

**PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAISO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA**

**“SITIO WEB Y SISTEMA DE RESERVA DE
CANCHAS PARA EL CLUB INGLES DE TENIS DE
VIÑA DEL MAR”**

MIGUEL ANTONIO CIFUENTES GODOY

**INFORME FINAL DEL PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO DE EJECUCIÓN EN INFORMÁTICA**

Julio de 2008

1 INTRODUCCION

La ingeniería del Software es la disciplina de ingeniería encargada de todos los aspectos relacionados con la producción de software desde sus etapas más tempranas de la especificación del sistema hasta el mantenimiento del sistema tras su puesta en marcha

Los fundamentos de un buen sistema o de alta calidad es aquel que cumple con las necesidades del usuario, aprovechando su utilidad real. Un buen software hace que la vida de los usuarios sea más fácil o mejor al optimizar procesos complejos con una mínima tasa de errores; pero sin dejar de ser flexible, accesible y con una alta disponibilidad. Todas estas, características deseables están presentes en el proceso unificado de software UP.

El Proceso Unificado esta basado en componentes, lo que quiere decir que el sistema en construcción esta formado por componentes de software interconectados a través de interfaces bien definidas. El Proceso Unificado utiliza el nuevo estándar de modelo visual, el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema de software.

Los aspectos relevantes del Proceso Unificado se resumen en tres claves: Dirigido por casos de uso, Centrado en la arquitectura, Iterativo e incremental.

Bajo estos preceptos la organización sin fines de lucro denominada “Club Inglés de tenis”, de la ciudad de Viña del Mar requiere la construcción de un sistema informático que incluya los preceptos antes definidos y que permita a los socios y la administración del club conocer, ordenar y agregar su información personalizada, lo cual implican un gran avance en la consolidación de una nueva etapa dentro de la organización; esto es la extensión a nivel global del club Inglés de tenis.

Toda la documentación técnica del presente proyecto ha sido desarrollada utilizando el proceso unificado de desarrollo de software (UP), el cual tiene varias etapas de desarrollo, incluidas las etapas transición y pruebas; es decir una etapa de aplicación práctica de toda la documentación desarrollada en las etapas anteriores.

2 OBJETIVOS DEL PROYECTO

2.1 OBJETIVO GENERAL

Diseñar e implementar un sitio Web informativo y promocional para la organización, el cual incluye un sistema de reserva de canchas y consulta de estados de pago, como la mejor solución informática a través de la cual se podrán hacer consultas online resguardadas por validaciones de sesión de usuario, lo cual permitirá satisfacer las necesidades de gestión e información de los socios.

2.2 OBJETIVOS ESPECIFICOS

- Construcción de prototipos e interfaces de ingreso de datos, con el fin de ayudar a los clientes y los desarrolladores a entender los requerimientos del sistema.
- Validación de los prototipos, según los requerimientos, los cuales servirán como base para la especificación del sistema.
- Estudiar en profundidad el lenguaje de Modelado UML, con el fin de desarrollar y diseñar el proyecto de software, aplicando patrones de diseño modernos para la construcción de la aplicación.
- Codificación del producto final, accesible y usable para los usuarios, de acuerdo a la validación de los prototipos y los parámetros técnicos establecidos como patrones de desarrollo.

3 ESTADO DEL ARTE

3.1 ORIGEN DEL PROBLEMA

El problema de flujo de la información en club inglés de tenis, tiene su génesis en la urgente necesidad de tener un medio de comunicación directo entre la administración y directiva de la organización y sus socios y simpatizantes.

Se necesita estandarizar los mensajes acerca de reuniones, noticias, eventos y avisos propios del club, entre ellos los de morosidad en los pagos de mensualidades.

Actualmente, el club cuenta con un sitio Web, el cual está implementado en forma muy artesanal y no cumple ninguna función, más que mostrar información general del club de tenis, la cual no representa lo mejor que el club inglés de tenis puede ofrecer a cibernautas que potencialmente pueden ser futuros nuevos socios.

La información, hoy en día, es traspasada a los socios en forma verbal – cuando ellos asisten al club – o por el contrario, a través de contacto telefónico, lo cual incrementa considerablemente los costos de operación del club.

A manera de resumen, se pueden mencionar los siguientes conflictos y confusiones en la entrega de la información propia del club y que hoy ocurre.

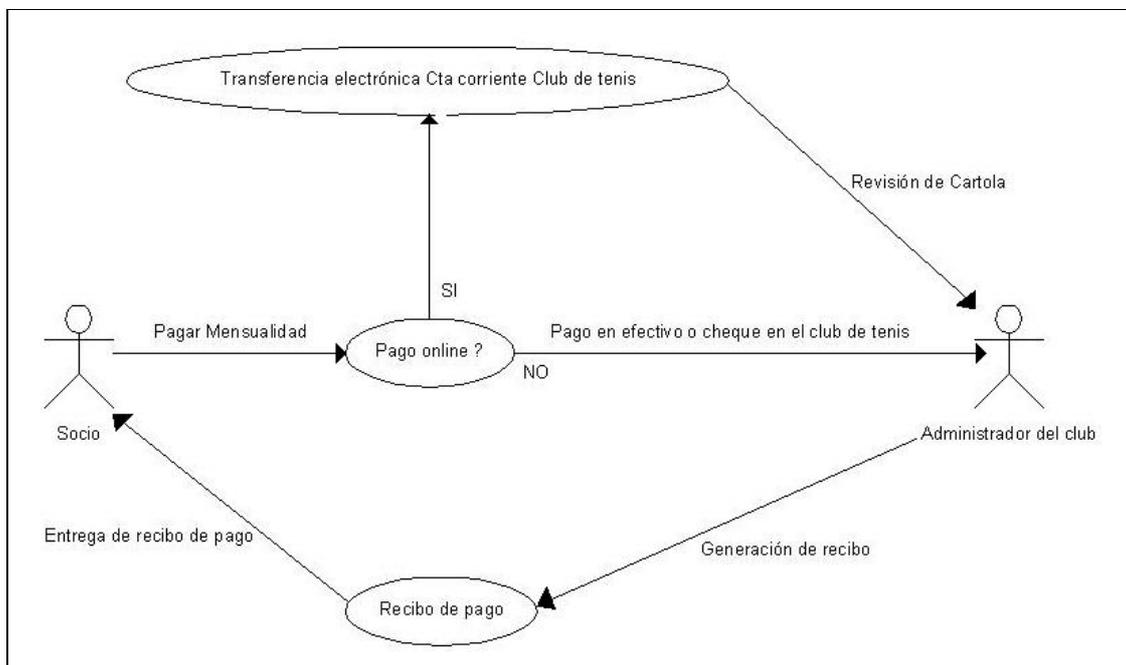
- Confusión en torno a fechas y horarios de reuniones.
- Desconocimiento por parte de algunos socios acerca de invitaciones y torneos a realizarse dentro y fuera del club.
- Alto costo operacional, debido a llamadas telefónicas para recordar pagos de mensualidades o avisos varios.
- Sitio Web poco funcional.
- Avisos de morosidad hechos en forma verbal, o por email; pero si un estándar definido.

3.2 CONOCIMIENTO DEL PROBLEMA

El problema de flujo de información y comunicación es el talón de Aquiles de la organización. Además ellos necesitan acercar a más socios con el fin de solventar todos los gastos operacionales del club y junto con ello poder invertir en mejoras de la infraestructura, canchas, rejas, casino, áreas verdes, duchas, etc.

Existen varios “modus operandi” que actualmente se utilizan para realizar pagos, reservar canchas, comunicación de eventos.

Para pagos de mensualidades:

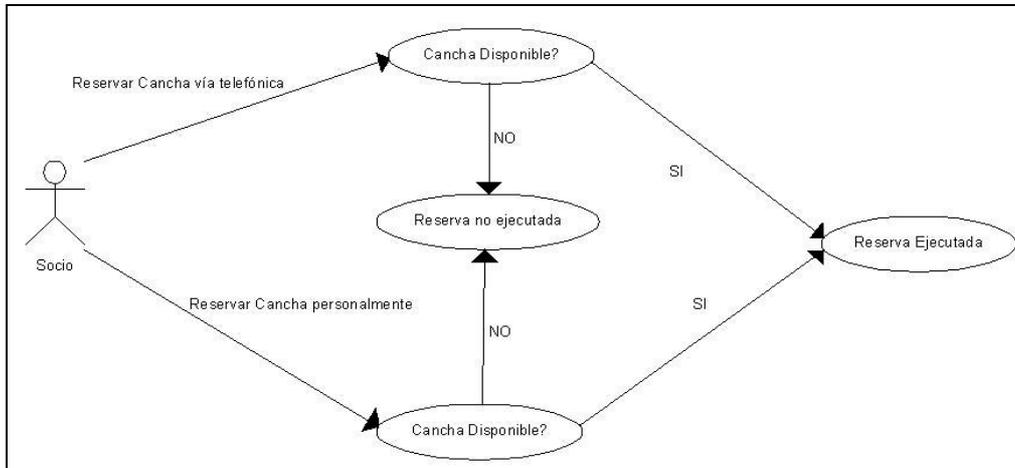


Esquema 1.1 Formas de Pago de las mensualidades de socio

Problemas detectados:

- Se debe dedicar mucho tiempo en la revisión de cartolas de cuenta corriente.
- No hay un registro cronológico de pagos que permita generar listados o informes de pagos en forma rápida
- Se pierde mucho tiempo al revisar los estados de pago de los socios y generar listados de aquellos que están morosos.
- Pérdida de recursos (tiempo y dinero) por el hecho de tener que llamar a socios morosos en los pagos, con el fin de que se pongan al día.

Para reserva de canchas de tenis:

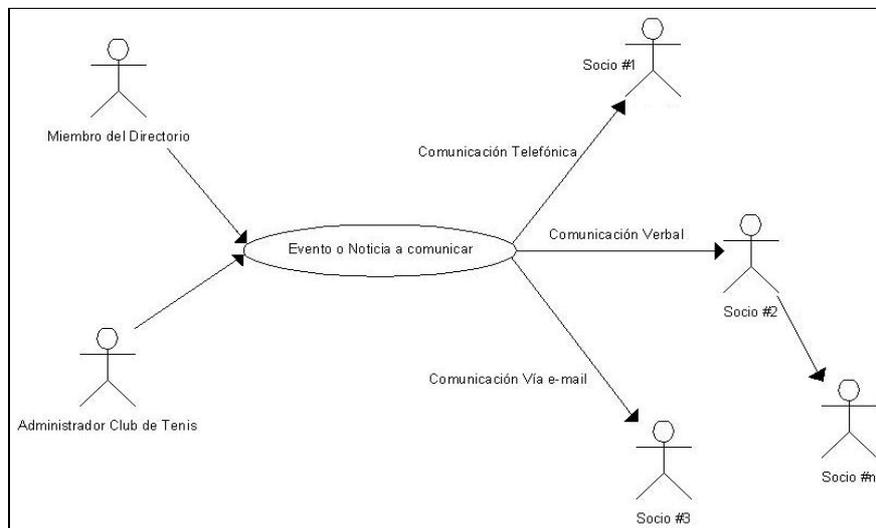


Esquema 1.2 Formas de reservar cancha

Problemas detectados:

- No hay una norma que defina el procedimiento regular para reservar canchas en fecha y hora determinada.
- Se han producido confusiones en la reserva de canchas, ya que un socio, habiendo reservado para un día y hora determinado, se ha encontrado con que su espacio está siendo ocupado por otros socios.

Para comunicación de eventos



Esquema 1.3 Formas de comunicar eventos y/o noticias entre socios

Problemas detectados:

- Pérdida parcial del mensaje o tergiversación del mismo.

4 METODOLOGIA

Para el cumplimiento de los objetivos del proyecto se deben realizar las siguientes actividades:

Análisis de Factibilidad:

Este análisis contempla el estudio de los distintos aspectos que influyen en el desarrollo del proyecto para ver si este es factible de ser realizado. La factibilidad del proyecto se debe ver desde los siguientes puntos de vista: técnica, económica, operacional y legal. Si el proyecto es factible de realizar desde todos estos aspectos; entonces se puede continuar con su implementación.

Reuniones con el cliente:

Se han organizado reuniones periódicas con el cliente para interiorizarse acerca de las normas, reglamentos y procedimientos que la organización utiliza. Se entiende como cliente a un conjunto de personas con distintos roles dentro de la organización, entre ellos están:

- Administrador del club de tenis
- Miembros del directorio
- Concesionario del casino
- Socios
- Encargado de manutención de canchas.

Además se hace uso de correo electrónico para comunicarse con el administrador del club de tenis para despejar dudas y obtener respuestas rápidas con respecto a algún tema en particular.

Identificación de riesgos del proyecto y su plan de contingencia:

Esta actividad comienza en las etapas iniciales del proyecto y se desarrolla a lo largo de todo su ciclo de vida (hasta la aceptación del producto por parte del cliente). Está compuesto por varias etapas bien documentadas, estas son: Identificación de riesgos, análisis, planificación, seguimiento y control.

Elección de paradigma y metodología a utilizar en el proyecto:

Se evalúan algunos paradigmas en el desarrollo de software y según la naturaleza del sistema a construir se optará por uno de ellos. Se tienen en cuenta los siguientes paradigmas: desarrollo iterativo y creciente, proceso racional unificado (RUP), modelo en cascada, desarrollo a través de prototipos y Modelo en V.

Evaluación de la arquitectura del sistema

El propósito de realizar evaluaciones a la arquitectura, es para analizar e identificar riesgos potenciales en su estructura y sus propiedades, que puedan afectar al sistema de

software resultante, verificar que los requerimientos no funcionales estén presentes en la arquitectura, así como determinar en qué grado se satisfacen los atributos de calidad

Herramientas de software utilizadas:

Se especifican las herramientas de software que se van a ocupar para cada una de las etapas del proyecto.

Levantamiento y análisis de requerimientos:

Los requerimientos son atributos del sistema; es decir algo que el sistema debe ser capaz de hacer para cumplir su propósito. Estos requerimientos se denominan “funcionales”. Se busca establecer con absoluta claridad qué es lo que el cliente realmente pide

Identificación y especificación de actores del sistema y casos de uso:

Un caso de uso puede ser definido como una secuencia de acciones, incluyendo variaciones, que el sistema puede ejecutar y que produce un resultado observable. Los actores son los que interactúan con el sistema valiéndose de los casos de uso.

Modelamiento de la solución con UML:

UML es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

Modelamiento de la base de datos:

El Modelado de la base de datos es la etapa en donde se identifica y dibuja el conjunto de datos que el usuario requiere en un sistema de información. Esta actividad contempla el diseño de la base de datos desde dos aspectos: lógico y físico.

Desarrollo de prototipos de interfaces bajo estándares de usabilidad:

Se crearán prototipos de interfaces con el fin de presentarlas ante los distintos perfiles de usuario para que ellos entreguen su opinión crítica acerca del trabajo realizado.

Se contemplan por lo menos 2 técnicas para evaluación de usabilidad del sistema, las cuales pueden ser evaluaciones heurísticas y pruebas de usabilidad con usuarios reales.

5 ELABORACION

5.1 ANALISIS DE FACTIBILIDAD

En general, los estudios de factibilidad, se completan durante la fase de diseño de sistemas, en general durante la consideración de la evaluación de las diferentes alternativas de solución propuestas. Los estudios de factibilidad consideran la factibilidad técnica, económica y operacional, así como si el proyecto es o no apropiado dados los factores políticos y/o legales, además de otros en el contexto institucional.

5.1.1 FACTIBILIDAD TÉCNICA

El análisis de factibilidad técnica evalúa si el equipo y software están disponibles (o, en el caso del software, si puede desarrollarse) y si tienen las capacidades técnicas requeridas por cada alternativa del diseño que se esté considerando.

5.1.1.1 Software

- Software para desarrollo de informes y documentos: Microsoft Office 2003
- Software para modelado del sistema y diagramas: Microsoft Visio 2003, Poseidon
- Software para modelado de la base de datos: Deziq 4.0
- Software para codificación del sistema: Framework CakePHP, editores PHP
- Software de motor de bases de datos: PostgreSQL
- Software para servidor Web: Apache
- Software para Sistema Operativo: GNU/Linux

Cabe mencionar que el Framework CakePHP posee licencia MIT, la cual es open source. Por otro lado el software de sistema operativo, base de datos y servidor Web también poseen este tipo de licencia.

5.1.1.2 Hardware

- Para las etapas de análisis de requisitos, diseño, construcción del sistema e implantación de versiones beta y prototipos; se cuenta con el siguiente hardware:
Un computador Pentium IV, 1 GB memoria RAM, 80 Gb de capacidad de almacenamiento en disco duro, conectado a Internet.
- Hardware para implantación de sistema en producción, parte servidor: Servidores de la empresa de hosting contratada por el club Inglés de tenis
- Hardware para implantación de sistema en producción, parte cliente. El club de tenis cuenta con un computador Pentium IV, 256 MB memoria RAM, 80 Gb de capacidad de almacenamiento en disco duro, conectado a Internet vía red Wi-Fi.

5.1.2 FACTIBILIDAD ECONOMICA

5.1.2.1 Valorización horas hombre del proyecto

Fase del proyecto	Duración (Semanas)	HH Estimadas	Valor unitario HH	Total
Inicio	7 semanas	210	\$ 2.500	\$ 525.000
Elaboración	10 semanas	300	\$ 2.500	\$ 750.000
Construcción	9 semanas	270	\$ 2.500	\$ 675.000
Transición	6 semanas	180	\$ 2.500	\$ 450.000
			Costo HH Proyecto	\$ 2.400.000

Cuadro 1.1 Valorización de horas hombre del proyecto

5.1.3 COSTOS FIJOS Y DE MANTENCIÓN DEL PROYECTO

Items	Costo/mes
Conexión Internet	\$ 15.000
Dominio .CL	\$ 1.500
Electricidad	\$ 30.000
Hosting	\$ 20.000
Total	\$ 66.500

Cuadro 1.2 Costos fijos del proyecto

5.1.4 INGRESOS MENSUALES PROMEDIO DEL CLUB INGLES

- Ingresos netos promedio al mes del club de tenis = \$ 4.125.000

- Porcentaje que representa el gasto computacional en el club de tenis con respecto al ingreso neto = 1.6 %

En conclusión, este gasto es muy reducido (menos del 2%, por lo cual el proyecto cuenta con la factibilidad económica.

5.1.5 FACTIBILIDAD OPERACIONAL

Esta factibilidad comprende una determinación de la probabilidad de que el nuevo sistema se use como se supone. Desde este punto de vista se consideran los siguientes aspectos:

- **El nuevo sistema puede ser demasiado complejo para los usuarios de la organización:**

Este problema operacional no aplica, ya que la interfaz Web del software será de fácil utilización, permitiendo a los distintos actores poder interactuar sin mayores problemas con el software. Para esto se tiene presente, como una importante actividad dentro del proyecto, la evaluación de usabilidad.

- **El sistema puede ser resistido por los usuarios:**

El sistema no debería tener este problema de uso, ya que el directorio del club de tenis ha informado a los socios de la construcción del software, con el cual se agilizarán las tareas de información desde y hacia los socios y demás actores de la organización.

- **Probabilidad de obsolescencia del sistema.**

El sistema tiene muy poca probabilidad de obsolescencia, ya que por una parte, no existe en este momento un sistema de información administrativo dentro de la organización que realice las funciones que el presente proyecto de software pretende satisfacer.

La obsolescencia está centrada en la forma en que hoy trabaja el club de tenis, ya que al no tener procedimientos estandarizados con respecto a pagos, arriendos, reservas y notificaciones, su funcionamiento se hace trabajoso, principalmente para el administrador del club de tenis.

5.1.6 FACTIBILIDAD LEGAL

El Viña del Mar Lawn tennis club, desde el punto de vista legal es una Corporación sin fines de lucro y también es un club deportivo sin fines de lucro, por lo cual no puede percibir ingresos económicos derivados de su actividad normal. Esto está normado a través del **REGLAMENTO SOBRE CONCESION DE PERSONALIDAD JURIDICA A CORPORACIONES Y FUNDACIONES (DTO-110, 1979, MINISTERIO DE JUSTICIA)** (Ver anexo E).

Debido a esta “figura legal”, la organización se encuentra estudiando una solución a la restricción legal para el arriendo de las canchas. Esta solución va por el camino de la concesión de las canchas de tenis, junto con la concesión ya existente del casino.

El directorio del club de tenis, está estudiando la firma de un nuevo contrato de concesión del casino que además incluya la concesión de las canchas de tenis, el cual permitiría la solución legal a esta restricción, ya que sería el concesionario, como persona jurídica, quien arrendaría las canchas de tenis, entregando boletas propias.

El directorio del club de tenis se encuentra negociando, en estos momentos, esta propuesta y un nuevo contrato con el actual concesionario del casino, por lo cual esta situación no sería una limitante de tipo legal para la realización del proyecto de software.

El actual concesionario ha manifestado su entusiasmo respecto de este nuevo contrato, ya que para él también significaría un ingreso económico extra.

5.2 ANALISIS DE RIESGOS

La identificación de riesgos en proyectos de software consiste en la determinación de elementos de riesgos potenciales mediante la utilización de algún método consistente y estructurado; este es, probablemente, el paso más importante entre todos aquellos que componen las actividades de Administración de Riesgos, ya que sin la correcta determinación de los mismos, no es posible desarrollar e implementar anticipadamente respuestas apropiadas a los problemas que puedan surgir en el proyecto [Futrell, *et al.*, 2002]. El resultado de la identificación de riesgos es una lista conteniendo los riesgos que se han identificados y su categoría correspondiente.

“Si los riesgos del proyecto se hacen realidad, es probable que la planificación temporal del proyecto se retrase y que los costos aumenten.” [3].

En el presente proyecto se aplicará la metodología RSGR [4] (reducción, supervisión y gestión del riesgo), la cual propone un enfoque metodológico para el riesgo, su control y mitigación.

Los riesgos se clasificarán de acuerdo a su probabilidad de ocurrencia y el impacto que ellos tendrían en el proyecto, dada la ocurrencia de los mismos.[4]

Probabilidad de que el riesgo ocurra: caso de concretarse:	Impacto que el riesgo provocaría en
---	--

Muy baja (< 10%)	Despreciable: No hay reducción en rendimiento
Baja (10 – 25%)	Marginal: Planificación realista y alcanzable
Moderada (25 – 50%)	Crítica: Reducción en rendimiento y retrasos en entrega
Alta (50 – 75%)	Catastrófica: Muy bajo rendimiento, grandes retrasos
Muy alta (> 75%).	Gran probabilidad de fracaso del proyecto

5.2.1 IDENTIFICACION DE RIESGOS DEL PROYECTO

Se identifican los siguientes tipos de riesgos en este proyecto de software, de acuerdo a la clasificación que se hace en el libro *“Ingeniería del software. Un enfoque práctico, quinta edición”*.

- Riesgos del negocio: Impacto negativo del software en el negocio, requerimientos captados no corresponden a los que el cliente pide, cambios de última hora en los requerimientos, bajo interés en usar el sistema, poco compromiso de la organización,
- Riesgos técnicos: Error en el dimensionamiento del producto, inadecuado entorno de desarrollo, poca experiencia del equipo de desarrollo, problemas de usabilidad del sistema, mayor número de usuarios de lo previsto usarán el software, problemas de hosting, problemas de seguridad (hacking).
- Riesgos económicos: Sobre valoración o sub-valoración del producto.

