34323443
Alejandra	Facuteu	207	Arg	Cordoba	30	Funcionaria	2445433443
Marioly	Rosado	208	Bol	La Paz	27	Ingeniera	1140786
Yery	Camacho	309	Bol	CBBA	20	Administrador	2399132

Registrar Limpiar

Descripción de Entradas y Salidas

Nombre de Entrada / Salida: Pantalla Registro Parte Diario

- Tipo de Forma: Entrada Salida
- Formato de Entrada / Salida: Reporte Pantalla Otro
- Tipo de Reporte: Interno Externo

CONTENIDO:

Pantalla Registro de Parte Diarios

CAMPOS:

Establecimiento: contiene los datos del tipo de establecimiento.

Nombre del Establecimiento: campo que contienen el nombre del establecimiento.

Dirección: contiene la dirección del establecimiento.

<p>Tipo de Pasajeros: contiene los datos del tipo de pasajeros (ingresantes, salientes, permanentes)</p> <p>Nombre: contiene el nombre del pasajero.</p> <p>Apellido: contiene el apellido del pasajero.</p> <p>Fecha Parte Diario: contiene la fecha del parte diario.</p> <p>Nro Habitación: contiene el número de habitación registrado del pasajero.</p> <p>Nacionalidad: contiene la nacionalidad del pasajero.</p> <p>Ciudad: contiene la ciudad de origen del pasajero.</p> <p>Edad: contiene la edad del pasajero registrado.</p> <p>Profesión: contiene la profesión del pasajero registrado.</p> <p>Documento: contiene el registro de su documento (CI, Pasaporte).</p> <p>BOTONES:</p> <p>Registrar: realiza el guardado de los datos del parte diario en tablas respectivas.</p> <p>Limpiar: limpia los campos Cancela la operación de registrado de aporte.</p>
<p>Descripción: realiza el registro de un parte diario por parte de los socios haciendo uso de los campos respectivos.</p>
<p>Comentario:</p>

Pantalla 13: Consulta Parte Diario General

Features Demo - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://127.0.0.1:1700/EXEC/1/4C4AAD00565474553FC7E340#

Google

Parte Diario Socios

Consulta Parte Diario

Parte Diario General Usuario: Nombre:

Fecha Parte Diario Mes: Febrero Año: 2009

Hoteles
 Residenciales
 Hostales
 Alojamientos

Generar Cancelar

Hotel
Los Ceibos

Bolivia		Argentina		Brasil		Colombia		Chile		Ecuador		Paraguay		Peru		Venezuela	
I	P	I	P	I	P	I	P	I	P	I	P	I	P	I	P	I	P
8	11	4	2	3	4	3	2	3	5	6	5	4	3	2	3	5	
6	3	6	8	2	9	7	4	5		4	6	8	7	2	2	5	4
8	4	3	6	2	2	2	5	4	3	7	6	4	3	7	1	9	6
12	4	7	1	1	1	6	8	7	4	9	3	6	11	6	6	8	
33	6	22	7	6	5	3	9	8	5	3	5	8	5	8	9	4	

Imprimir

Pág. 12 Sec. 1 12/12 A 2,4 cm Lín. 1 Col. 39 GRB MCA EXT SOB Español (Es)

Pantalla 13: Consulta parte diario Entidades Externas

Features Demo - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://127.0.0.1:1392/EXEC/1/343EAA00EE5552492BC8E340#

Google

Parte Diario Socios

Consulta E.Externas

Usuario:
Nombre:

Fecha P.Diario:

Establecimiento: Hoteles Hostales Residenciales Alojamientos

Nacionalidad:

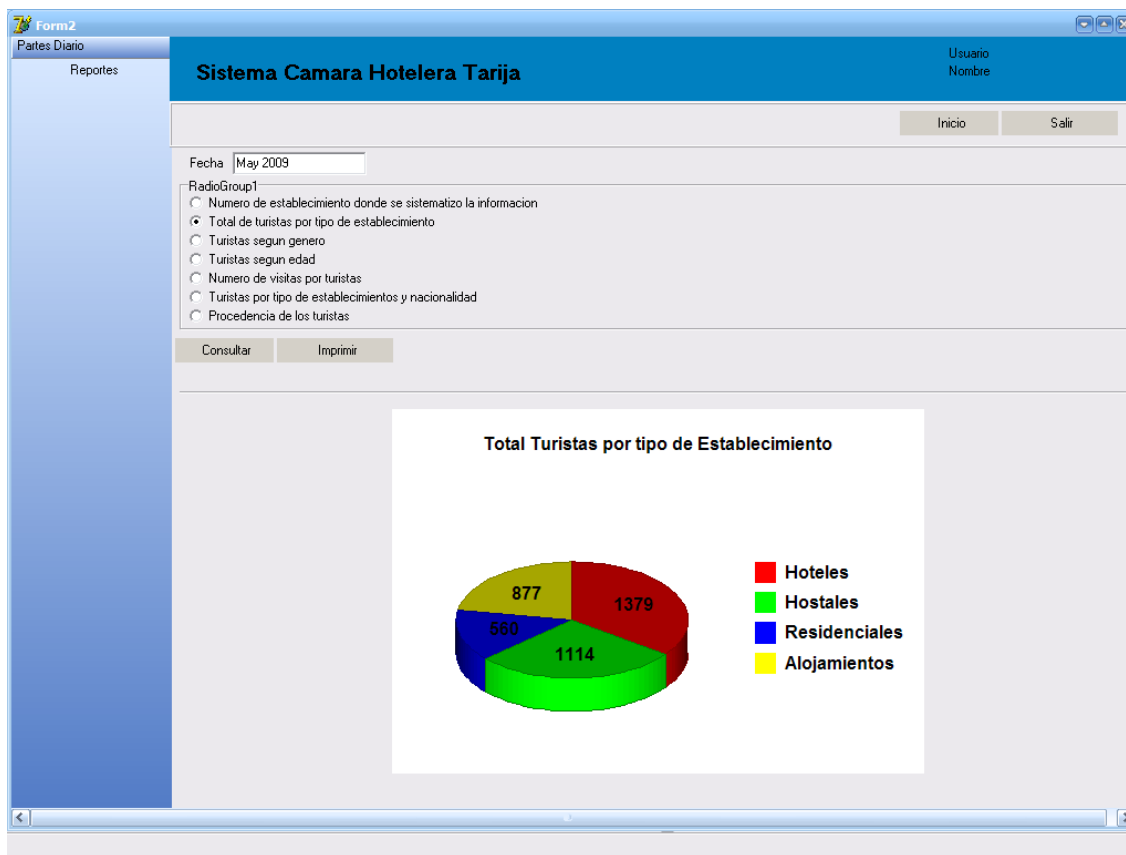
Sexo:

Procedencia:

⏪ ⏩ ⏴ ⏵

Ceibos		Los Parrales		El Sol		Victoria		Cristal		Londres		Ame	
I	P	I	P	I	P	I	P	I	P	I	P	I	P
All	All	All	All	All	All	All	All	All	All	All	All	All	All
8	11	4	3	5	4	7	6	4	5	6	6	7	
5	11	3	4	9	4	2	6	4	11	4			
1,6	11	4	3	5	4	7	6	4	5	6	6	7	
8	11	4	7	5	4	3	6	4	5	6	6	7	
2	4	4	3	4	7	6	4	5	6	7			
6	11	4	3	5	4	7	6	4	5	6	6	7	

Pantalla 14: Cuadros Estadísticos



CONCLUSIONES
Y
RECOMENDACIONES

5.1. CONCLUSIONES

Al finalizar el presente trabajo se puede concluir lo siguiente:

- Se identificó claramente los requisitos de información de la Cámara Hotelera de Tarija, utilizando entrevistas, revisión de documentación y la observación, documentando el análisis de requisitos empleando el método ISAC.
- La metodología RUP (Proceso Unificado Rational) permite realizar varias iteraciones para lograr un mejor desarrollo del trabajo, debido a que es un proceso iterativo e incremental, se documenta solamente la última iteración en cada fase de análisis y diseño.
- Se diseñó la base de datos acorde a las necesidades identificadas y planteadas en el análisis, documentando el diagrama de base de datos, estructura y código SQL de la creación de las tablas.
- Se realizó el diseño de los prototipos de interfaces de pantallas y reportes de acuerdo a los requerimientos de los usuarios.

5.2. RECOMENDACIONES

- Se recomienda tomar en cuenta la documentación del presente trabajo para realizar la programación e implementación del sistema.
- Se recomienda la ampliación del presente trabajo, añadiendo nuevos módulos, o reportes que se consideren necesarios, como por ejemplo un módulo contable.