5.2.2 PLAN DE RSGR DEL PROYECTO

Tipo de riesgo: Del negocio	Probabilidad: Moderada	Impacto: Marginal
Riesgo: Impacto negativo del software en el negocio		
<p>Descripción:</p> <p>El software o sistema a construir podría generar un efecto contrario al esperado; esto es apatía y rechazo al uso del software debido a la inexperiencia relativa de algunas personas vinculadas al club de tenis; especialmente de socios con un rango etáreo más avanzado. Se calcula – según antecedentes entregados por el administrador del club de tenis - que el número de personas de la organización que no tienen afinidad o experiencia en el uso de computador y/o Internet es bajo; es decir menor a 10%</p>		
<p>Reducción del riesgo:</p> <ul style="list-style-type: none"> - Trabajar en una estrategia comunicacional y de difusión junto al administrador del club de tenis y el directorio para informar a los socios acerca de los beneficios que tendrá la implementación del nuevo sistema. Esto se puede hacer a través de algunos afiches que se pueden colocar en dependencias del club de tenis, a través de correos electrónicos y con notas informativas colocadas en el actual sitio Web del club 		
<p>Plan de contingencia:</p> <ul style="list-style-type: none"> - Construir un prototipo que demuestre en forma práctica parte de las funcionalidades que el futuro sistema tendrá. En lo posible demostrarlo ante un conjunto de socios, como una actividad contemplada en una asamblea general del club de tenis. 		
<p>Estado actual:</p> <ul style="list-style-type: none"> - Se ha implementado un prototipo semi-funcional que demuestra algunas de las características que tendrá el futuro sistema. Se ha puesto en Internet a disposición de los miembros de la organización en general. 		

Tabla 1.1. Riesgo: Impacto negativo del software en el negocio

Tipo de riesgo: Del negocio	Probabilidad: Moderada	Impacto: Catastrófico
Riesgo: Requerimientos captados no corresponden a los que el cliente pide.		
Descripción:		
<p>Algunos requerimientos pueden ser mal entendidos, debido a que en alguna reunión o entrevista, un integrante de la organización no supo explicar correctamente sus ideas o el entrevistador no ha usado una metodología correcta para obtener la información. Esto puede ser muy grave debido a que esta información es la base de los requerimientos funcionales del futuro sistema.</p>		
Reducción del riesgo:		
<ul style="list-style-type: none"> - Preparar cuestionarios con preguntas precisas y con un lenguaje lo más didáctico posible y que sea familiar y natural para el entrevistado. - Repetir al entrevistado las ideas que el ha expresado, ya que así se obtiene una verificación de lo que quiso decir y lo que el entrevistador en realidad ha entendido (retroalimentación). - Realizar las entrevistas y otras actividades en lugares que tengan la menor cantidad de distracciones posibles, evitando desviarse del propósito principal de la actividad, que es la obtención de los requerimientos del futuro sistema. 		
Plan de contingencia:		
<ul style="list-style-type: none"> - Utilizar un proceso de desarrollo de software que permita manejar de buena forma los cambios en los requerimientos, básicamente en lo que se refiere a flexibilidad, modularidad y escalabilidad. 		
Estado actual:		
<ul style="list-style-type: none"> - En el presente informe se da cuenta de la elección del paradigma, metodología y herramientas de programación que cumplen con las características requeridas para atenuar o eliminar el efecto de este riesgo. 		

Tabla 1.2. Riesgo: Requerimientos captados no corresponden a los que el cliente pide

Tipo de riesgo: Del negocio	Probabilidad: Baja	Impacto: Crítico
Riesgo: Cambios de última hora en los requerimientos		
Descripción: Alguna de las personas de la organización pueden tener interés en que el software haga alguna tarea extraordinaria o que no haya sido registrada en la toma de requerimientos preliminar. Ejemplo: Pagos online a través del mismo sistema.		
Reducción del riesgo: - Presentar al cliente esquemas y/o diagramas de flujo que muestren como será el funcionamiento del futuro sistema. Pedir al cliente su opinión con respecto a lo presentado y eventualmente tomar nota de los cambios que pudiera pedir.		
Plan de contingencia: - Utilizar un proceso de desarrollo de software que permita manejar de buena forma los cambios en los requerimientos, básicamente en lo que se refiere a flexibilidad, modularidad y escalabilidad.		
Estado actual: - En el presente informe se da cuenta de la elección del paradigma, metodología y herramientas de programación que cumplen con las características requeridas para atenuar o eliminar el efecto de este riesgo. - Se han hecho algunos cambios en algunos requerimientos a pedido del administrador del club de tenis.		

Tabla 1.3. Riesgo: Cambios de última hora en los requerimientos

Tipo de riesgo: Del negocio	Probabilidad: Moderada	Impacto: Catastrófico
Riesgo: Bajo interés en usar el sistema		
Descripción:		
<p>Los socios del club de tenis, que serán los usuarios reales del futuro sistema, no aprecian los beneficios que este les entrega.</p>		
Reducción del riesgo:		
<ul style="list-style-type: none"> - Trabajar en una estrategia comunicacional y de difusión junto al administrador del club de tenis y el directorio para informar a los socios acerca de los beneficios que tendrá la implementación del nuevo sistema. Esto se puede hacer a través de algunos afiches que se pueden colocar en dependencias del club de tenis, a través de correos electrónicos y con notas informativas colocadas en el actual sitio Web del club y en reuniones de socios. 		
Plan de contingencia:		
<ul style="list-style-type: none"> - Demostración práctica del uso del sistema en reunión de socios. - El directorio del club de tenis, fiel a la confianza que ha puesto en el proyecto, debe fijar políticas internas que conduzcan a los socios a usar de forma recurrente el nuevo sistema. - Diseñar buenas interfaces de usuario desde el punto de vista de la usabilidad, ya que esto mejora la calidad del portal Web y puede generar el entusiasmo necesario para que los usuarios utilicen el sistema. 		
Estado actual:		
<ul style="list-style-type: none"> - Se ha implementado un prototipo semi-funcional que demuestra algunas de las características que tendrá el futuro sistema. Se ha puesto en Internet a disposición de los miembros de la organización en general. 		

Tabla 1.4. Riesgo: Bajo interés en usar el sistema

Tipo de riesgo: Del negocio	Probabilidad: Moderada	Impacto: Catastrófico
Riesgo: Poco compromiso de la organización		
Descripción:		
<p>Los directivos y la administración del club de tenis no tienen certeza de los beneficios que el sistema les entregará a los socios y usuarios; por lo tanto su nivel de interés en la realización del proyecto es bajo, La entrega de información y entrevistas no es fluida o derechamente es negada.</p>		
Reducción del riesgo:		
<ul style="list-style-type: none"> - Realizar una presentación, en lo posible al directorio y administrador, que muestre en forma realista las funcionalidades y beneficios que el sistema entregará a la organización. NO crear falsas expectativas acerca de las bondades del sistema. Ejemplo: No se podrá hacer los pagos de las mensualidades online a través del portal Web. - Entregar informe preliminar a los directivos y administración del club de tenis, el cual contenga la visión general del problema de gestión de la información que la organización posee y junto con ello entregar una propuesta de solución al problema, describiendo las etapas que se deben cumplir y los plazos medios para su cumplimiento 		
Plan de contingencia:		
<ul style="list-style-type: none"> - Implementar un prototipo funcional que demuestre las funcionalidades que tendrá el futuro sistema. 		
Estado actual:		
<ul style="list-style-type: none"> - Se ha entregado el informe preliminar que contiene la solución propuesta - Se ha implementado el prototipo que demuestra parte de la funcionalidad del futuro sistema. 		

Tabla 1.5. Riesgo: Poco compromiso de la organización

Tipo de riesgo: Técnico	Probabilidad: Moderada	Impacto: Crítico
Riesgo: Error en el dimensionamiento del producto		
Descripción: El sistema puede ser más extenso o complejo de lo que se ha estimado, por lo tanto la planificación de las distintas tareas y etapas del proyecto pueden verse afectadas, lo que podría generar retrasos en la fecha de entrega final del producto y eventualmente generar costos extras no previstos.		
Reducción del riesgo: <ul style="list-style-type: none"> - Una vez obtenidos los requerimientos de los usuarios y del sistema, hay que determinar los alcances prácticos, en base a casos de uso, que el futuro software tendrá. Según esto se aplicarán métricas para cuantificar el trabajo y tiempo necesario para desarrollar el proyecto. Las métricas de software pueden ser a través de puntos de función o COCOMO. - Verificar constantemente los plazos e hitos entregables del proyecto y contrastarlos con las fechas previstas en la carta gantt. 		
Plan de contingencia: <ul style="list-style-type: none"> - Hacer uso de tiempo extra previsto en la planificación temporal del proyecto. 		
Estado actual: <ul style="list-style-type: none"> - Hasta el momento las actividades planificadas se han realizado en los plazos previstos. 		

Tabla 1.6. Riesgo: Error en el dimensionamiento del producto

Tipo de riesgo: Técnico	Probabilidad: Moderada	Impacto: Crítico
Riesgo: Inadecuado entorno de desarrollo		
Descripción:		
<p>Las herramientas de desarrollo en su conjunto no entregan toda la flexibilidad y el soporte técnico para que los requerimientos funcionales y no funcionales que el sistema demanda sean cumplidos.</p>		
Reducción del riesgo:		
<ul style="list-style-type: none"> - Estudiar detenidamente las herramientas disponibles para desarrollar el proyecto, pidiendo opinión a personas expertas y considerando la disponibilidad de manuales, tutoriales, ayudas de ciber-comunidades. - En lo posible revisar otros sistemas que estén en producción y que hayan sido desarrollados con las herramientas a evaluar. - Consultar la opinión a personas con mayores conocimientos acerca de la viabilidad en el uso de las herramientas de desarrollo elegidas para el proyecto. - Tener segundas opciones que se ajusten al paradigma y metodología elegidos. Estas opciones deben estar disponibles y viables desde todos los aspectos considerados en el estudio de factibilidad. 		
Plan de contingencia:		
<ul style="list-style-type: none"> - Hacer uso de las segundas opciones previstas para esta contingencia 		
Estado actual:		
<ul style="list-style-type: none"> - Ya se tienen consideradas las segundas opciones para: Lenguaje de programación, Framework y Base de datos. 		

Tabla 1.7. Riesgo: Inadecuado entorno de desarrollo

Tipo de riesgo: Técnico	Probabilidad: Alta	Impacto: Crítico
Riesgo: Poca experiencia del equipo de desarrollo		
Descripción:		
<p>El equipo de desarrollo no domina a cabalidad las técnicas y/o herramientas de desarrollo utilizadas para cumplir en buena forma con las exigencias del proyecto. Pueden haber incongruencias o errores conceptuales con respecto a metodologías usadas en la gestión de proyectos de software. Esto se puede deber a la poca experiencia en proyectos de similares características.</p>		
Reducción del riesgo:		
<ul style="list-style-type: none"> - El estudio de metodologías a implementar, debe ser una constante. - Consultar a personas con experiencia en los temas que no se dominan en un 100% - Revisar proyectos anteriores, con el fin de reforzar los conocimientos. 		
Plan de contingencia:		
<ul style="list-style-type: none"> - Pedir asesoría a personas expertas en el tema que no se domina. - Destinar tiempo extra al aprendizaje de las técnicas o implementación de metodologías poco dominadas por el equipo de desarrollo de software 		
Estado actual:		
<ul style="list-style-type: none"> - Se ha repasado y leído material académico utilizado en las asignaturas de: SIA, Ingeniería de software, Bases de datos, HCI. - Se han bajado desde Internet manuales e información acerca del paradigma a utilizar y las herramientas de desarrollo. 		

Tabla 1.8. Riesgo: Poca experiencia del equipo de desarrollo

Tipo de riesgo: Técnico	Probabilidad: Moderada	Impacto: Crítico
Riesgo: Problemas de usabilidad del sistema		
Descripción: La interfaz de presentación del sistema no es clara en la entrega de la información y la presentación de la ayuda para la correcta navegación de los usuarios. Los colores, las tipologías de fuentes, las animaciones y el diseño en general de la aplicación no generan un entorno armónico y amigable.		
Reducción del riesgo: - Planificar un conjunto de evaluaciones de usabilidad (evaluaciones heurísticas, pruebas de usuario), con el objeto de medir la facilidad de uso del sistema.		
Plan de contingencia: - Repetir pruebas de usabilidad - Usar otras técnicas para medir usabilidad (ej: Thinking aloud)		
Estado actual: - El equipo de desarrollo del proyecto ha aprobado el curso <i>Human computer interaction</i> dictado en la escuela de ingeniería informática de la PUCV, el cual entregó conocimientos teóricos y prácticos en torno a la implementación de variadas técnicas y pruebas para medir la usabilidad.		

Tabla 1.9. Riesgo: Problemas de usabilidad del sistema

Tipo de riesgo: Técnico	Probabilidad: Baja	Impacto: Crítico
Riesgo: Mayor número de usuarios de lo previsto usarán el software		
Descripción: Una explosiva demanda en el uso del sistema puede provocar saturación de los servidores (Web, correo y de Bases de datos), lo que provocaría fallos en las transacciones y en la integridad de la información.		
Reducción del riesgo: <ul style="list-style-type: none"> - Consultar y verificar con encargados del Web hosting la capacidad de accesos concurrentes a los servidores. - Tener alternativas de Web hosting en caso de que la primera opción no entregue garantías de funcionalidad. 		
Plan de contingencia: <ul style="list-style-type: none"> - Pedir ampliación del ancho de banda en la conexión a los servidores y aumentar el número de conexiones concurrentes. - Cambiar de proveedor de Web hosting. 		
Estado actual: <ul style="list-style-type: none"> - El club de tenis tiene un contrato de Web hosting, el cual cumple con las características técnicas deseadas; sin embargo se debe buscar alguna alternativa en caso de que se presentara alguna situación imprevista. 		

Tabla 1.10. Riesgo: Mayor número de usuarios de lo previsto usarán el software

Tipo de riesgo: Técnico	Probabilidad: Baja	Impacto: Crítico
Riesgo: Problemas de hosting		
Descripción:		
<p>El proveedor de Web hosting no cuenta con las herramientas, la capacidad técnica y/o personal calificado para entregar un servicio eficiente que de plena seguridad y tranquilidad con respecto al ancho de banda, sistemas de respaldo, soporte de servidores, up-time de servidores, estadísticas de tráfico, atención al cliente, accesos concurrentes, etc.</p>		
Reducción del riesgo:		
<ul style="list-style-type: none"> - Contactarse a priori con el encargado técnico del Web hosting para entregarle las características reales y los requerimientos de hardware y software que el futuro sistema tendrá y a la vez confirmar que el hardware y software de los servidores soporte correctamente el sistema a implantar. - Realizar pruebas para medir funcionamiento y eficiencia real del servicio de Web hosting. Estas pruebas deben evaluar los servidores de base de datos, servidor Web y servidores de correo. - Tener alternativas de Web hosting en caso de que la primera opción no entregue garantías de funcionalidad o su rendimiento no sea el esperado. 		
Plan de contingencia:		
<ul style="list-style-type: none"> - Cambiar de proveedor de Web hosting. 		
Estado actual:		
<ul style="list-style-type: none"> - El club de tenis tiene un contrato de Web hosting, el cual cumple con las características técnicas deseadas; sin embargo se debe buscar alguna alternativa en caso de que se presentara alguna situación imprevista. 		

Tabla 1.11. Riesgo: Problemas de hosting

Tipo de riesgo: Técnico	Probabilidad: Alta	Impacto: Catastrófico
Riesgo: Problemas de seguridad (cracking de sistemas)		
<p>Descripción:</p> <p>La proliferación de ataques y técnicas para ejecutar aplicaciones con el objetivo de robar contraseñas y romper un sistema a través de malware y spyware es conocido como cracking. Todas estas técnicas están ampliamente distribuidas por el ciberespacio, lo cual representa un serio problema para la seguridad de sistemas informáticos, especialmente aquellos que utilizan Internet como componente de su arquitectura.</p>		
<p>Reducción del riesgo:</p> <ul style="list-style-type: none"> - Implementar un sistema seguro de validación de sesiones de usuario, que contemple la encriptación de claves de acceso. - Programa de respaldo de base de datos de acuerdo a las necesidades. - Implementar un sistema de auditorias en la bases de datos que permita enviar informes de acciones realizadas sobre los datos. - Interiorizarse de los sistemas de seguridad que el Web hosting aplica en relación a: Firewall, monitoreo de puertos, control antispam, control de robots, etc. - Estudio de las vulnerabilidades de seguridad de la base de datos y el lenguaje de programación que han sido reportadas y documentadas en sitios de seguridad informática. - Estudio de los parches de seguridad y métodos de protección que se pueden implementar con el fin de prevenir ataques informáticos y vulnerabilidades descritas en el punto anterior. 		
<p>Plan de contingencia:</p> <ul style="list-style-type: none"> - Automatizar el informe de errores y caídas en la base de datos o la aplicación a través del envío automático de correos electrónicos al Web master y encargados del hosting. 		
<p>Estado actual:</p> <ul style="list-style-type: none"> - Estudio de los distintos ataques informáticos existentes - El equipo de desarrollo asiste a un curso de seguridad informática en la escuela de ingeniería informática de la PUCV. 		

Tabla 1.12. Riesgo: Problemas de seguridad (cracking de sistemas)

Tipo de riesgo: Económico	Probabilidad:	Impacto:
Riesgo: Sobre valoración o sub-valoración del producto		
Descripción: <p>Los problemas de sub-valoración económica del proyecto pueden generar situaciones catastróficas que llevan a proyectos a un mal resultado. Para el caso particular de este proyecto, estos riesgos no aplican, ya que su propósito fundamental es de tipo académico; por lo tanto los aspectos relacionados al lucro tienen una significancia relativa y casi marginal; sin embargo es importante mencionar estos riesgos ya que muchas veces condicionan el éxito de proyectos de software.</p>		
Reducción del riesgo: <p>- No aplica</p>		
Plan de contingencia: <p>- No aplica</p>		
Estado actual: <p>- No aplica</p>		

Tabla 1.13. Riesgo: Sobre valoración o sub-valoración del producto.

5.3 PARADIGMA DE DESARROLLO

5.4 MODELO INCREMENTAL O PROCESO UNIFICADO DE SOFTWARE

El modelo incremental combina elementos del modelo lineal secuencial (aplicados repetidamente) con la filosofía interactiva de construcción de prototipos. El modelo incremental aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un “incremento” del software. Se debería tener en cuenta que el flujo del proceso de cualquier incremento puede incorporar el paradigma de construcción de prototipos.

Cuando se utiliza un modelo incremental, el primer incremento a menudo es un producto esencial. Es decir, se afrontan requisitos básicos, pero muchas funciones suplementarias (algunas conocidas, otras no) quedan sin extraer. El cliente utiliza el producto central (o sufre la revisión detallada). Como un resultado de utilización y/o de evaluación, se desarrolla un plan para el incremento siguiente. El plan afronta la modificación del producto central a fin de cumplir mejor las necesidades del cliente y la entrega de funciones, y características adicionales. Este proceso se repite siguiendo la entrega de cada incremento, hasta que se elabore el producto completo [1].

Un proceso de desarrollo de software basado en el modelo incremental, es el proceso unificado de software. El proceso unificado es más que un simple proceso, es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos [2].

El proceso unificado está dirigido por casos de uso esta Dirigido por Casos de Uso, los cuales son un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante; por lo tanto representan requisitos funcionales.

5.4.1 ELECCION DE PARADIGMA

Para desarrollar el presente proyecto se ha elegido el paradigma denominado *incremental o proceso unificado de software*, debido a los siguientes motivos:

- El Proceso Unificado es un marco de desarrollo iterativo e incremental compuesto de cuatro fases denominadas Inicio, Elaboración, Construcción y Transición. Todas estas fases han sido ampliamente estudiadas y aplicadas por parte del equipo de trabajo en proyectos de software anteriores.

- Está ampliamente difundido y documentado en bibliotecas universitarias e Internet.
- Está enfocado en los riesgos del proyecto, por lo cual el análisis RSGR del presente proyecto que ha sido desarrollado en forma muy minuciosa es una gran fuente de información para construir la solución de modelado.
- El Proceso Unificado (UP) está soportado ampliamente por UML (Lenguaje de Modelado Unificado), lo que facilita su implementación ya que existen muchas herramientas para modelar software utilizando UML, muchas de ellas con licencias *open source*.
- Tanto UP como UML están diseñados para modelar software usando la metodología de análisis y paradigma de programación orientada a objetos.

5.5 PARADIGMA DE PROGRAMACIÓN

Un paradigma de programación representa un enfoque particular o filosofía para la construcción del software. [13]. Para este proyecto se ha optado por el paradigma orientado a objetos (POO), la cual define los programas en términos de "clases de objetos", objetos que son entidades que combinan *estado* (es decir, datos), *comportamiento* (esto es, procedimientos o *métodos*) e identidad (propiedad del objeto que lo diferencia del resto). La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar.

La programación orientada a objetos es una nueva forma de programar que trata de encontrar una solución a estos problemas. Introduce nuevos conceptos, que superan y amplían conceptos antiguos ya conocidos. Entre ellos destacan los siguientes:

- **Objeto:** entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos). Corresponden a los objetos reales del mundo que nos rodea, o a objetos internos del sistema (del programa).
- **Clase:** definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas.
- **Método:** algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.
- **Evento:** un suceso en el sistema (tal como una interacción del usuario con la máquina, o un mensaje enviado por un objeto). El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente. También se puede definir como evento, a la reacción que puede desencadenar un objeto, es decir la acción que genera.
- **Mensaje:** una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.
- **Propiedad o atributo:** contenedor de un tipo de datos asociados a un objeto (o a una clase de objetos), que hace los datos visibles desde fuera del objeto, y cuyo valor puede ser alterado por la ejecución de algún método.
- **Estado interno:** es una propiedad invisible de los objetos, que puede ser únicamente accedida y alterada por un método del objeto, y que se utiliza para indicar distintas situaciones posibles para el objeto (o clase de objetos).
- **Componentes de un objeto:** atributos, identidad, relaciones y métodos.

El lenguaje UML y el Framework CakePHP están hechos para modelar y programar aplicaciones orientadas a objetos respectivamente; por lo tanto hay una correspondencia coherente entre el modelo de desarrollo (UP), el lenguaje de modelado (UML) y el paradigma de programación (POO).

5.6 HERRAMIENTAS

5.6.1 HERRAMIENTA DE MODELADO UML: POSEIDON



Poseidon para UML[12] puede simplificar la compleja tarea de desarrollo de software ayudando a estructurar pensamientos, a clarificar la comunicación, y a encontrar la correcta abstracción.

Poseidon para UML evita estas distracciones. La intuitiva interfaz hace de Poseidon la herramienta más rápida de UML para dominar el análisis orientado a objetos, liberando al diseñador para centrarse solamente en su modelo. La abundancia de características disponibles realza la productividad desde el comienzo sin ser un obstáculo en el camino.

Existen diversas ediciones del software, de las cuales las más importantes son:

- **Community Edition:** Perfecta para estudiantes, principiantes, y otros usuarios no comerciales, esta edición libre es la introducción ideal a UML, complementada además, con la ingeniería de Java.
- **Standard Edition:** Dirigida a los analistas, esta edición hace el trabajo de diseño y documentación de modelos rápido y fácil, junto con la ingeniería inversa de Java, UMLdoc, y plugins.
- **Professional Edition:** Los desarrolladores encontrarán un software lleno de características de gran alcance, tales como, la integración de Eclipse, tecnología Java, y generación del código para muchas otras lenguajes.
- **Enterprise Edition:** Equipos de desarrolladores pueden utilizar funciones y herramientas colaborando en tiempo real junto con las tecnología Java, para desarrollar software a través de una habitación o a través del mundo.

La community edition es open source, por lo cual será la utilizada en el presente proyecto.

5.6.2 SISTEMA ADMINISTRADOR DE BASES DE DATOS: POSTGRESQL



PostgreSql es un poderoso sistema administrador de base de datos de código abierto. Tiene más de 15 años de desarrollo activo y de una arquitectura probada que esta apoyada por una fuerte confiabilidad, integridad de datos, y su exactitud. Este sistema funciona en la mayoría de los sistemas operativos importantes, incluyendo Linux, UNIX, Windows. Soporta el uso de llaves foráneas, joins, triggers, y procedimientos almacenados (en múltiples lenguajes). Incluye la mayoría de los tipos de datos de SQL92 y SQL99, incluyendo datos del tipo INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, y TIMESTAMP. También soporta el almacenaje de objetos binarios grandes, incluyendo imágenes, sonidos o video. Tiene interfaces de programación nativos para /C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, entre otros [6].

PostgreSql es un sistema de gran alcance que permite manejar gran cantidad de datos, es escalable, flexible, de gran rendimiento, provee de gran seguridad, incluyendo el control de acceso a los clientes y las comunicaciones cifradas protegen datos valiosos y asegura el acceso permanente a los empleados y a los clientes. PostgreSQL es un motor de base de datos, es servidor de base de datos relacional libre, liberado bajo la licencia BSD.[6]

5.6.3 FRAMEWORK PHP: CAKEPHP



En el desarrollo de software, un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros softwares para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

CakePHP es un framework de desarrollo de aplicaciones Web escrito en PHP, el cual facilita al usuario la interacción con la base de datos mediante el uso de ActiveRecord (patrón de diseño enfocado al acceso a bases de datos, el cual maneja las tablas como objetos). Además hace uso del patrón Modelo – Vista - Controlador (patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos). Otras características de CakePHP son: Compatible con PHP4 y PHP5, URLs amigables, sistema de plantillas rápido y flexible, helpers para AJAX, javascript, HTML, forms y más, trabaja en cualquier subdirectorio del sitio, validación integrada, aAccess Control Lists, sanitización de datos, componentes de seguridad y sesión. [8].

CakePHP tiene Licencia MIT (*Massachusetts Institute of Technology*), la cual permite su libre uso y distribución, pudiendo ser instalado en múltiples plataformas.

5.6.4 SERVIDOR WEB: APACHE



El servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementan el protocolo HTTP. Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración. [10]

La mayoría de las vulnerabilidades de seguridad descubiertas y resueltas puede, en la mayoría de los casos, ser abusada solamente por los usuarios locales y no puede ser accionada remotamente.

Algunas de las ventajas de Apache son: Modularidad, open source, multi-plataforma, extensible, popular (fácil conseguir ayuda/soporte), gratuito.

5.7 REQUERIMIENTOS

5.7.1 ACTORES DEL SISTEMA

Este apartado contiene los diferentes actores que se han identificado, especificados mediante la plantilla para actores de casos de uso.

ACT-01	Administrador
Actores asociados	Este actor representa a la persona que administra el club de tenis
Comentarios	Los usuarios con perfil de administrador serán 2

Tabla 2.1. Actores del sistema: Administrador

ACT-02	Socio
Actores asociados	Este actor representa a los socios del club de tenis
Comentarios	Los usuarios con perfil socio son 100, con un aumento del 5% anual

Tabla 2.2. Actores del sistema: Socio

ACT-03	Colaborador
Actores asociados	Este actor representa a las personas encargadas de aportar noticias e información al sitio Web.
Comentarios	Se estima que el máximo de usuarios con este perfil será de 10

Tabla 2.3. Actores del sistema: Colaborador

5.7.2 REQUERIMIENTOS DEL SISTEMA

En este apartado se define una lista con los diferentes objetivos que se esperan alcanzar cuando el sistema software a desarrollar esté en explotación.

OBJ-01	Gestionar socios
Actores asociados	El sistema deberá gestionar los socios del club: ingresos, bajas, modificaciones de datos, listar datos socios.
Estabilidad	alta

Tabla 3.1. Requerimientos del sistema: Gestionar socios

OBJ-02	Gestionar colaboradores
Actores asociados	El sistema deberá gestionar los colaboradores del club: ingresos, modificaciones, eliminaciones, asignación de privilegios.
Estabilidad	alta

Tabla 3.2. Requerimientos del sistema: Gestionar colaboradores

OBJ-03	Gestionar reservas de canchas
Actores asociados	El sistema deberá gestionar los alquileres de canchas: Ingresar, modificar, eliminar, listar reservas de canchas.
Estabilidad	alta

Tabla 3.3. Requerimientos del sistema: Gestionar reservas de canchas

OBJ-04	Gestionar noticias
Actores asociados	El sistema deberá gestionar las noticias que se verán en el sitio Web: Ingresar, modificar, eliminar, listar las noticias.
Estabilidad	

Tabla 3.4. Requerimientos del sistema: Gestionar noticias

OBJ-05	Gestionar pagos
Actores asociados	El sistema deberá gestionar los pagos de los socios del club: ingresos, modificaciones, eliminaciones, listados de pagos de socios.
Estabilidad	alta

Tabla 3.5. Requerimientos del sistema: Gestionar pagos

OBJ-06	Gestionar mensajería
Actores asociados	El sistema deberá gestionar los mensajes que se enviarán a los socios: Configurar mensajes, modificar mensajes, eliminar mensajes, activar mensajes.
Estabilidad	Alta

Tabla 3.6. Requerimientos del sistema: Gestionar mensajería

OBJ-07	Gestionar membresías
Actores asociados	El sistema deberá gestionar los tipos de membresía: Ingresar membresía, modificar, eliminar, listar membresía.
Estabilidad	Alta

Tabla 3.7. Requerimientos del sistema: Gestionar membresías

OBJ-08	Gestionar canchas
Actores asociados	El sistema deberá gestionar las canchas y sus características: Ingresar cancha, Modificar características, eliminar cancha, listar canchas.
Estabilidad	Alta

Tabla 3.8. Requerimientos del sistema: Gestionar canchas

5.7.3 REQUERIMIENTOS DE ALMACENAMIENTO DE INFORMACIÓN

Esta sección contiene la lista de requisitos de almacenamiento de información que se han identificado en este análisis.

	Información sobre socios
Objetivos asociados	OBJ-01 Gestionar los socios
Requisitos asociados	RF- 01 Ingresar nuevo socio RF- 02 Modificar datos de socios RF- 03 Listar datos de socios RF- 04 Eliminar socio RF- 00 Login de usuario socio.
Actores asociados	ACT-01 – ACT-02
Datos específicos	Número de socio, que deberá ser único para cada socio Número del documento nacional de identidad Nombres y apellidos Fecha de nacimiento Sexo Fotografía Fecha de ingreso como socio Dirección Teléfonos E-mail Reservas efectuadas en un momento dado Tipo de membresía Login de socio
Comentarios	Ninguno

Tabla 4.1. Requerimientos de almacenamiento de información: Información de socios

	Información sobre colaboradores
Objetivos asociados	OBJ-02 Gestionar los colaboradores
Requisitos asociados	RF-05 Ingresar nuevo colaborador RF-06 Modificar datos colaborador RF-07 Eliminar colaborador RF-30 Asignar privilegios RF-00 Login de usuario colaborador
Actores asociados	ACT-01
Datos específicos	Número de colaborador, que deberá ser único para cada colaborador Rut del colaborador Nombres y apellidos Fecha de nacimiento Sexo Fotografía Fecha de ingreso como colaborador Dirección Teléfonos E-mail Login de colaborador
Comentarios	Ninguno

Tabla 4.2. Requerimientos de almacenamiento de información: Información sobre colaboradores

	Información sobre reservas de canchas
Objetivos asociados	OBJ-03 Gestionar las reservas de canchas de tenis
Requisitos asociados	RF-08 Ingreso de reserva de cancha RF-09 Modificación de reserva de cancha RF-10 Eliminar reserva de cancha RF-11 Listar reserva de canchas RF-00 Login
Actores asociados	ACT-01 – ACT-02
Datos específicos	Número de cancha que debe ser única Fecha de reserva cancha Rut socio que reserva Hora reserva cancha Reservas efectuadas en cada momento Disponibilidad de canchas
Comentarios	Ninguno

Tabla 4.3. Requerimientos de almacenamiento de información: Información sobre reservas de canchas

	Información sobre noticias
Objetivos asociados	OBJ-04 Gestionar las noticias de canchas de tenis
Requisitos asociados	RF-12 Ingresar noticia RF-13 Modificar noticia RF-14 Listar noticia RF-15 Eliminar noticia RF-00 Login
Actores asociados	ACT-01 – ACT-03
Datos específicos	Número noticia, que será única para cada noticia Fecha y Hora de publicación noticia Titulo de la noticia Rut del autor de la noticia Contenido de la noticia

Tabla 4.4. Requerimientos de almacenamiento de información: Información sobre noticias

	Información sobre los pagos de mensualidades
Objetivos asociados	OBJ-05 Gestionar los pagos
Requisitos asociados	RF-16 Ingresar pago RF-17 Modificar pago RF-18 Eliminar pago RF-19 Listar pago RF-00 Login
Actores asociados	ACT-01 – ACT-02
Datos específicos	Número pago, que será única para cada noticia Rut socio que paga Fecha pago Mes pagado Número comprobante Monto pago Forma de pago (Efectivo, Cheque, Traspaso electrónico)
Comentarios	Ninguno

Tabla 4.5. Requerimientos de almacenamiento de información: Información sobre los pagos de mensualidades

	Información sobre mensajería
Objetivos asociados	OBJ-06 Gestionar los mensajes del sistema
Requisitos asociados	RF-20 Configurar mensaje RF-21 Modificar mensaje RF-22 Eliminar mensaje RF-23 Activar mensaje RF-00 Login
Actores asociados	ACT-01
Datos específicos	Número de mensaje, que debe ser único Fecha de creación Hora de creación Contenido mensaje

Tabla 4.6. Requerimientos de almacenamiento de información: Información sobre mensajería

	Información sobre membresía
Objetivos asociados	OBJ-07 Gestionar membresías
Requisitos asociados	RF-24 Ingresar membresía RF-25 Modificar membresía RF-26 Listar membresía RF-27 Eliminar membresía
Actores asociados	ACT-01
Datos específicos	Número de membresía, que debe ser único Fecha de creación Nombre de la membresía Valor membresía Descripción de la membresía
Comentarios	Ninguno

Tabla 4.7. Requerimientos de almacenamiento de información: Información sobre membresía

	Información sobre canchas
Objetivos asociados	OBJ-08 Gestionar las canchas
Requisitos asociados	RF-28 Ingresar nueva cancha RF-29 Modificar datos de una cancha RF-30 Listar canchas RF-31 Eliminar cancha RF-00 Login de usuario colaborador
Actores asociados	ACT-01
Datos específicos	Número de cancha, que deberá ser único para cada colaborador Tipo de cancha Fotografía Posee Iluminación artificial Descripción de cancha Disponibilidad
Comentarios	Ninguno

Tabla 4.8. Requerimientos de almacenamiento de información: Información sobre canchas

5.8 ANÁLISIS DE REQUERIMIENTOS

5.8.1 DIAGRAMAS DE CASOS DE USO

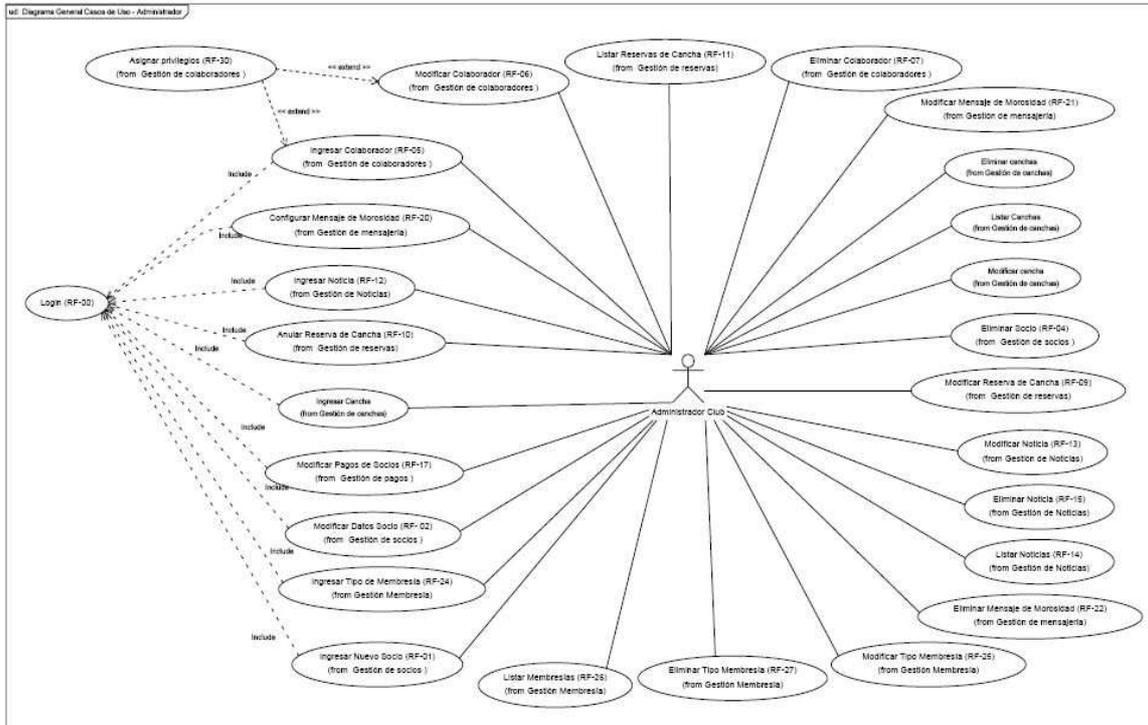


Diagrama 1.1. Diagrama general de casos de uso administrador

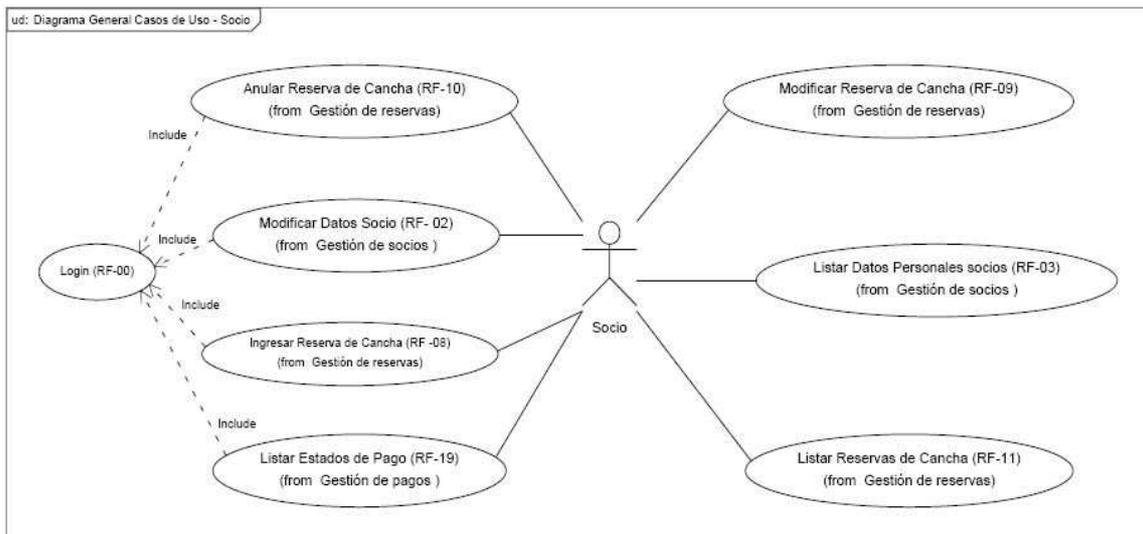


Diagrama 1.2. Diagrama general de casos de uso socio

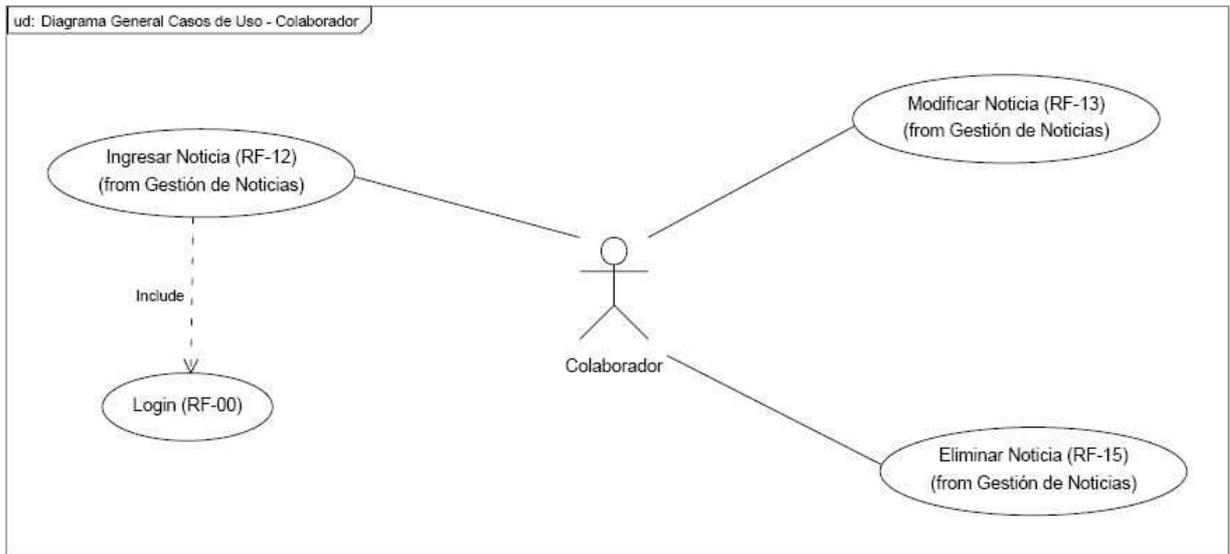


Diagrama 1.3. Diagrama general de casos de uso colaborador

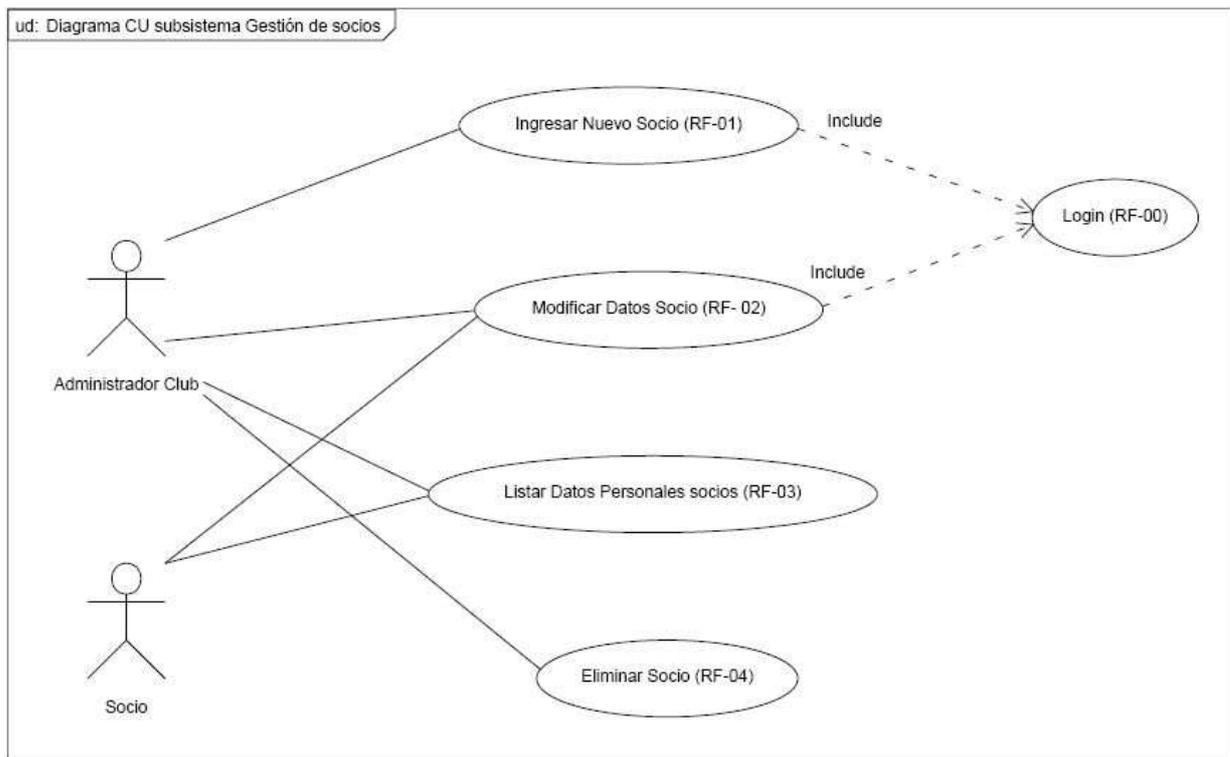


Diagrama 2.1. Casos de uso. Subsistema de gestión de socios

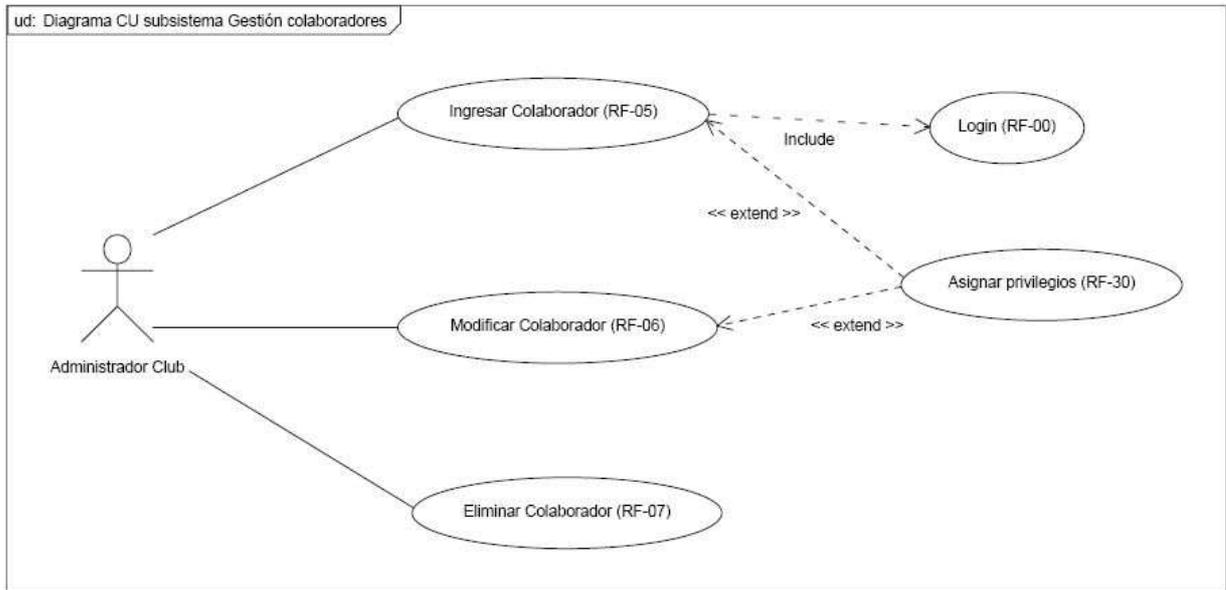


Diagrama 2.2. Casos de uso. Subsistema de gestión de colaboradores

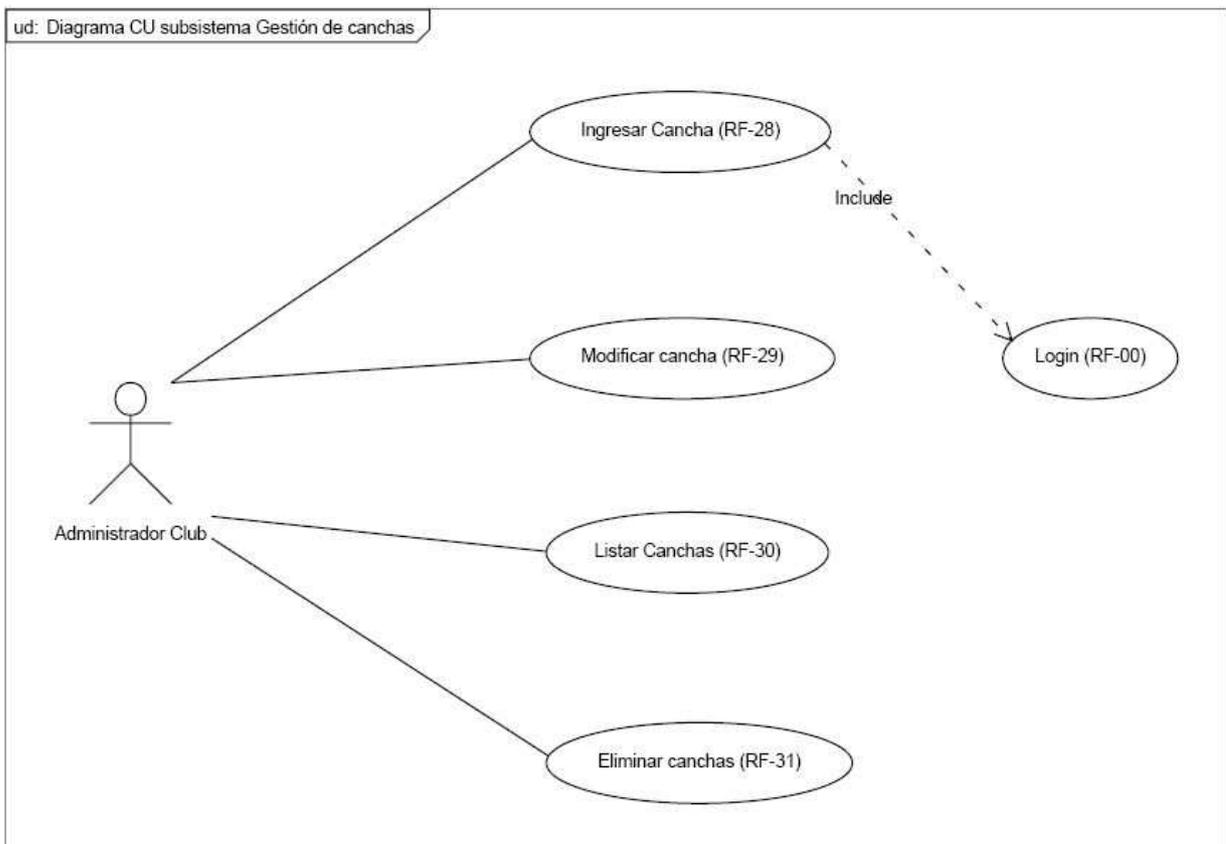


Diagrama 2.3. Casos de uso. Subsistema de gestión de canchas

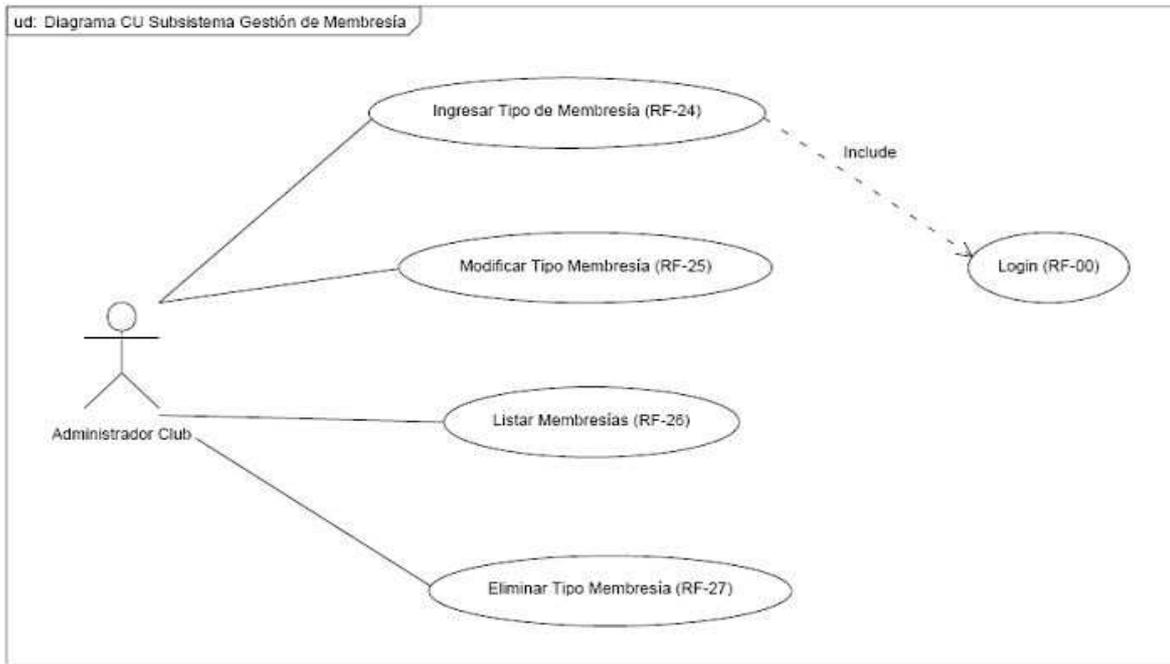


Diagrama 2.4. Casos de uso. Subsistema de gestión de membresías

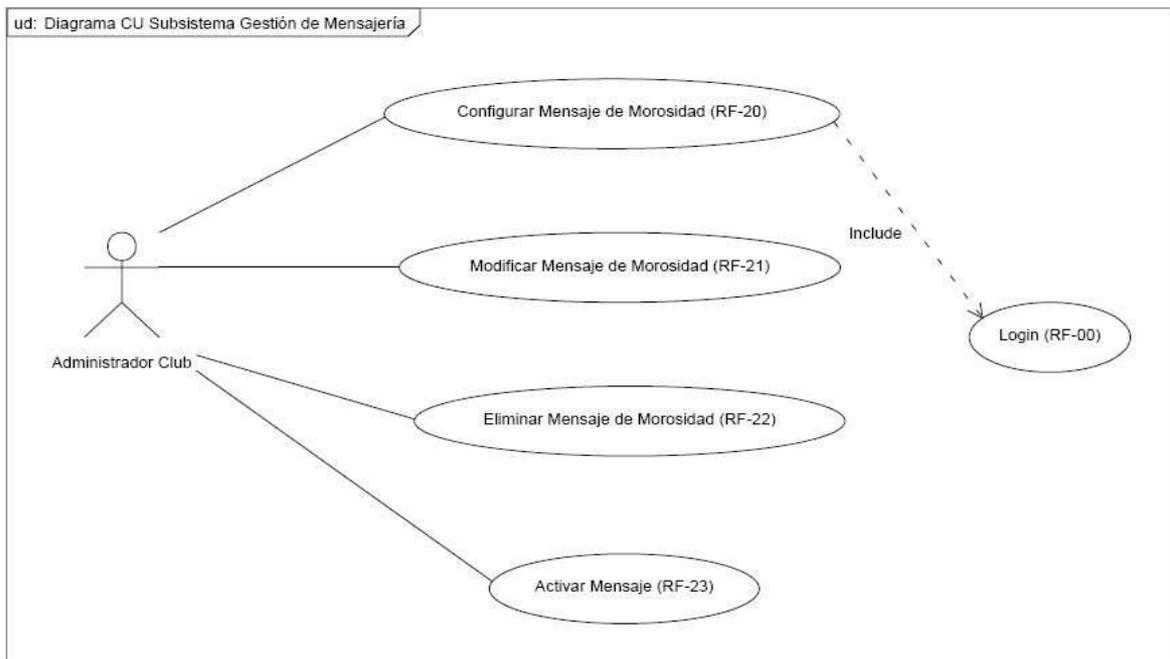


Diagrama 2.5. Casos de uso. Subsistema de gestión de mensajería

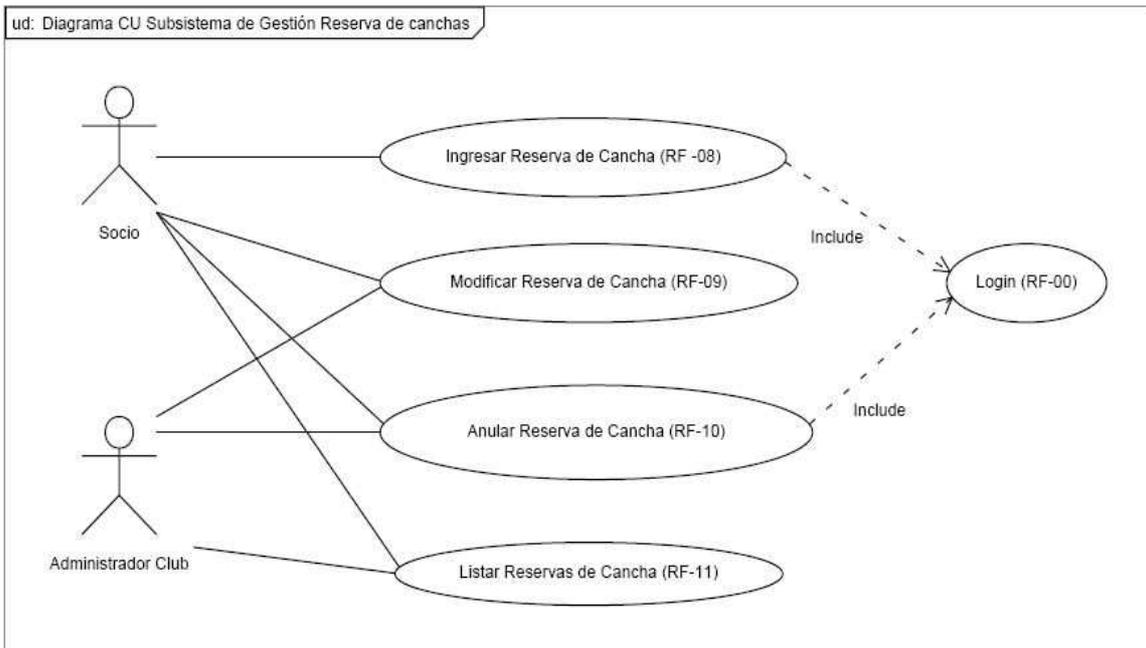


Diagrama 2.6. Casos de uso. Subsistema de gestión de reserva de canchas

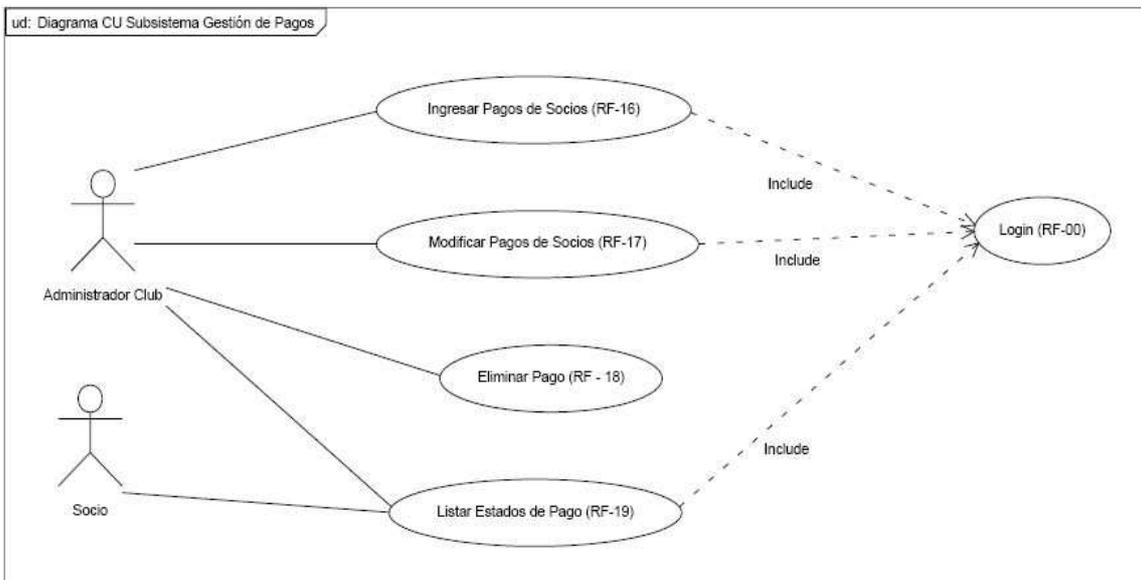


Diagrama 2.7. Casos de uso. Subsistema de gestión de pagos

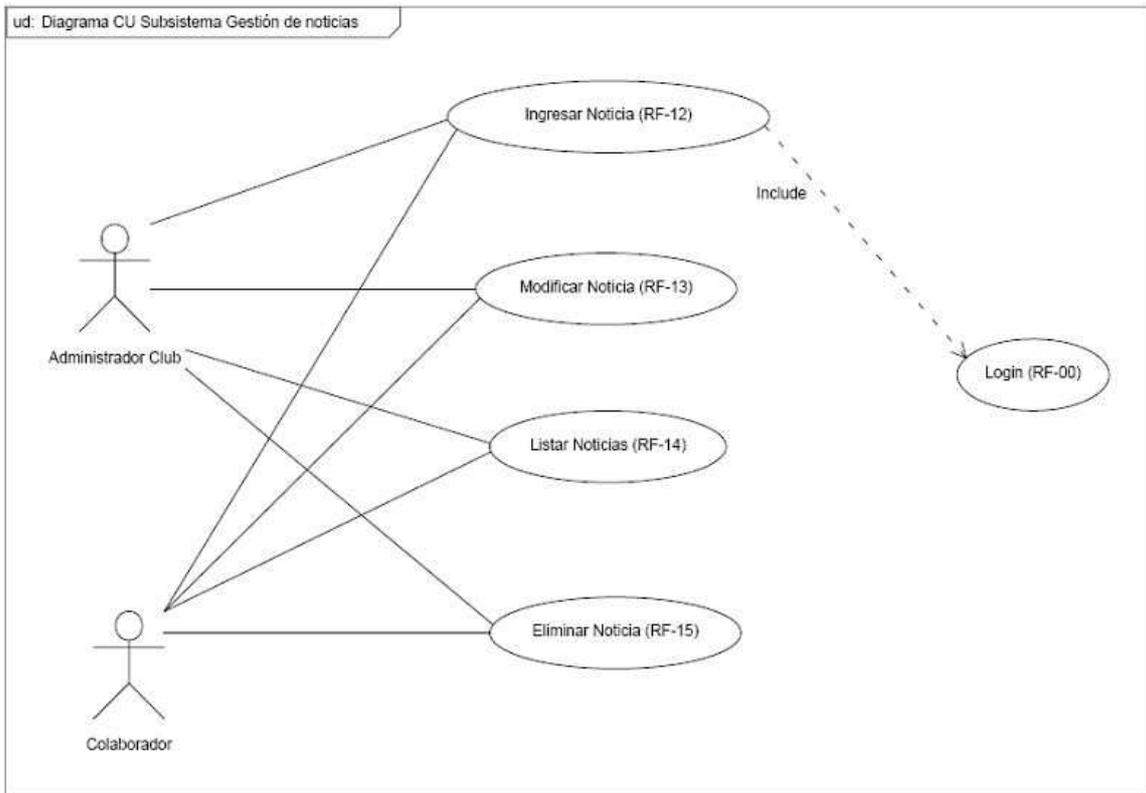


Diagrama 2.8. Casos de uso. Subsistema de gestión de noticias

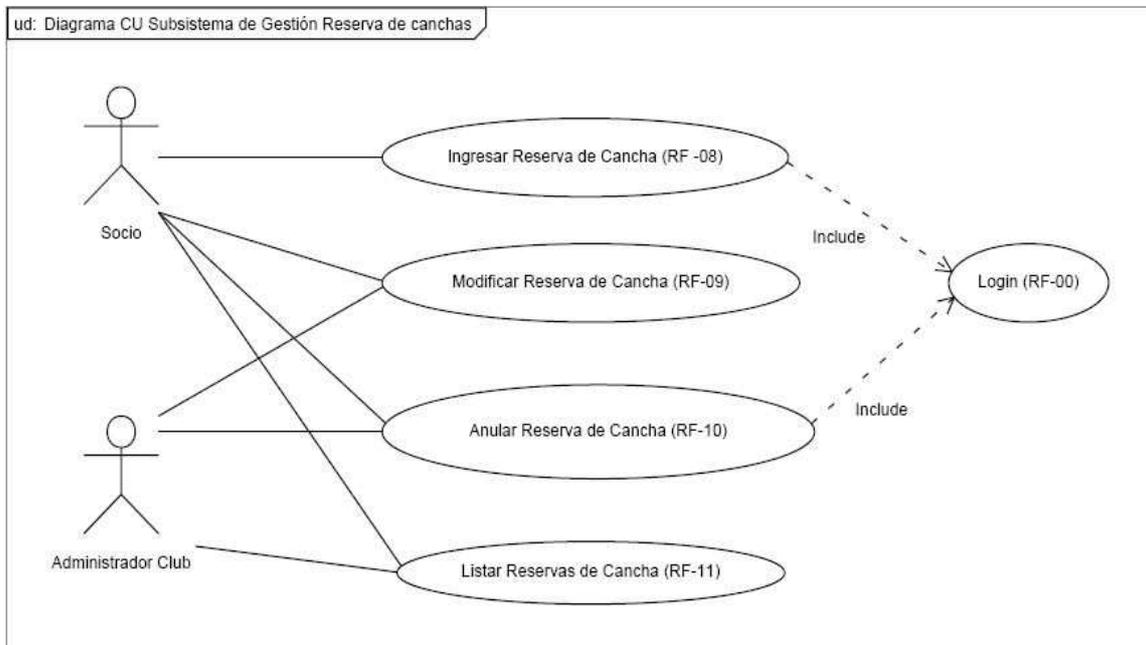


Diagrama 2.9. Casos de uso. Subsistema de gestión de reserva de canchas

5.9 DIAGRAMA DE DOMINIO

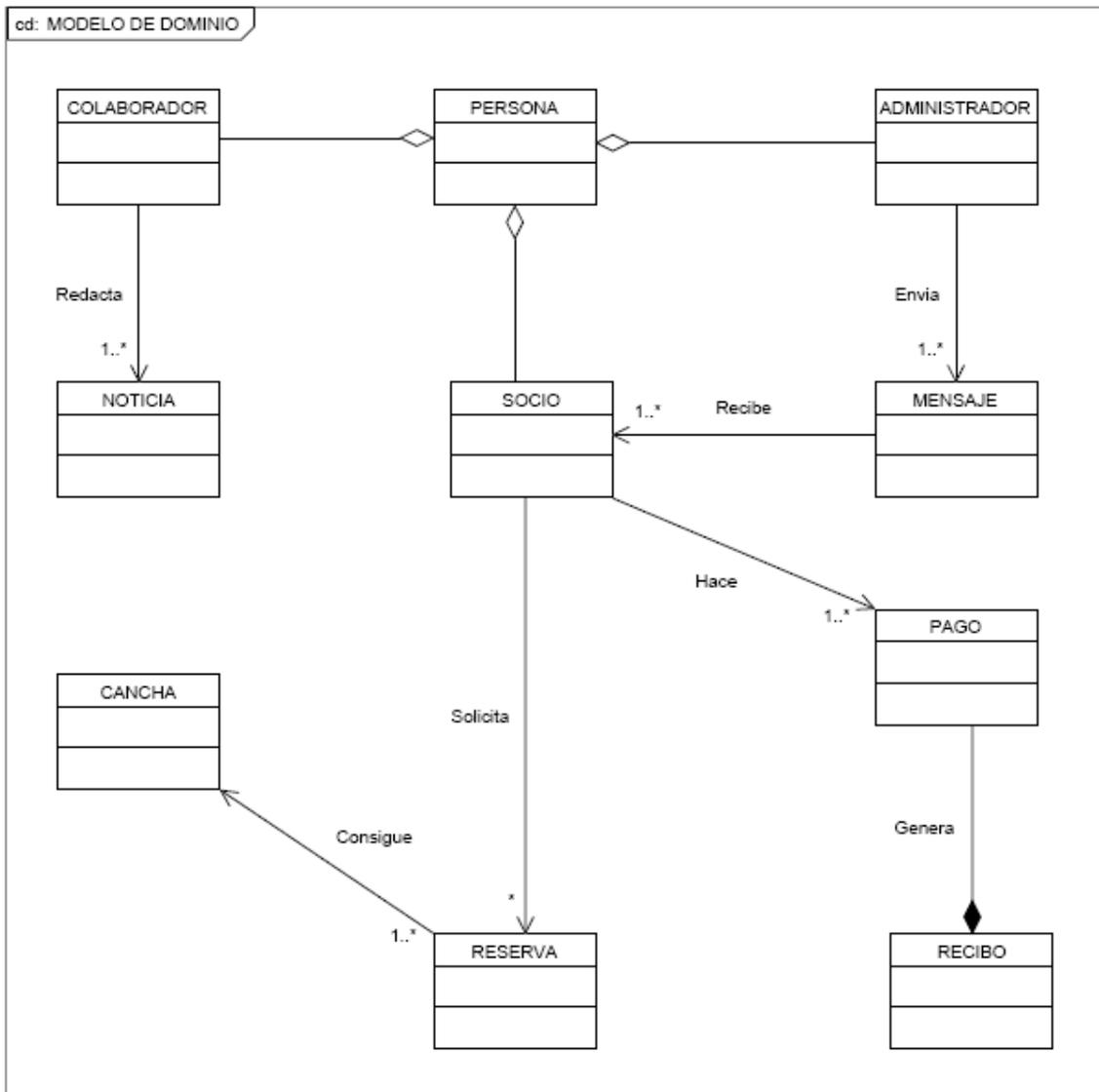


Diagrama 3.1. Diagrama de dominio de la aplicación

5.10 DIAGRAMA DE CLASES

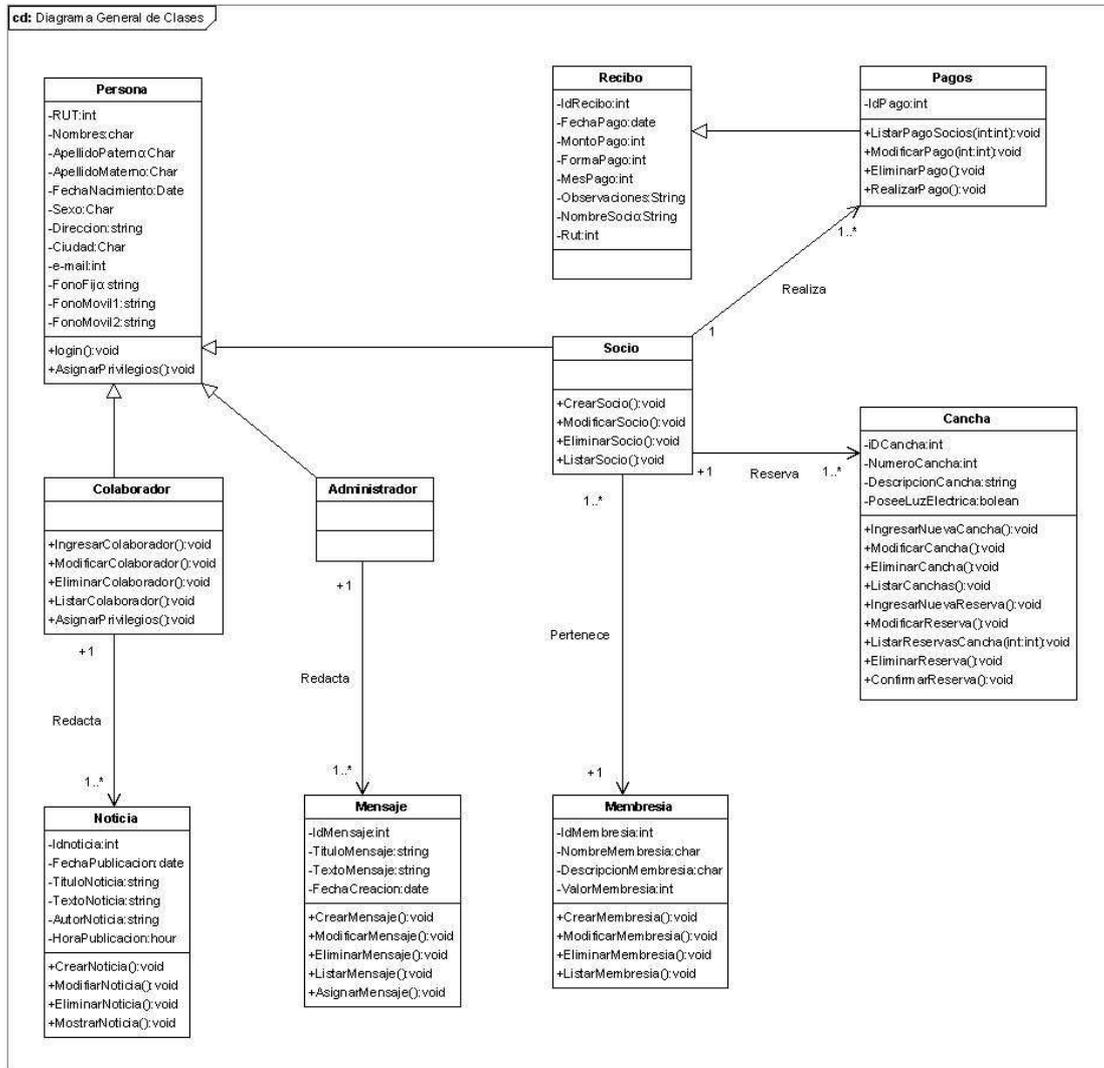


Diagrama 3.2. Diagrama de clases

5.11 TRASPASO DESDE DIAGRAMA DE CLASES A MODELO RELACIONAL

El Diagrama de Clase presenta un mecanismo de implementación neutral para modelar los aspectos de almacenamiento de datos del sistema. Las clases persistentes, sus atributos, y sus relaciones pueden ser implementadas directamente en una base de datos orientada a objetos. Aun así, en el entorno de desarrollo actual, la base de datos relacional es el método más usado para el almacenamiento de datos.

Es en el modelado de esta área donde UML es deficiente. El diagrama de clase de UML se puede usar para modelar algunos aspectos del diseño de bases de datos relacionales, pero no cubre toda la semántica involucrada en el modelado relacional, mayoritariamente la noción de atributos clave que relacionan entre sí las tablas unas con otras. Para capturar esta información, un Diagrama Entidad – Relación (ER) se recomienda como extensión a UML.

El Diagrama de Clases se puede usar para modelar la estructura lógica de la base de datos, independientemente de si es orientada a objetos o relacional, con clases representando tablas, y atributos de clase representando columnas. Si una base de datos relacional es el método de implementación escogido, entonces el diagrama de clase puede ser. Las clases persistentes y sus atributos hacen referencia directamente a las entidades lógicas y a sus atributos; el modelador dispone de varias opciones sobre cómo inferir asociaciones en relaciones entre entidades. Las relaciones de herencia son referenciadas directamente a súper o sub relaciones.

Al implementar el diseño relacional, es una estrategia encaminada a hacer referencia al diagrama de relación de entidad lógico a un diagrama físico que represente el objetivo, el RDBMS. El diagrama físico puede ser desnormalizado para lograr un diseño de base de datos que tiene tiempos eficientes de acceso a los datos. Las relaciones super-sub entre entidades se resuelven por las estructuras de tablas actuales.

Para definir las tablas que serán constituyentes de la Base de Datos, se identifican las clases persistentes desde el diagrama de clases. Estas son:

- Socios
- Membresía
- Cancha
- Noticia
- Pagos
- Reservas

5.12 MODELO DE LA BASE DE DATOS

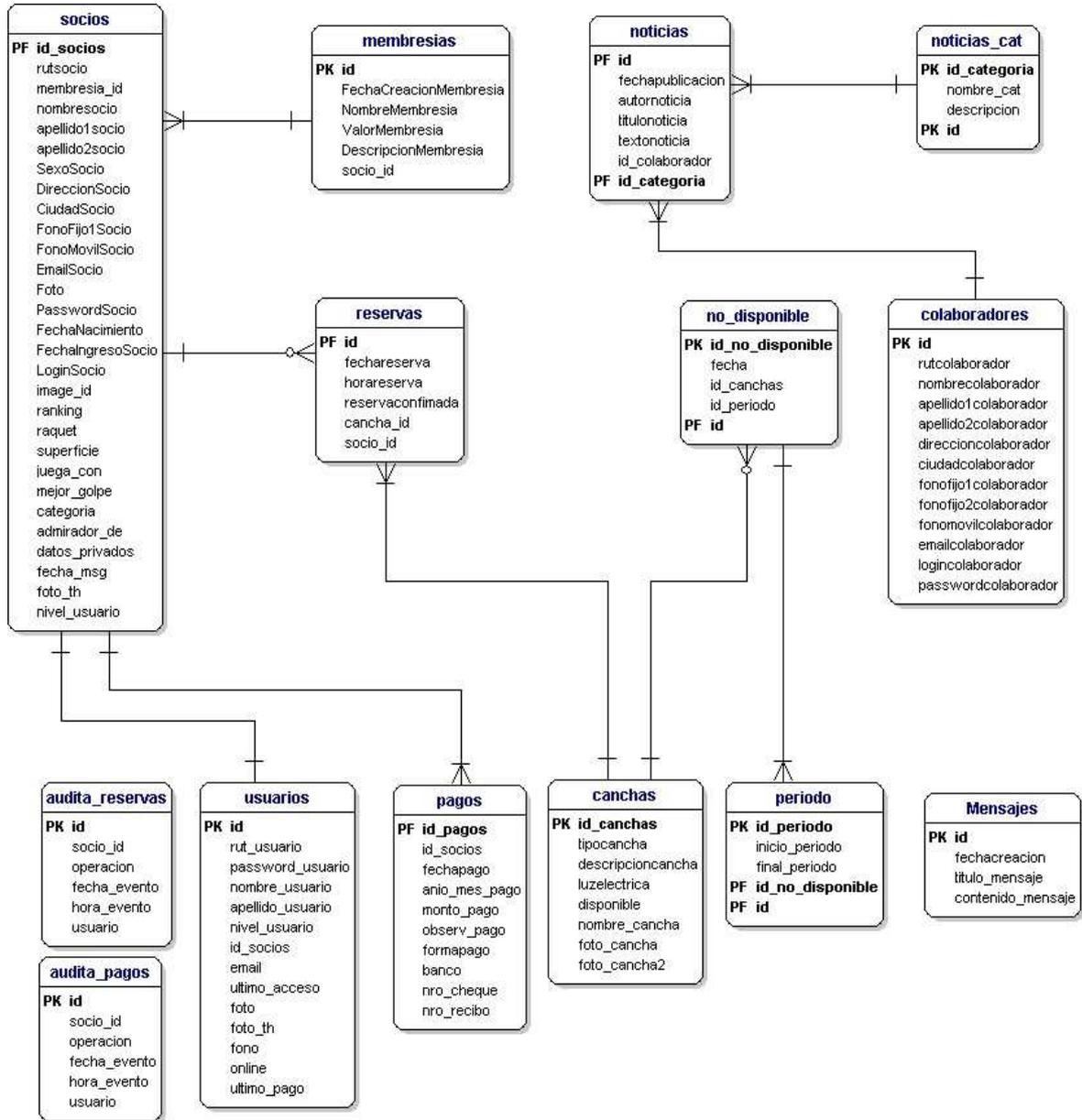


Diagrama 3.3. Diagrama del modelo relacional de la base de datos

5.13 DISEÑO DE INTERFACES DE USUARIO

Un diseño cuidadoso de la interfaz de usuario es parte fundamental del proceso de diseño general del software. Un buen diseño de la interfaz de usuario es crítico para la confiabilidad del sistema, es por eso que se ha decidido seguir una serie de principios [15] para el diseño de estas, los cuales se presentan a continuación:

Principio	Descripción
Familiaridad del usuario	La interfaz debe utilizar términos y conceptos obtenidos de la experiencia de las personas que más utilizan el sistema.
Uniformidad	Siempre que sea posible, la interfaz debe ser uniforme en el sentido de que las operaciones comparables se activen de la misma forma.
Mínima sorpresa	El comportamiento del sistema no debe provocar sorpresa en los usuarios.
Recuperabilidad	La interfaz debe incluir mecanismos para permitir a los usuarios recuperarse de los errores.
Guía de usuario	Cuando ocurran errores, la interfaz debe proporcionar retroalimentación significativa y características de ayuda sensible al contexto.
Diversidad de usuarios	La interfaz debe proporcionar características de interacción apropiadas para los diferentes tipos de usuarios del sistema.

A continuación se muestran algunas interfaces del sistema, éstas fueron diseñadas en un principio y fueron refinadas con el pasar de las iteraciones.



The image shows a web interface for a tennis club. At the top, there is a banner with a collage of tennis-related images: a player in a red shirt, a tennis racket, a yellow tennis ball, and a player in a red and white shirt. The text on the banner reads "VIÑA DEL MAR LAWN TENNIS CLUB" at the top right, "SISTEMA DE RESERVAS Y CONTROL DE PAGOS" at the bottom left, and "CLUB INGLÉS" at the bottom right. Below the banner is a login form titled "Iniciar Sesión". The form contains two input fields: "Rut" and "Contraseña". Below these fields are three buttons: "Entrar!", "Olvidaste tu Contraseña.", and "Volver".

Interfaz gráfica 1.1. Logueo de usuario

Esta interfaz es la interfaz de ingreso al Sistema de reservas. Acá son ingresados los datos de los distintos usuarios del sistema, como es el Rut y su respectiva contraseña, los cuales son proporcionados por el administrador del sistema mediante el ingreso de nuevo socio o colaborador.



Tu último inicio de sesión fue el jueves 26 de junio del 2008 a las 18:06:10 en este momento tenemos 1 visitas en línea

	Perfil Aquí Modificar nuestros datos como correo electrónico, email, contraseña Submenús: Ver mis datos , Modificar mis datos
	Socios Administrar Socios, modificar, agregar, editar, borrar Submenús: Gestionar Socios
	Noticias Aquí podremos añadir, editar, borrar noticias crear categorías Submenús: Añadir Noticia , Editar Noticias , Añadir Categoría , Editar Categoría
	Mensajes Aquí podremos enviarte mensajes a otros socios... Submenús: Mis Mensajes Enviar Mensaje a todos los socios
	Reserva de Canchas Aquí es donde añadimos y editamos nuestras reservas. Submenús: Ver Reservas Reservas para hoy Reservas por socio
	Gestionar Canchas Canchas Submenús: Gestionar Canchas , Agregar No Disponibilidad de canchas , Modificar No Disponibilidad de canchas
	Gestionar Membresías Membresías Submenús: Gestionar Membresías
	Pagos Aquí puede ver sus pagos deudas, etc.... Submenús: Ingresar Pagos , Ver Pagos por Mes , Ver Socios Morosos

Interfaz gráfica 1.2. Menú principal para perfil socios

Esta interfaz es el home, es la segunda página que ve el usuario luego de que ha ingresado correctamente en el sistema. Acá se carga el menú dependiendo del tipo de usuario que ha ingresado en el sistema. En este caso el usuario identificado es el administrador el cual es único, y es el que puede ejecutar una gran cantidad de funciones, entre ellas, gestionar socios, Ingresar pagos, gestionar canchas y membresías, noticias, reservas, etc.



Registro de nuevo socio

<p>Rut <input type="text" value="12598619-6"/></p> <p>Apellido Paterno <input type="text"/> Ingrese el Apellido Paterno</p> <p>Fecha de Nacimiento <input type="text" value="17-1-1979"/> <input type="button" value="Seleccionar fecha"/></p> <p>Dirección <input type="text"/> Ingrese la Dirección del Socio</p> <p>Fono Fijo <input type="text" value="6546545"/></p> <p>Email Socio <input type="text" value="test_sqa@gmail.com"/></p> <p>Membresías <input type="text" value="ADULTOS [VARONES]"/></p> <p><input type="button" value="Enviar"/></p>	<p>Nombres socio <input type="text"/> Ingrese el Nombre del nuevo socio</p> <p>Apellido Materno <input type="text"/> Ingres el Apellido Materno</p> <p>Sexo <input type="text" value="Masculino"/></p> <p>Ciudad <input type="text"/> Ingrese la Ciudad</p> <p>Fono Movil <input type="text" value="096546555"/></p> <p>Fecha de Ingreso <input type="text" value="30-6-2008"/> <input type="button" value="Seleccionar fecha"/></p>
--	--

Interfaz gráfica 1.3. Ingreso de nuevo socio. Perfil Administrador

Esta interfaz permite el ingreso de un nuevo socio al sistema. Se ingresan todos los datos validados y se envía un correo electrónico al socio para avisar al socio de la creación de su cuenta de usuario en el sistema.



Editar Socio: Humberto Fuentealba

Nombres socio

Apellido Materno

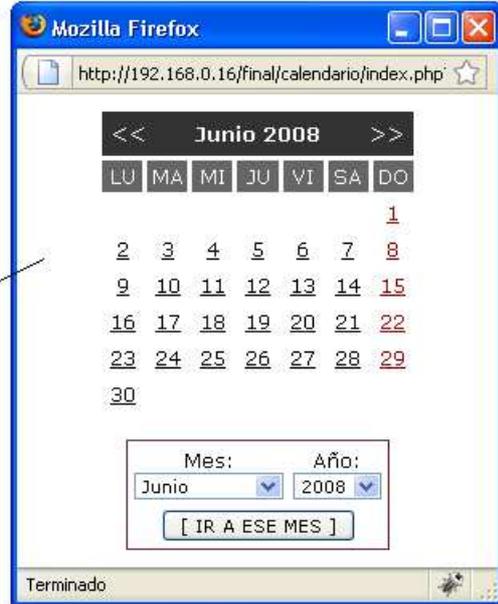
Dirección

Fono Fijo

Email Socio

Fecha de Ingreso

Seleccione una fecha



Interfaz gráfica 1.4. Editar datos de socios. Perfil Administrador

Interfaz de usuario para que el administrador del sistema pueda cambiar los datos de ingreso de los socios.



Interfaz gráfica 1.5. Cambio de contraseña desde perfil socio

Interfaz de usuario con perfil de socio que permite cambiar los datos personales, agregar datos tenísticos, cambiar password y fotografia.



[Arriba](#)

Interfaz gráfica 1.6. Vista de datos personales desde perfil socio



Lista de Socios

Nombre	Fono Fijo	Fono Movil	Email	Acciones
Ivan Mercado Valdes	2343234	35445544	imercado@ucv.cl	Ver - Editar Agregar Pago
José Miguel Cifuentes Henriquez	53541490	0965452218	josemiguel.cifuentes@gmail.com	Ver - Editar Agregar Pago
Rodrigo Alejandro Alvarez Martinez	65496875	096545552	rodrigo_a@gmail.com	Ver - Editar Agregar Pago

[Ir arriba](#)

Interfaz gráfica 1.7. Listado de socios desde perfil administrador

Vista del perfil administrador en la cual se puede ver un listado completo de todos los socios registrados en la base de datos. Existen links para ver la información del socio, editar sus datos y agregar pagos para cada uno de ellos.



Ingresar mi Reserva

Fecha de Ingreso

Reservar para el día: 2-7-2008

Inicio - Fin	Cancha 1	Cancha 2	Cancha 3	Cancha 4	Cancha 5	Cancha 6
09:00 - 09:50	Disponible	Disponible	Disponible	Disponible	No disponible	Disponible
10:00 - 10:50	Disponible	Disponible	Disponible	Disponible	No disponible	Disponible
11:00 - 11:50	Disponible	Reservada	Disponible	Disponible	No disponible	Disponible
12:00 - 12:50	Disponible	Disponible	Disponible	Disponible	No disponible	Disponible
13:00 - 13:50	Disponible	Disponible	Disponible	Disponible	No disponible	Disponible
14:00 - 14:50	Disponible	Disponible	Disponible	Disponible	No disponible	Disponible
15:00 - 15:50	Disponible	Disponible	Disponible	Disponible	No disponible	Disponible
16:00 - 16:50	Disponible	Disponible	No disponible	Disponible	Disponible	Disponible
17:00 - 17:50	Disponible	Disponible	No disponible	Disponible	Disponible	Disponible
18:00 - 18:50	Disponible	Disponible	No disponible	Disponible	Disponible	Disponible
19:00 - 19:50	Disponible	Disponible	No disponible	Disponible	Disponible	Disponible
20:00 - 20:50	Disponible	Disponible	No disponible	Disponible	Disponible	Disponible

Interfaz gráfica 1.8. Reservas de canchas desde perfil socio.



Todos mis pagos

Fecha Pago	Mes pagado	Monto Pagado	Forma de Pago	Banco	Nro.Cheque/Transac.	Nro.Recibo
26-06-2008	Mayo, 2008	\$ 25000	Traspaso electrónico	Banco BCI	3242324	223454
15-05-2008	Abril, 2008	\$ 25000	Cheque	Santander Santiago	4423423	23433
09-04-2008	Marzo, 2008	\$ 25000	Cheque	Banco BCI	324234	23423

[Volver al menu](#)

[Ir arriba](#)

Interfaz gráfica 1.9. Listado pagos de socios desde perfil administrador

Vista del perfil de socio, en la cual se listan todos sus pagos ordenados por fecha decreciente.

5.14 ARQUITECTURA LOGICA DEL SISTEMA

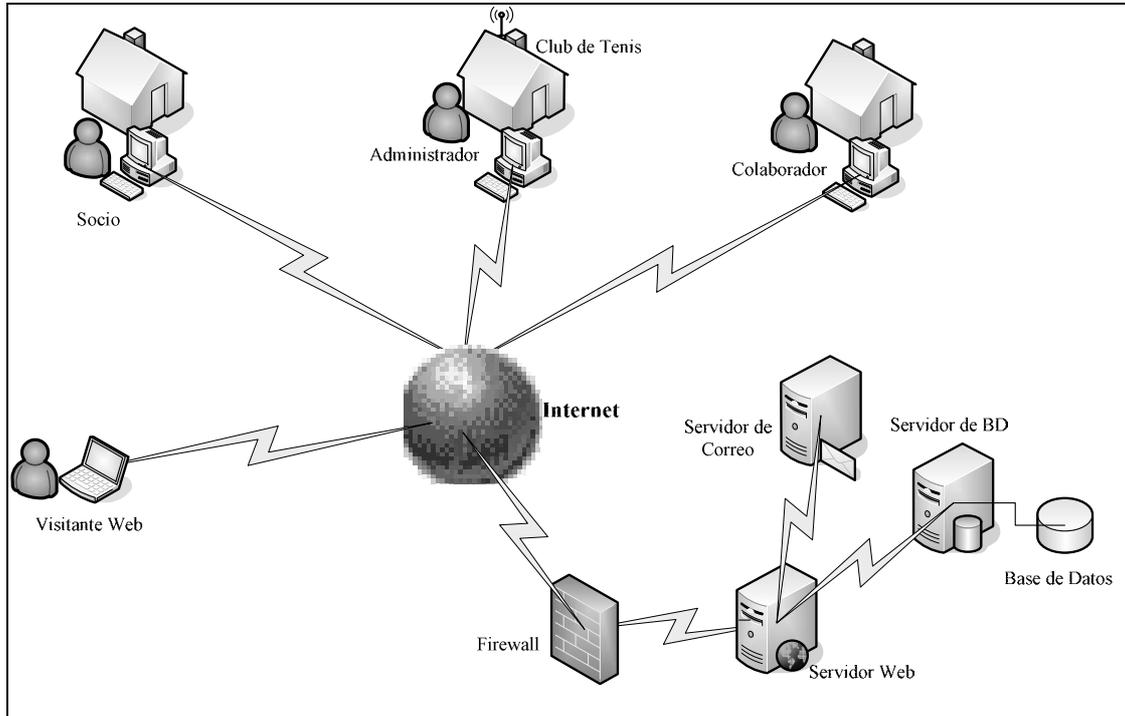


Diagrama 3.4. Arquitectura lógica del sistema.

Los usuarios se conectan a través de browser a sus respectivos ISP, los cuales resuelven las direcciones IP a las cuales conectarse para concretar las transacciones. Los componentes y etapas se detallan a continuación:

Rol del servidor Web:

- Recibir los pedidos del cliente
- Buscar las páginas en su disco
- Enviar las páginas al cliente
- Permanece en estado Pasivo en espera de la llegada de clientes

Rol del cliente Web:

- Enviar pedidos al servidor
- Recibir las páginas HTML
- Armar la visualización de las páginas
- Reaccionar ante la interacción del usuario
- Permanece en estado Activo Envía los pedidos de conexión
- Cada máquina tiene una dirección (IP)
- Cada proceso en cada máquina tiene un número (Puerto TCP)

Protocolo HTTP

- Protocolo de transferencia de hipertextos
- Montado sobre TCP/IP
- Puerto predeterminado 80
- Recibir las páginas HTML
- Armar la visualización de las páginas
- Reaccionar ante la interacción del usuario

Todas las transacciones implican resguardos respecto de la seguridad de los servidores Web, correo y Base de datos, como lo son: activación de módulos de seguridad en servidor Web Apache, Configuración de script de seguridad a nivel de lenguaje de desarrollo PHP, configuración de sesiones de usuario y accesos recurrentes a nivel de Base de datos.

5.15 ARQUITECTURA FISICA DEL SISTEMA

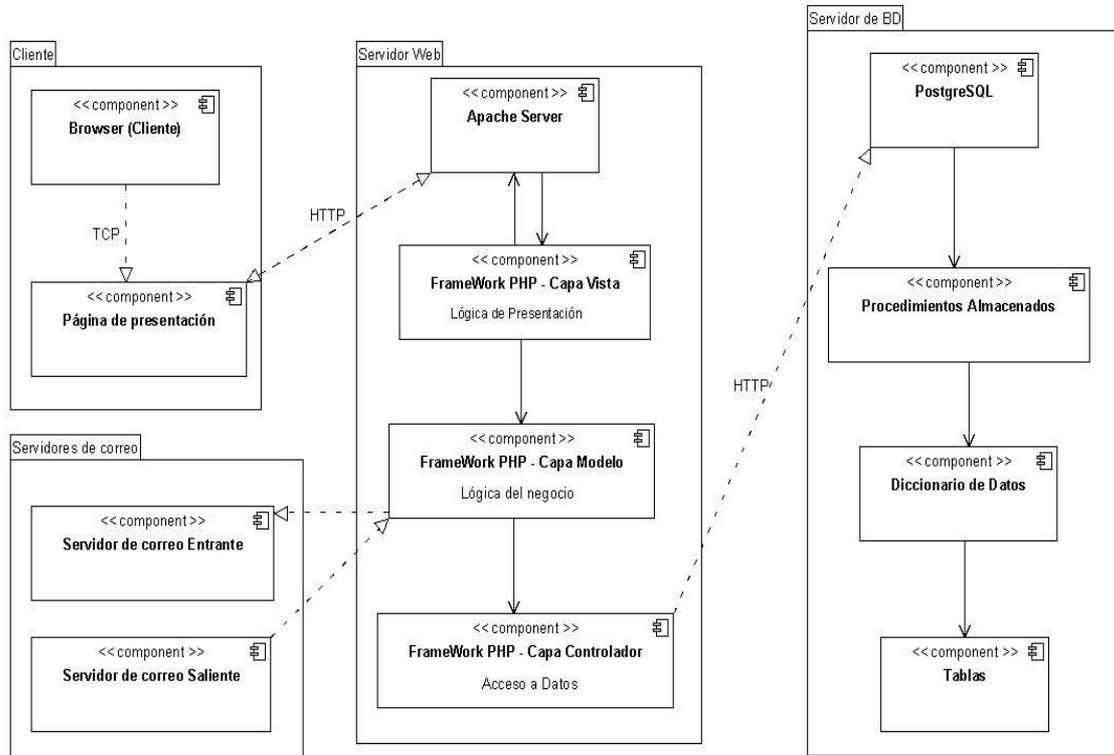


Diagrama 3.5. Arquitectura física del sistema

La arquitectura, mirada desde la óptica de los componentes físicos está planteada según el diagrama 3.6 cuya descripción particular se presenta en este apartado:

Computador Cliente:

Los diferentes clientes ocuparán sus computadores equipados con browser (sistema será optimizado para Internet Explorer y Mozilla Firefox). La conexión a Internet en el club inglés de tenis es a través de red de área local LAN y WiFi.

Servidor Web:

El servidor Apache estará instalado en una máquina del proveedor de Hosting, junto con el núcleo de componentes del lenguaje de programación PHP.

Toda la capa de negocios y lógica de la aplicación estará contenida en este servidor.

Servidores de correo:

Los servidores de correo estarán instalados en máquinas dedicadas a la gestión de correo entrante y saliente. SMTP y POP3 gestionan el correo entrante y saliente.

El proveedor de hosting tiene implementado estos servicios y han sido probados con resultados satisfactorios.

Servidor de Bases de datos:

El DBMS Postgresql es un potente gestor de base de datos. Se debe instalar lenguaje PL/Pgsql para poder ejecutar los procedimientos almacenados, triggers y funciones.

6 USABILIDAD

La usabilidad es la característica de un sistema que pretende ser utilizado por:

- El tipo o tipos de usuario/s
- La tarea o tareas que para las cuales el sistema se ha hecho
- El contexto en que se da la interacción.

El grado de usabilidad de un sistema es, por su parte, una medida empírica y relativa de la usabilidad del mismo. El concepto de usabilidad se refiere a una aplicación informática de software o un aparato (hardware), aunque también puede aplicarse a cualquier sistema hecho con algún objetivo en particular.

6.1 PRUEBAS DE USABILIDAD A IMPLEMENTAR

Se evaluará la usabilidad del presente sistema apoyándose en las siguientes pruebas de usabilidad:

- Evaluación heurística
- Test de usabilidad

6.1.1 EVALUACION HEURISTICA

La entrevista de evaluación heurística es una técnica exploratoria para conocer la opinión de los usuarios. Útiles para hacernos una idea general de sus preferencias y poder enfocar posteriormente el diseño o el rediseño de nuestro sitio o sistema hacia algo más usable. La idea es poder captar de primera mano y mediante el discurso que desarrolla el usuario hablando de nuestro sistema el grado de satisfacción, opinión acerca del contenido, de su organización, de su diseño gráfico, etc. Mediante ella, queremos saber los porqués de su aceptación o rechazo, y lo que hay detrás de los mismos. Evidentemente, necesitamos usuarios reales del sistema y que la comunicación, en la medida de lo posible, sea física y cara a cara. Gran parte de la información nos la pueden proporcionar sus pausas, gestos y comunicación no verbal.

Una de las ventajas de la evaluación por criterios es su bajo costo, en realidad este tipo de evaluación puede tener el costo que se desee. Un número mínimo de tres evaluadores permite realizar una evaluación por criterios. Los costos son por tanto mucho menores que cualquier otro método de evaluación.

En comparación con otras técnicas de evaluación donde el observador debe interpretar las acciones del usuario, en la evaluación heurística no es necesaria la interpretación externa, porque las ideas, comentarios e información elaborada por los evaluadores está contenida en sus informes. Otra ventaja es que en la evaluación por criterios es posible interrogar a los evaluadores, profundizar en determinadas cuestiones de interés y ayudarles cuando tienen problemas. En los tests de usuario por el contrario, los usuarios no deben disponer de más información que la necesaria para permitir su comportamiento espontáneo.

Detectar previamente los problemas más graves de usabilidad en la evaluación por criterios permite realizar posteriormente pruebas de usuarios con mejores resultados.

En base a los mismos principios heurísticos, diferentes personas encuentran diferentes tipos de problemas en un sitio Web, por ello se han de utilizar varios evaluadores.

Los estudios de Nielsen(12) muestran que un número de entre 3 y 5 evaluadores es suficiente para la evaluación por criterios de un sitio Web. Este número puede ser incrementado en caso de que la usabilidad sea crucial para el éxito del sitio Web.

Esta evaluación detecta un porcentaje significativo de los problemas graves de diseño y otros problemas menores, dependiendo del número de evaluadores que revisen el sitio.

Procedimiento

- Los evaluadores inspeccionan los sitios Web individualmente y sólo después de la evaluación pueden comunicarse sus hallazgos.

- Las sesiones de evaluación duran aproximadamente una o dos horas por página. Los evaluadores utilizan una checklist de criterios y cuando sea necesario, incorporarán nuevos principios a las categorías existentes a su "checklist" de criterios.

- Los evaluadores se juntan, discuten y explican los problemas encontrados por cada uno, haciendo una lista general que incluye la suma de todos los problemas encontrados por los evaluadores.

- Los evaluadores no sólo elaboran una lista de problemas de usabilidad en el sitio, sino que han de explicar los problemas de acuerdo con los principios de la usabilidad.

- La jerarquización de la gravedad de los problemas del sitio facilitará la aplicación posterior de políticas de rediseño del sitio Web efectivas. La gravedad de los problemas de usabilidad es medida por tres factores:

- La frecuencia con la que el problema ocurre, ¿es común o poco frecuente?

2. El impacto o severidad del problema cuando sucede, ¿es fácil o difícil para los usuarios superarlo?

3. La persistencia del problema, ¿el problema es resuelto la primera vez que se use el sitio Web o aparece repetidamente?

La escala de medición para la evaluación va desde 0 a 4, siendo 0 el valor para representar la inexistencia del problema y 4 el valor correspondiente a una ocurrencia extrema.

Tabla usada para tabular los puntajes de los problemas encontrados por los evaluadores.

	Listado de problemas encontrados	Observador 1			Observador 2			Observador 3			Promedios		
		S	F	C	S	F	C	S	F	C	S	F	C
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													
13													

S Severidad
F Frecuencia
C Criticidad

6.1.2 TEST DE USABILIDAD

Estudia el comportamiento de los usuarios al ejecutar determinadas tareas. Para ponerlo en marcha, se entrega a los participantes una lista predefinida de tareas que deben cumplir. El test consiste en que los participantes ejecuten las tareas y al mismo tiempo expresen en voz alta lo que hacen, lo que piensan y lo que sienten. Para completar el test se suele hacer posteriormente una entrevista.

Mientras los usuarios hacen las tareas, hay observadores que toman nota de los comportamientos. Mientras el usuario realiza los ejercicios, el conductor de la prueba debe intervenir lo menos posible. Es recomendable que sólo hable para aclarar dudas sobre la tarea, para hacer preguntas puntuales pero nunca para contestar a preguntas directas del usuario sobre la interfaz. Por ejemplo, si el usuario pregunta "¿esta sección es donde se descargan los programas?", lo apropiado es que el conductor de la prueba responda devolviendo la pregunta: "tú qué crees, es esa la sección donde se descargan los programas?".

También se pueden implementar sistemas más complejos de observación, por ejemplo hay programas que graban los movimientos de teclado y ratón para después estudiarlos más detenidamente.

Pasos para ejecutar el test de usabilidad

- Empezar registrando la sesión con una videocámara, orientándola de modo que se capturen las manos del usuario interactuando con el sistema.
- Seguir el protocolo para la sesión
- Observar al participante a medida que evoluciona en sus tareas
- Registrar cualquier incidente crítico que tenga lugar y considerar proseguir, cuando ello ocurra, con una o dos preguntas abiertas (¿tiene algún problema?, ¿está bloqueado?, ¿necesita una pista?, ¿es ese el resultado que buscaba?, ¿resulta más difícil de lo que debería?)
- Tras los incidentes de carácter positivo (¿se siente ahora más confiado?, ¿encontró alguna pista específica que le ayudara a resolver el problema?, ¿qué le hizo pensar que funcionaría esta aproximación?)
- Tomar notas de forma cuidadosa en tiempo real, pues va a resultar más eficiente que haciéndolo de forma retrospectiva
- Agradecer y gratificar al participante.

7 PLAN DE PRUEBAS

Una parte muy importante para verificar errores y asegurar que se cumplen los requerimientos, es el plan de pruebas. Para diseñar el plan de pruebas, se ha escogido utilizar pruebas tipo caja negra. Las pruebas de caja negra, también denominadas pruebas de comportamiento, se centran en los requisitos funcionales del software, esta intenta encontrar errores como: funciones incorrectas o ausentes, errores de interfaz, errores de rendimiento, errores de inicialización y de terminación. Los casos de prueba han sido obtenidos a partir de los casos de uso especificados. En cada caso de prueba se especifican los pasos a seguir, para cada uno de los cuales hay ciertas entradas válidas e inválidas las que producirán una salida (Para ver en detalle los casos prueba dirigirse al Anexo D).

8 CONSTRUCCION DEL SISTEMA

8.1 CODIFICACION

Como ya se ha especificado, la herramienta de programación que se utilizó es PHP. Antes de empezar a programar, se establecieron ciertas normas entre los desarrolladores de este sistema para evitar ciertos problemas que suelen ocurrir cuando se trabaja en equipo y se codifica un sistema. Algunas de las normas establecidas son:

- Para algoritmos de complejidad media/alta siempre poner comentarios, para así poder tener mejor comunicación entre los desarrolladores y así comprender mejor la estructura del código.
- Crear respaldos del código cada cierto tiempo para poder mitigar una posible pérdida de información.
- Iniciar código PHP siempre con `<?php` y cerrar con `?>`.
- Todas las páginas son creadas gracias a una página base que tiene el diseño general para todas las páginas.
- Todas las páginas creadas deben ser de extensión `.php`.

En anexo C, se detallan las principales clases utilizadas en la codificación del sistema.

8.2 SEGURIDAD

Durante los últimos años los servidores Web se han convertido en una excelente fuente de diversión para piratas: cualquier empresa que se precie, desde las más pequeñas a las grandes multinacionales, tiene una página Web en las que al menos trata de vender su imagen corporativa. Si hace unos años un pirata que quisiera atacar a una empresa (y no a todas, ya que muy pocas tenían representación en la red) se las tenía que ingeniar para obtener primero información de la misma y después buscar errores de configuración más o menos comunes de sus sistemas, hoy en día le basta con teclear el nombre de su objetivo en un navegador y añadir la coletilla `“.com”` detrás del mismo para contactar con al menos una de sus máquinas: su servidor Web.

La mayor parte de estos ataques tiene éxito gracias a una configuración incorrecta del servidor o a errores de diseño del mismo: si se trata de grandes empresas, los servidores Web suelen ser bastante complejos (alta disponibilidad, balanceo de carga, sistemas propietarios de actualización de contenidos) y difíciles de administrar correctamente, mientras que si la empresa es pequeña es muy posible que haya elegido un servidor Web simple en su instalación y administración pero en el cual es muy difícil garantizar una mínima seguridad.

Sea por el motivo que sea, la cuestión es que cada día es más sencillo para un pirata ejecutar órdenes de forma remota en una máquina, o al menos modificar contenidos de forma no autorizada, gracias a los servidores Web que un sistema pueda albergar.

Por lo descrito precedentemente, la seguridad es un punto crítico en cada sistema computacional, existen documentos los cuales proponen buenas prácticas en sistemas informáticos, un buen ejemplo de esto es la norma Iso 17799/ Bs. 7799. Además existe un documento que especifica la norma nacional de seguridad (Nch 2777). Al hablar del concepto de seguridad informática se deben tener en cuenta tres conceptos, que conforman la triada de seguridad en cada sistema informático, según la norma Nch 2777:

- **Confidencialidad:** Asegura que la información sea accesible solo por aquellos usuarios autorizados a hacerlo.
- **Integridad:** Salvaguardar que la información y los métodos de procesos sean exactos y completos.
- **Disponibilidad:** asegurar que los usuarios autorizados tengan acceso a la información y bienes asociados cuando lo requieran.

En este sistema se puso especial atención en la *confidencialidad*, a continuación se explicará lo hecho:

- **Manejo de Claves:** Las claves de usuario son enviadas encriptadas a la base de datos, de manera que sea imposible obtener la clave real en la base de datos.
- **Manejo de Sesión:** La sesión comienza cuando el usuario ingresa al sistema, con el comando `SESSION_START()`, luego se debe setear la variable `$_SESSION['ID']`, a la cual se le asigna un número aleatorio y luego es encriptada la ID mediante la función MD5, luego en cada interfaz del sistema se pregunta si la sesión es válida, la condición es que la variable `$_SESSION['ID']` haya sido asignada y encriptada.
- **Manejo de Permisos:** Este sistema trabaja bajo el concepto de perfiles de usuario, es decir, se listan las funciones del sistema según el perfil del usuario. Tan importante como darle permiso a los usuarios correctos, es denegarle los permisos a aquellos que no los posean, es por ello, que en este sistema en cada interfaz se valida que el perfil de usuario de dicho usuario sea el esperado, de lo contrario no puede ver aquella página.
- **Fin de Sesión:** Se lleva a cabo con el comando `SESSION_DESTROY()`, ocurre cuando un usuario sale del sistema. Si se desea hacer “back” en el navegador luego de que se haya cerrado la sesión no podrá ingresar al sistema, pues la sesión expiró.

9 CONCLUSIONES

Acerca del proyecto en general:

El sistema de reserva de canchas de tenis es un desafío importante para el equipo de desarrollo, ya que supone el dominio de técnicas de alto nivel en lo que respecta a ingeniería de software, programación y manejo de herramientas de desarrollo de software. Estos conocimientos han tenido que ser reforzados a través de la lectura de textos especializados y consulta a páginas Web especializadas en ingeniería de software y desarrollo Web. Por otro lado el levantamiento de requerimientos es una actividad no trivial, ya que la concreción de entrevistas con distintos actores de la organización a veces se torna complicada por tiempo y disposición de las personas. Es necesario por lo tanto actuar rápido y obtener la información más relevante para el proyecto. Para el caso de este proyecto, se ha podido obtener información necesaria como para afirmar que sobre el 100% de los requerimientos funcionales están considerados y tabulados.

Respecto de la metodología de desarrollo, se optó por el desarrollo de software usando el modelo UP (Proceso unificado), por lo cual el plan RSGR (Reducción, supervisión y gestión del riesgo) es un aspecto de mucha importancia dentro del proyecto. La definición de la arquitectura del software es un componente muy importante dentro del proyecto; esta va sufriendo modificaciones, conforme se van sucediendo las iteraciones propias del modelo de desarrollo de software UP; sin embargo la base sólida de la arquitectura ha sido definida desde los puntos de vista lógico y físico.

Acerca del trabajo realizado:

Se realizó el análisis de factibilidad del proyecto, el cual ha demostrado que éste puede llegar a buen término. La factibilidad legal del proyecto quedó claramente establecida, ya que por el hecho de que la organización es una entidad sin fines de lucro, las restricciones al respecto hacían peligrar una parte importante de la funcionalidad del sistema; sin embargo esta variable ha sido estudiada y manejada con diligencia por parte de los directivos del club de tenis y del equipo a cargo del desarrollo del presente proyecto.

Según el modelo de desarrollo de software UP, se han concluido exitosamente todas las iteraciones presupuestadas para el proyecto y en el aspecto técnico referido a la herramienta de desarrollo cakePHP utilizada, se hizo un proceso intensivo de aprendizaje, con la cual se han implementado un los módulos que componen el sistema a través de la codificación de clases para las distintas funcionalidades del sistema que implican una de las propiedades más deseables en la programación orientada a objetos; la herencia.

La aplicación de test usabilidad ha permitido evidenciar deficiencias funcionales y no funcionales del sistema, lo que facilita su corrección y de esta manera elevar la calidad del software en desarrollo. Se realizaron 3 prototipos para el sitio Web del club inglés los cuales se han sometido a la evaluación y crítica de los usuarios finales, que son los socios, directivos y administración del club inglés de Viña del Mar; junto a ello se hicieron 3 test de usabilidad, cuyos resultados han sido satisfactorios desde el punto de vista de la recopilación de datos y mejoramiento de la funcionalidad evaluada. Las 2 evaluaciones de usabilidad hechas (evaluación heurística y test de usabilidad) y testing QA con el fin de incrementar la calidad de las interfaces gráficas y las validaciones de datos, cumplieron los objetivos primarios descritos para el proyecto.

La revisión constante del software permite detectar deficiencias, cada vez más pequeñas; pero no menos importantes de resolver; sin embargo el sistema ha alcanzado una madurez suficiente como para liberar una primera versión Beta de explotación, la cual fue implantada según las necesidades que el club inglés de tenis estime conveniente.

La culminación exitosa de este proyecto responde a un gran esfuerzo y vocación profesional; no ha estado exento de problemas imponderables; pero de los cuales se ha salido airoso, lo cual re-afirma y deja en evidencia la capacidad de reacción e iniciativa que se necesitaba para que la “obra” terminara de la mejor forma posible. El resultado final ha dejado satisfecho al equipo de desarrollo del proyecto.

10 REFERENCIAS

1. Ingeniería del software, séptima edición. Autor: Ian Sommerville, páginas N° 62, 63, 64, 65. Editorial McGraw Hill.
2. Ingeniería del software, Un enfoque práctico, quinta edición. Autor: Roger Pressman, páginas N° 21, 22,23. Editorial McGraw-Hill.
3. Ingeniería del software, Un enfoque práctico, quinta edición. Autor: Roger Pressman, páginas N° 23, 24. Editorial McGraw-Hill.
4. El Proceso Unificado de Desarrollo de Software. Autores: Ivar Jacobson, Grady Booch, James Rumbaugh, Editorial:Madrid : Addison Wesley, 2006.
5. Object Management Group - UML: <http://www.uml.org>
6. PostgreSQL: The world's most advanced open source database: <http://www.postgresql.org>
7. ¿Qué es el PHP?: <http://www.maestrosdelweb.com/editorial/phpintro/>
8. CakePHP: the rapid development php framework.Home: <http://cakephp.org/>
9. Foros del Web www.forosdelweb.com
10. Apache Resource : <http://www.apache.com/>
11. La Web del Programador: www.lawebdelprogramador.com
12. Gentleware - model to business: gentleware homepage <http://www.gentleware.com/products.html>
13. Paradigma de programación - Wikipedia, la enciclopedia libre: http://es.wikipedia.org/wiki/Paradigma_de_programaci%C3%B3n

ANEXO A: TEST DE USABILIDAD

TEST DE USABILIDAD #1. HOJA DE RESPUESTAS

Registro del usuario:

Nombreusuario: _____

Tareas a realizar:

Logeo como usuario administrador:

Una vez abierto un browser, digitar la url del sistema:
http://200.24.237.135/cake/users/login

- Login: admin
- Password: usabilidad
- Respuesta/Comentarios: _____

Hacer click en la acción VER del socio Juan Carlos Estuardo Lawn, luego hacer click en Ver perfil del socio

Respuesta/Comentarios: _____

Cambiar campo Fono fijo por 5942810. Grabar cambios

Cual es su número de ranking?

Respuesta/Comentarios: _____

Registrar un nuevo socio. Hacer click en nuevo socio.

- Llenar los campos requeridos en el formulario, según estime conveniente

- Respuesta/Comentarios: _____

Clickear en opción LOGOUT

Loguearse con el login y password del usuario creado. Verificar los datos que están registrados para el socio creado.

- Respuesta/Comentarios: _____

Hacer Click en VER MI PERFIL. Modificar el valor del campo nombre foto, por “tenista1” si el socio ingresado por usted es hombre o por “tenista2”, si el socio ingresado es mujer. Hacer click en GRABAR.

Verificar el cambio del campo modificado, haciendo click en acción VER del socio al cual se le modificó el campo.

- Respuesta/Comentarios: _____

Cuestionario

- Responda en qué grado está de acuerdo con cada una de las aseveraciones.
- Trate de seleccionar la opción lo más espontáneamente posible sin meditar la respuesta por mucho tiempo.
- Si siente que no puede responder una pregunta marque el punto central de la escala
- Tiempo:5 minutos

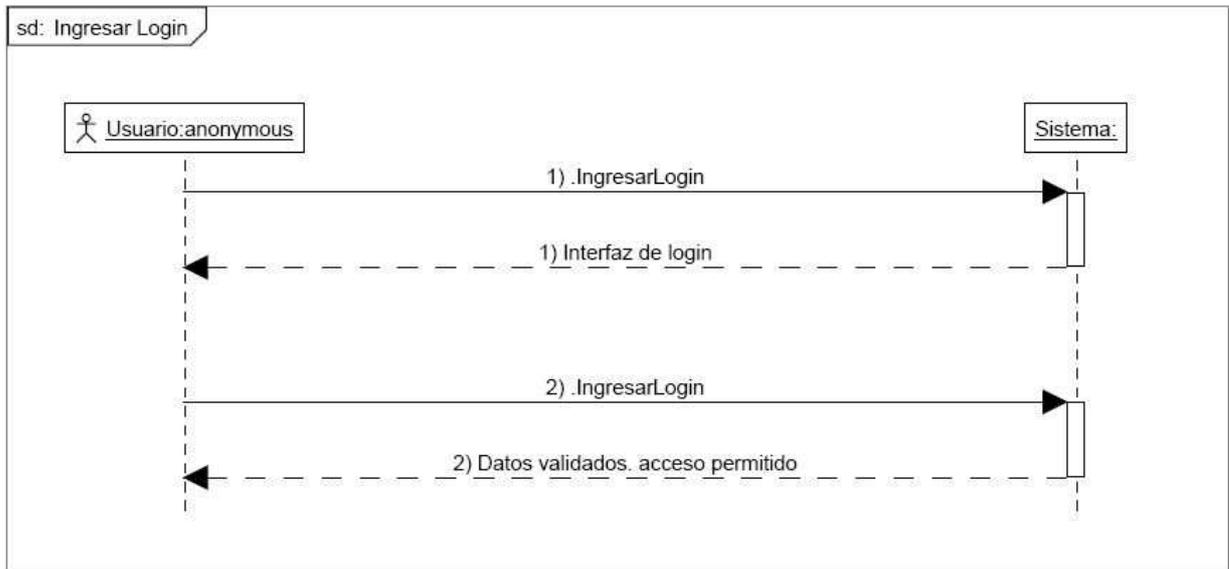
	Completamente desacuerdo			de acuerdo	
1. Pienso que me gustaría usar el sistema frecuentemente	1	2	3	4	5
2. Encontré el sistema innecesariamente complejo	1	2	3	4	5
3. Pienso que el sistema es fácil de usar	1	2	3	4	5
4. Creo que necesitaría del apoyo de un experto para usar el sistema	1	2	3	4	5
5. Encontré que las distintas funciones del sistema están bien integradas	1	2	3	4	5
6. Pienso que hay muchas inconsistencias en este sistema	1	2	3	4	5
7. Imagino que la mayoría de las personas aprenderían muy rápidamente a utilizar este sistema	1	2	3	4	5
8. Encontré el sistema muy engorroso de usar	1	2	3	4	5
9. Me sentí muy confiado en el manejo del sistema	1	2	3	4	5
10. Tuve que aprender muchas cosas antes de poder usar el sistema en forma apropiada	1	2	3	4	5

Preguntas de cierre e impresiones finales

- Las respuestas son optativas
- ¿Cuál es su impresión general del grupo de funcionalidades hasta ahora probadas?
- ¿Qué es lo que más le gustó del módulo ?
- ¿Qué es lo que menos le gustó del módulo ?
- ¿Hay algo que esté faltando en el módulo
- ¿Tiene algún otro comentario o pregunta ?

ANEXO B: CASOS DE USO – DIAGRAMAS DE SECUENCIA - CONTRATOS

Caso de uso	RF-00 Login	
Objetivos asociados	Realizar el login de los usuarios	
Actores	ACT-01 Administrador, ACT-02 Socio, ACT-03 Colaborador	
Precondición	1. La cuenta de usuario debe estar creada.	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el usuario quiere acceder al sistema.
	2	El sistema solicita que se introduzca el nombre de usuario
	3	El usuario ingresa nombre de usuario
	4	Sistema solicita ingresar password
	5	Usuario ingresa password
	6	El sistema valida la clave introducida.
Postcondición		
Secuencias alternativas	Paso	Acción
	6.1	La clave es introducida erróneamente menos de tres veces consecutivas
	6.1.1	El sistema indica error
	6.1.2	Volver a paso 2
	6.2	La clave es introducida erróneamente por tercera vez
	6.2.1	El sistema indica error en validación de usuario
	6.2.2	Finalizar caso de uso
Frecuencia esperada	Muy alta	
Comentarios		

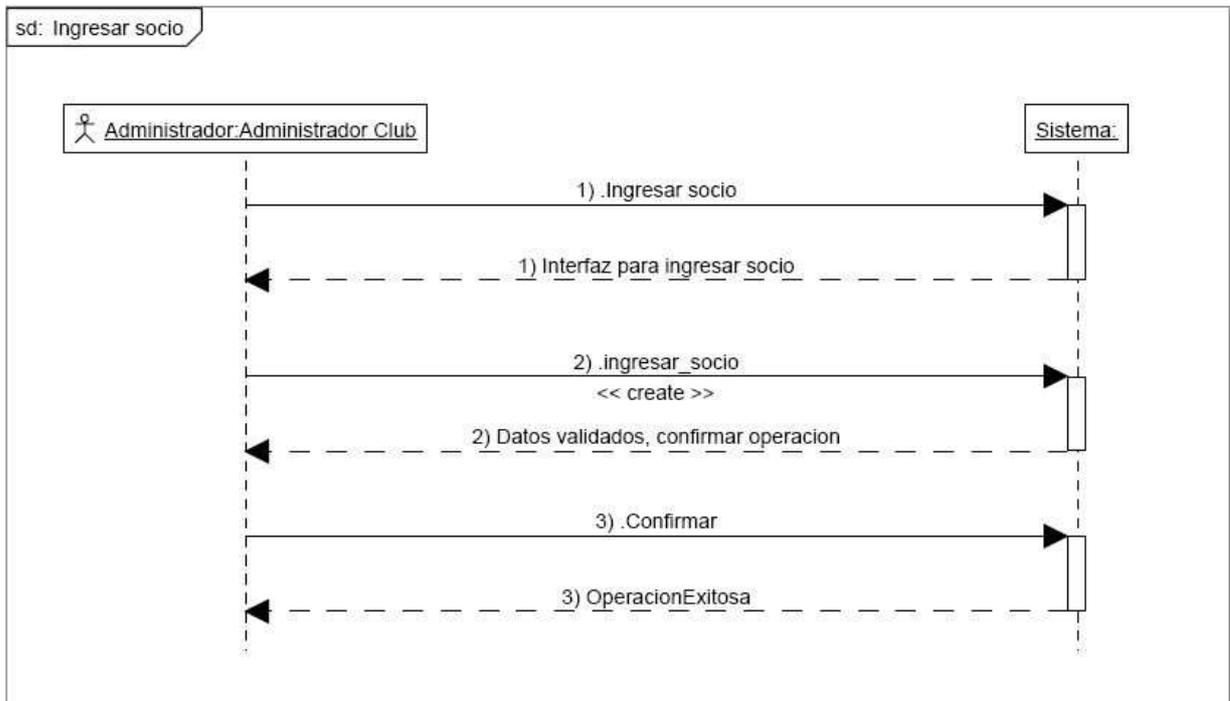


Operación	Login()
Precondición	Usuario tiene cuenta creada en el sistema
Poscondición	No hay

Operación	Login(usuario, passwd)
Precondición	.No hay
Poscondición	Usuario tiene acceso según privilegios

Operación	Confirmar()
Precondición	No hay
Poscondición	Datos de usuario validados

Caso de Uso	RF– 01 Ingresar nuevo socio	
Objetivos asociados	Registrar un nuevo socio en la base de datos del Club de tenis	
Actores asociados	ACT-01 Administrador club de tenis	
Precondición	1. El socio no se encuentra ingresada en el sistema. 2. Estar logueado e identificado	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el administrador del club desea registrar un nuevo socio.
	2	El administrador ingresa los datos de identificación del socio (Número de socio, RUT, nombres, apellidos, fecha de nacimiento, sexo, fotografía, fecha de ingreso como socio, dirección, teléfonos, E-mail, tipo de membresía).
	3	El sistema valida los datos ingresados para el nuevo socio
	4	El sistema registra el nuevo socio
	5	Finaliza el caso de uso
Postcondición		
Secuencias alternativas	Paso	Acción
	3.1	Datos ingresados son incorrectos
	3.1.1	El sistema indica error
	3.1.2	El administrador elige cancelar o seguir para arreglar campos no válidos.
	3.1.3	Si administrador elige seguir, la ejecución vuelve a paso 2
	3.1.4	Finaliza el caso de uso
Frecuencia esperada	Media	
Comentarios		



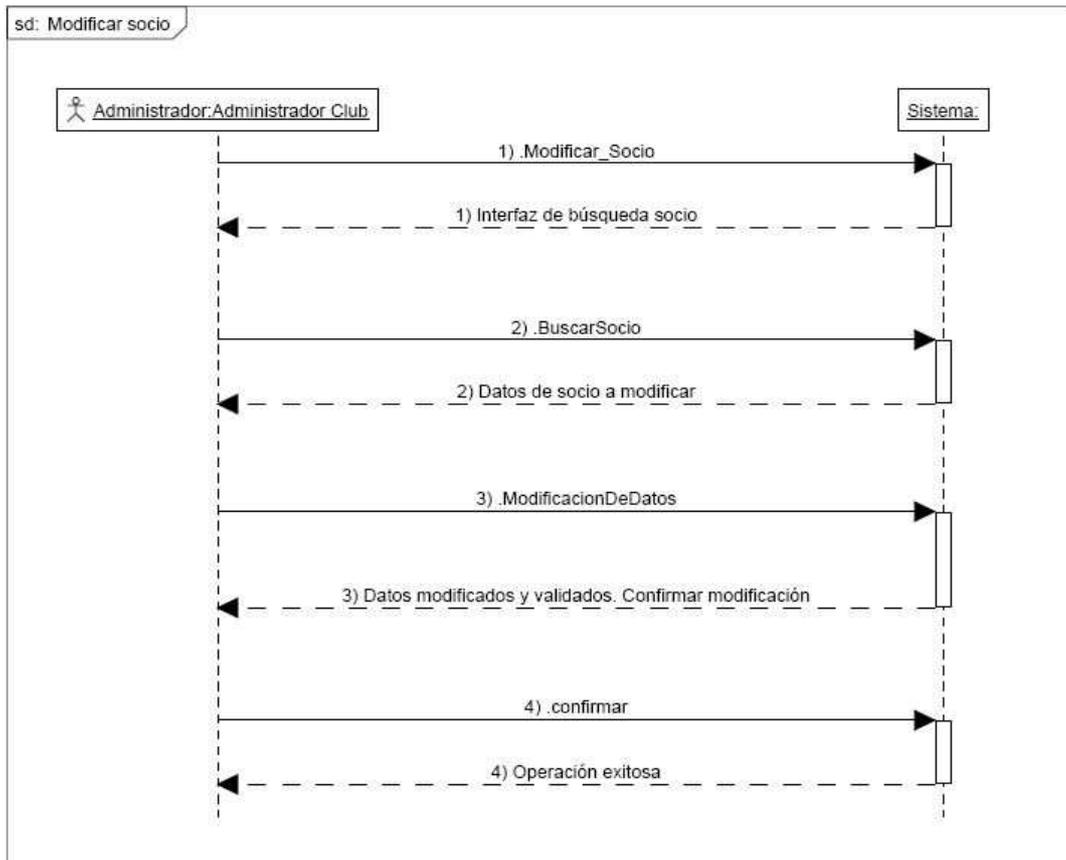
Operación	ingresar_socio()
Precondición	Administrador está registrado.
Poscondición	No hay.

Operación	ingresar(rut, nombres, apellidos, fecha de nacimiento, sexo, fotografía, fecha de ingreso como socio, dirección, teléfonos, E-mail, tipo de membresía, login)
Precondición	Login, tipo membresía están registrados en el sistema.
Poscondición	Datos de socio son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos de socio son ingresados en el sistema.

Caso de Uso	RF– 02 Modificar datos de socios	
Objetivos asociados	Cambiar datos personales del registro de un socio	
Actores asociados	ACT-01 Administrador club de tenis	
Precondición	1. El socio se encuentra ingresado en el sistema. 2. Estar logueado e identificado	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el administrador del desea modificar los datos del socio
	2	El administrador selecciona el registro de un socio
	3	El sistema presenta por pantalla los datos personales de un socio
	4	El administrador modifica los campos editables y/o datos deseados.
	5	El sistema valida los cambios realizados y pide confirmación al administrador para ejecutar las modificaciones
	6	El administrador confirma los cambios
	7	El sistema registra las modificaciones y despliega un mensaje por pantalla indicando el éxito de la operación.
	8	Finaliza el caso de uso
Postcondición	Se modifican los datos de un registro contenido en la base de datos.	
Secuencias alternativas	Paso	Acción
	1.1	La acción la puede ejecutar el mismo socio
	5.1	Los datos ingresados son incorrectos
	5.2	El sistema presenta mensaje de error en la validación de los datos.
	5.3	El control de ejecución vuelve al paso número 3

	6.1	El administrador cancela la modificación de los datos
	6.2	Finaliza el caso de uso
Frecuencia esperada	Media – Baja	
Comentarios		

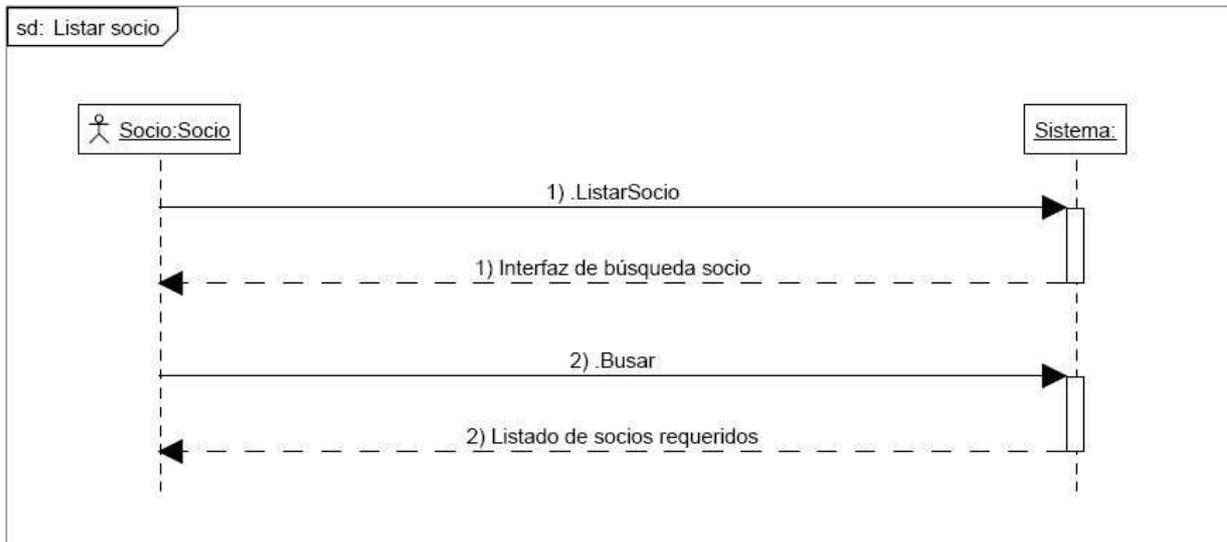


Operación	modificar_socio()
Precondición	Administrador está registrado.
Poscondición	No hay.

Operación	modificar(datos_modificados, login)
Precondición	Usuario ya está registrado en el sistema.
Poscondición	Datos de socio son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos de socio modificados e ingresados al sistema.

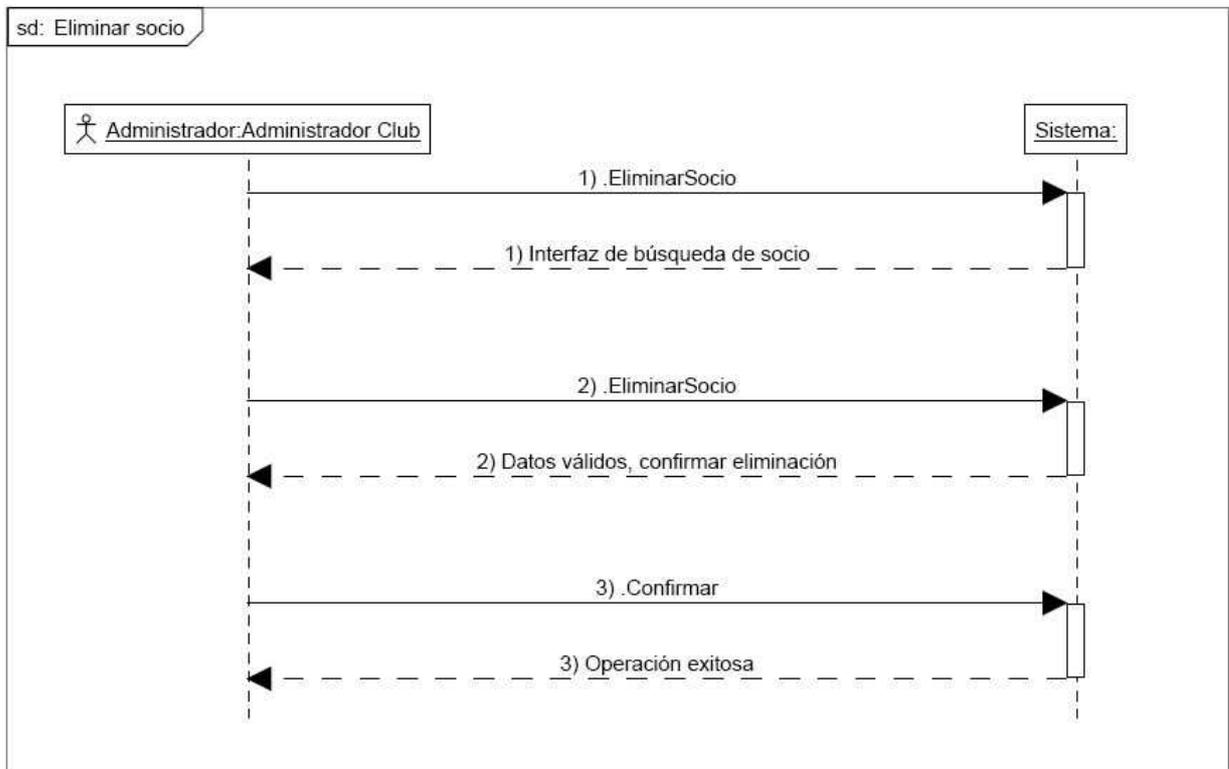
Caso de Uso	RF- 03 Listar datos de socios	
Objetivos asociados	Mostrar lista completa y ordenada de los socios del club desde donde se puede obtener la información parametrizada de ellos.	
Actores asociados	ACT-01 Administrador club de tenis ACT-02 Socio	
Precondición	Estar logueado e identificado	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el administrador desea listar los datos de socios.
	2	El sistema muestra en forma alfabética el listado de socios
	3	El administrador elige socio para ver sus datos personales
	4	El sistema despliega los datos personales del socio elegido
	5	Finaliza el caso de uso.
Postcondición	Obtención de información de contacto de uno o varios socios.	
Secuencias alternativas	Paso	Acción
	1.1	El caso de uso también puede ser iniciado por un usuario logueado en el sistema con rol de socio.
Frecuencia esperada	Media.	
Comentarios		



Operación	Listar_socios()
Precondición	Usuario está registrado.
Poscondición	No hay.

Operación	buscar(rut)
Precondición	No hay
Poscondición	No hay.

Caso de Uso	RF– 04 Eliminar socio	
Objetivos asociados	Eliminar un registro de un socio desde la base de datos	
Actores asociados	ACT-01 Administrador club de tenis	
Precondición	1. Socio debe existir en la base de datos 2. Estar logueado e identificado	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el administrador desea eliminar un socio.
	2	El administrador selecciona el registro de un socio
	3	El sistema presenta por pantalla los datos personales del socio seleccionado y pide confirmación para la eliminación del registro.
	4	El administrador confirma eliminación.
	5	El sistema procede a realizar la eliminación del registro desde la base de datos y muestra mensaje de eliminación del registro.
	6	Finaliza el caso de uso
Postcondición	Se elimina el socio; pero se guarda el registro en otra tabla histórica	
Secuencias alternativas	Paso	Acción
	4.1	El administrador cancela la eliminación
	4.2	Finaliza el caso de uso
Frecuencia esperada	Baja	
Comentarios		

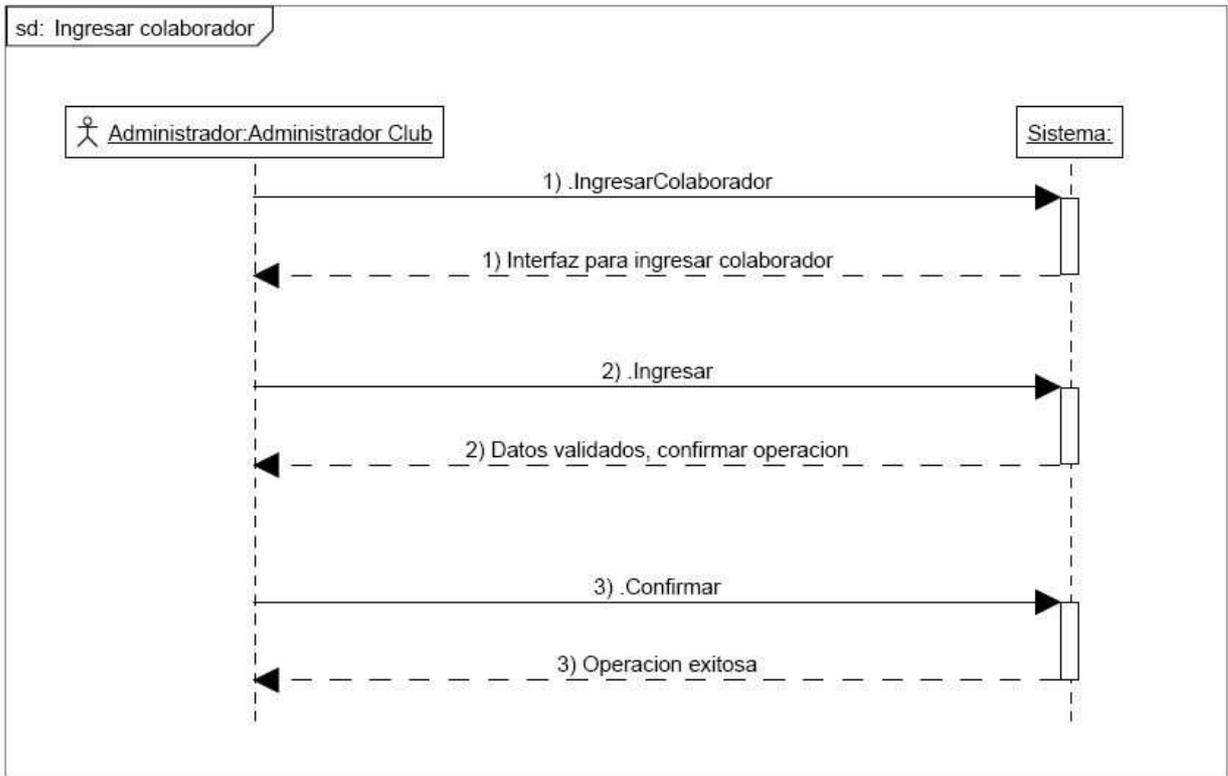


Operación	eliminar_socio()
Precondición	Administrador está registrado.
Poscondición	No hay.

Operación	eliminar(rut_socio)
Precondición	Socio está registrado en el sistema.
Poscondición	Datos ingresados son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos de socio son eliminados del sistema

Caso de Uso	RF-05 Ingresar nuevo colaborador	
Objetivos asociados	Registrar un nuevo colaborador en la base de datos del Club de tenis	
Actores asociados	ACT-01 Administrador club de tennis	
Precondición	1. El colaborador no se encuentra ingresado en el sistema. 2. Estar logueado e identificado	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el administrador del club desea registrar un nuevo socio.
	2	El administrador ingresa los datos de identificación del nuevo colaborador (Número de colaborador, RUT, Nombres y apellidos, Fecha de nacimiento, Sexo, Fotografía, Fecha de ingreso como colaborador, Dirección Teléfonos, E-mail, Login de colaborador).
	3	El sistema valida los datos ingresados para el nuevo colaborador
	4	El sistema registra el nuevo socio
	5	Finaliza el caso de uso
Postcondición		
Secuencias alternativas	Paso	Acción
	3.1	Datos ingresados son incorrectos
	3.1.1	El sistema indica error
	3.1.2	El administrador elige cancelar o seguir para arreglar campos no válidos.
	3.1.3	Si administrador elige seguir, ejecución vuelve a paso 2
	3.1.4	Finalizar el caso de uso
Frecuencia esperada	Media - Baja	
Comentarios		

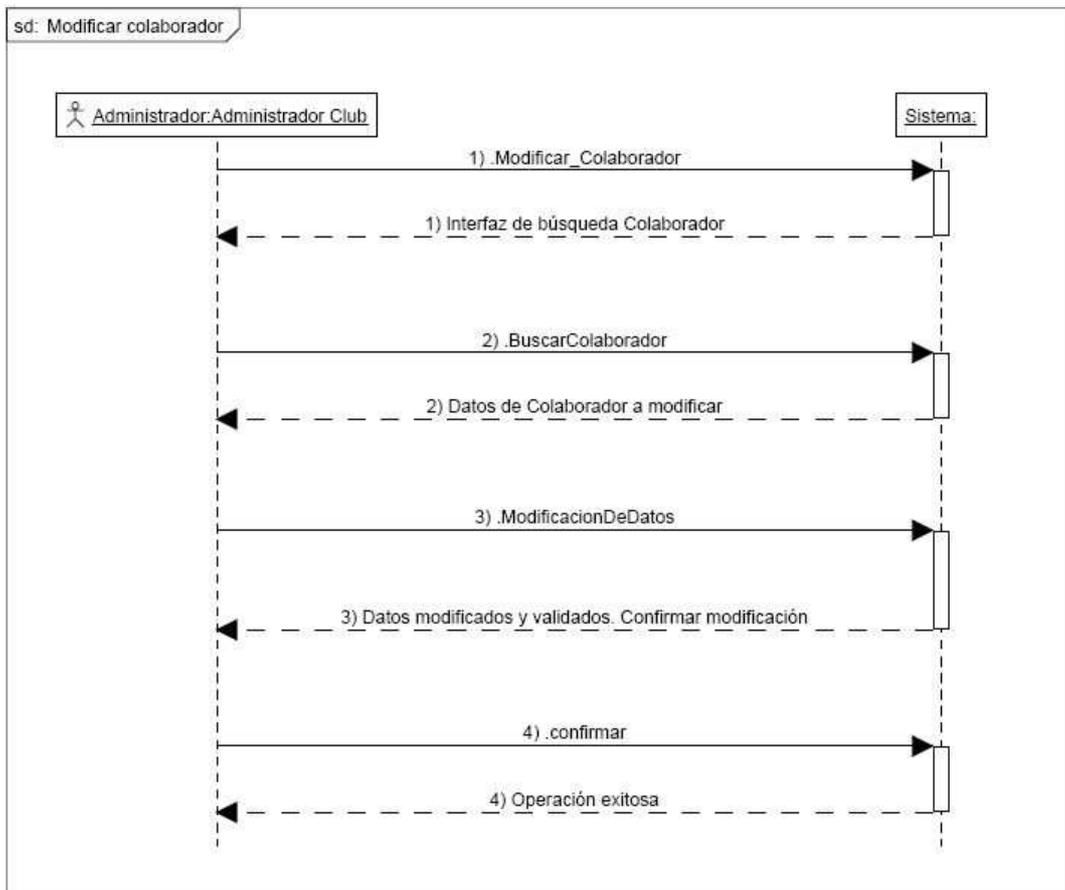


Operación	ingresar_colaborador()
Precondición	Administrador está registrado.
Poscondición	No hay.

Operación	ingresar(rut, nombres, apellidos, Fecha de nacimiento, Sexo, Fotografía, Fecha de ingreso como colaborador, Dirección, teléfonos, E-mail, Login)
Precondición	No hay
Poscondición	Datos de socio son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos de colaborador son ingresados en el sistema.

Caso de Uso	RF-06 Modificar datos colaborador	
Objetivos asociados	Cambiar datos personales del registro de un colaborador	
Actores asociados	ACT-01 Administrador club de tenis	
Precondición	1. Estar logueado e identificado	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el administrador desea modificar los datos de un colaborador.
	2	El administrador selecciona el registro de un colaborador
	3	El sistema presenta por pantalla los datos personales de un socio.
	4	El administrador modifica los campos editables y/o datos deseados
	5	El sistema valida los cambios realizados y pide confirmación al administrador para ejecutar las modificaciones.
	6	El administrador confirma los cambios.
	7	El sistema registra las modificaciones del registro en la BD
	8	Finaliza el caso de uso
Postcondición		
Secuencias alternativas	Paso	Acción
	1.1	La acción la puede ejecutar el mismo colaborador
	5.1	Los datos ingresados son incorrectos
	5.2	El sistema presenta mensaje de error en la validación de los datos
	5.3	El control de ejecución vuelve al paso número 3
	6.1	El administrador cancela la modificación de los datos
	6.2	Finaliza el caso de uso.
Frecuencia esperada	Baja	
Comentarios		

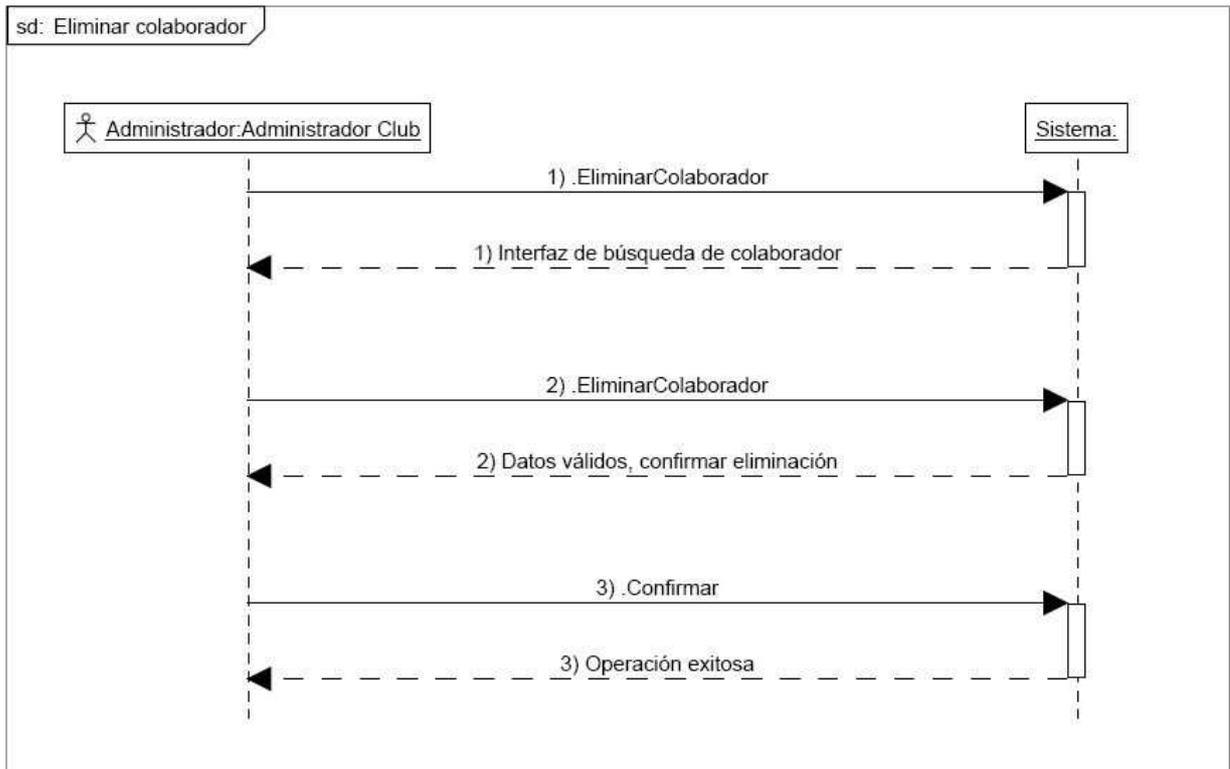


Operación	modificar_colaborador()
Precondición	Administrador está registrado.
Poscondición	No hay.

Operación	modificar(datos_modificados, login)
Precondición	Usuario ya está registrado en el sistema.
Poscondición	Datos de socio son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos modificados e ingresados al sistema.

Caso de Uso	RF-07 Eliminar colaborador	
Objetivos asociados	Eliminar registro de un colaborador	
Actores asociados	ACT-01 Administrador club de tenis	
Precondición	Estar logueado e identificado	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el administrador desea eliminar los datos del colaborador.
	2	El administrador selecciona el registro de un colaborador.
	3	El sistema presenta por pantalla los datos personales del colaborador seleccionado y pide confirmación para la eliminación del registro.
	4	El administrador confirma eliminación.
	5	El sistema procede a realizar la eliminación del registro desde la base de datos y muestra mensaje de eliminación del registro
	6	Finaliza el caso de uso
Postcondición	Se elimina el colaborador; pero se guarda el registro en otra tabla histórica. Las noticias que fueron creadas por el colaborador eliminado, serán traspasadas al usuario administrador.	
Secuencias alternativas	Paso	Acción
	4.1	El administrador cancela la eliminación.
	4.2	Finaliza el caso de uso
Frecuencia esperada	Baja	
Comentarios		

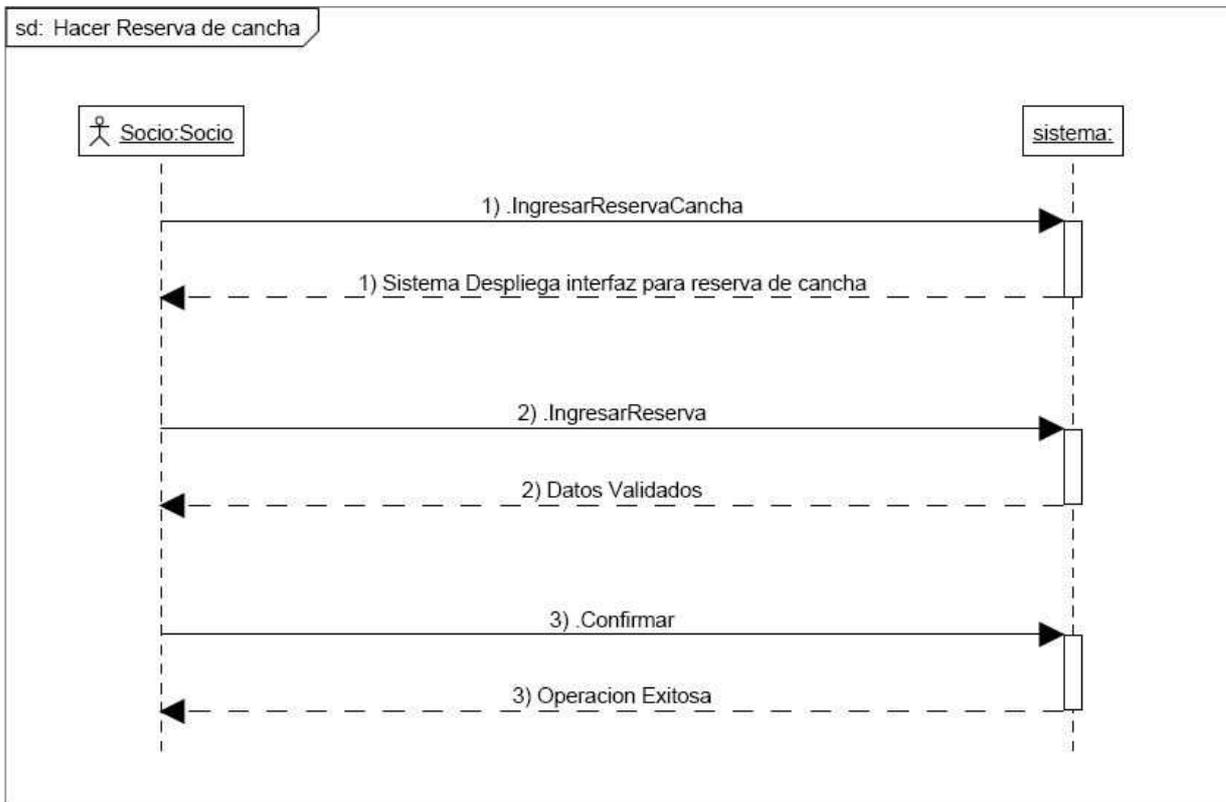


Operación	eliminar_colaborador()
Precondición	Administrador está registrado.
Poscondición	No hay.

Operación	eliminar(rut_ colaborador)
Precondición	Colaborador está registrado en el sistema.
Poscondición	Datos ingresados son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos de colaborador son eliminados del sistema

Caso de Uso	RF-08 Hacer reserva de cancha	
Objetivos asociados	Realizar una reserva de cancha de tenis	
Actores	ACT-02 Socio	
Precondición	Socio se ha identificado correctamente a través de login y password	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el socio desea realizar una reserva de cancha de tenis.
	2	El sistema muestra información de las canchas disponibles para reservar
	3	El socio ingresa Fecha, Hora y número de cancha a reservar
	4	Sistema devuelve disponibilidad para la reserva
	5	Socio confirma reserva
	6	Sistema registra la nueva reserva de cancha
	7	Finaliza el caso de uso
Postcondición		
Secuencias alternativas	Paso	Acción
	3.1	Se ha ingresado fecha, hora o cancha no disponible
	3.1.1	Sistema avisa al usuario del error en el ingreso de reserva
	3.1.2	Finalizar caso de uso
	5.1	El socio no confirma la reserva (cancelar)
	5.1.1	Finalizar caso de uso
Frecuencia esperada	Media - Alta	
Comentarios		



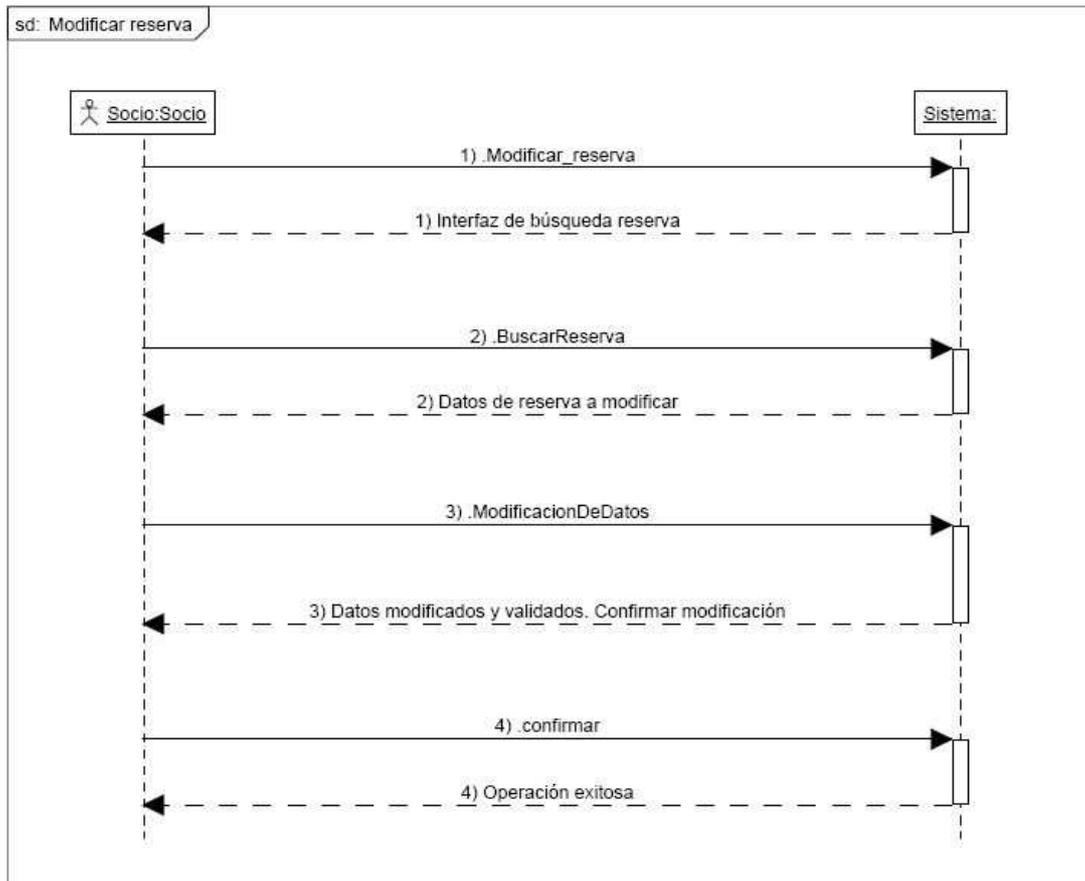
Operación	ingresar_reserva()
Precondición	Socio está registrado y logueado.
Poscondición	No hay.

Operación	ingresar_reserva(Fecha_reserva, Hora_reserva, rut socio, número_cancha)
Precondición	Cancha no debe estar reservada para el mismo día y hora
Poscondición	Datos de la reserva son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos de la reserva son ingresados en el sistema.

Caso de Uso	RF-09 Modificar reserva de cancha	
Objetivos asociados	Actualizar los datos de una reserva de cancha, manteniendo la integridad y la unicidad de los datos.	
Actores asociados	ACT-01 Administrador club de tenis ACT-02 Socio	
Precondición	La reserva de cancha ya se encuentra ingresada en el sistema. 2. El actor se encuentra identificado y autenticado en el sistema.	
Secuencia Normal	Paso	Acción
	1	El caso de uso comienza cuando el actor ACT-02 solicita al sistema modificar los datos de una reserva de cancha.
	2	El sistema despliega una interfaz con los campos de búsqueda.
	3	El usuario socio elige una reserva efectuada por el, la cual procederá a editar
	4	El sistema valida los datos, y despliega una interfaz con los campos modificables.
	5	El usuario socio ingresa los datos que desea modificar y presiona modificar.
	6	El sistema despliega la información ingresada, y solicita confirmación de los datos.
	7	El administrador confirma los datos, presiona confirmar.
	8	El sistema confirma el éxito de la operación
	9	Fin del caso de uso.
Postcondición	El sistema actualiza los datos de una reserva de cancha	
Secuencias alternativas	Paso	Acción
	4.1	El sistema no valida los datos ingresados por el usuario socio. El control vuelve al paso 3.
	5.1	El sistema no valida los datos ingresados. El control vuelve al paso 5.

	5.2	El usuario socio no confirma los datos ingresados.
	5.3	Fin del caso de uso
Frecuencia esperada	Media - Alta	
Comentarios		

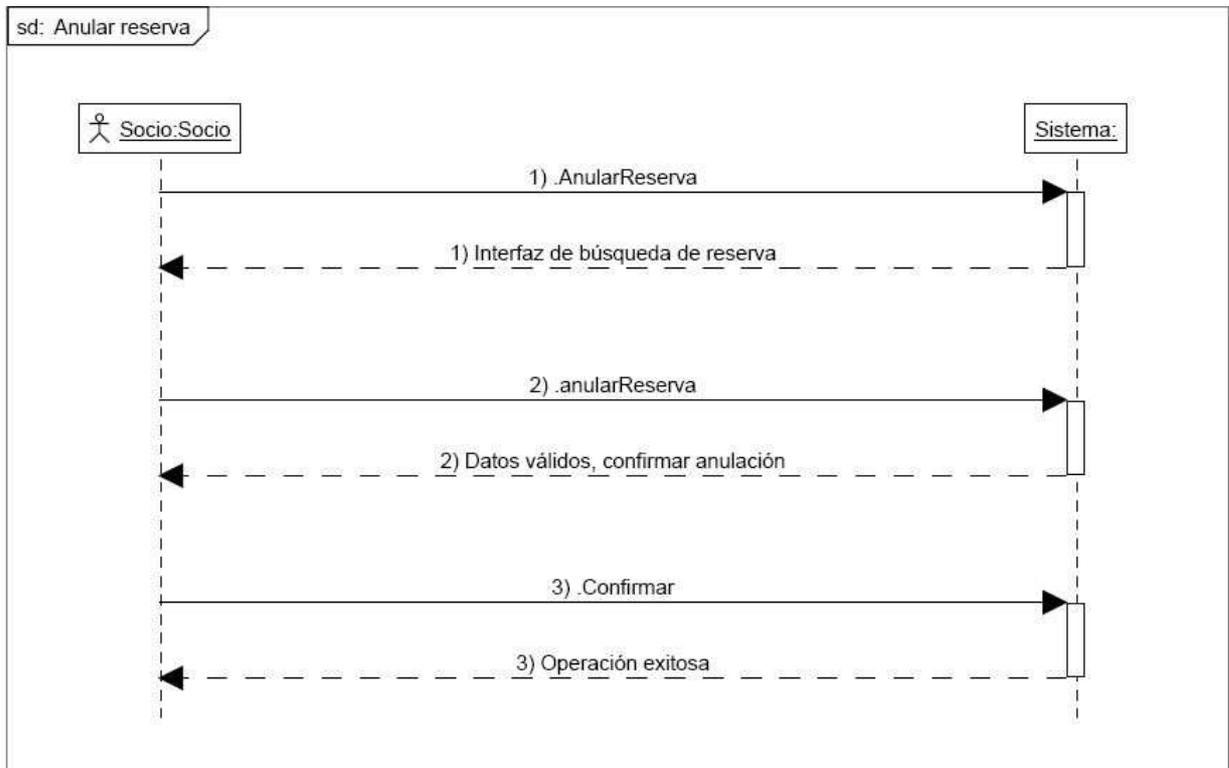


Operación	modificar_reserva()
Precondición	Administrador está registrado.
Poscondición	No hay.

Operación	modificar(datos_modificados, login)
Precondición	Usuario ya está registrado en el sistema.
Poscondición	Datos de socio son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos son modificados e ingresados en el sistema.

Caso de Uso	RF-10 Anular reserva de cancha	
Objetivos asociados	Realizar la anulación de una reserva hecha con anterioridad	
Actores	ACT-02 Socio, ACT-01 Administrador	
Precondición	1. La reserva de cancha ya se encuentra ingresada en el sistema. 2. El actor se encuentra identificado y autenticado en el sistema.	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el socio desea anular una reserva hecha con anterioridad.
	2	El socio selecciona reserva a anular
	3	El sistema pide confirmación para anulación
	4	El socio confirma anulación de reserva
	5	El sistema anula la reserva de cancha de tenis
	6	Fin del caso de uso
Postcondición		
Secuencias alternativas	Paso	Acción
	1	El caso de uso también puede ser iniciado por el administrador del club
	3.1	El socio cancela anulación de reserva
	3.1.1	Fin del caso de uso
	5.1	El sistema no anula la reserva, debido a que el plazo para hacerlo ha expirado.
	5.1.1	Fin del caso de uso
Frecuencia esperada	Baja	
Comentarios		

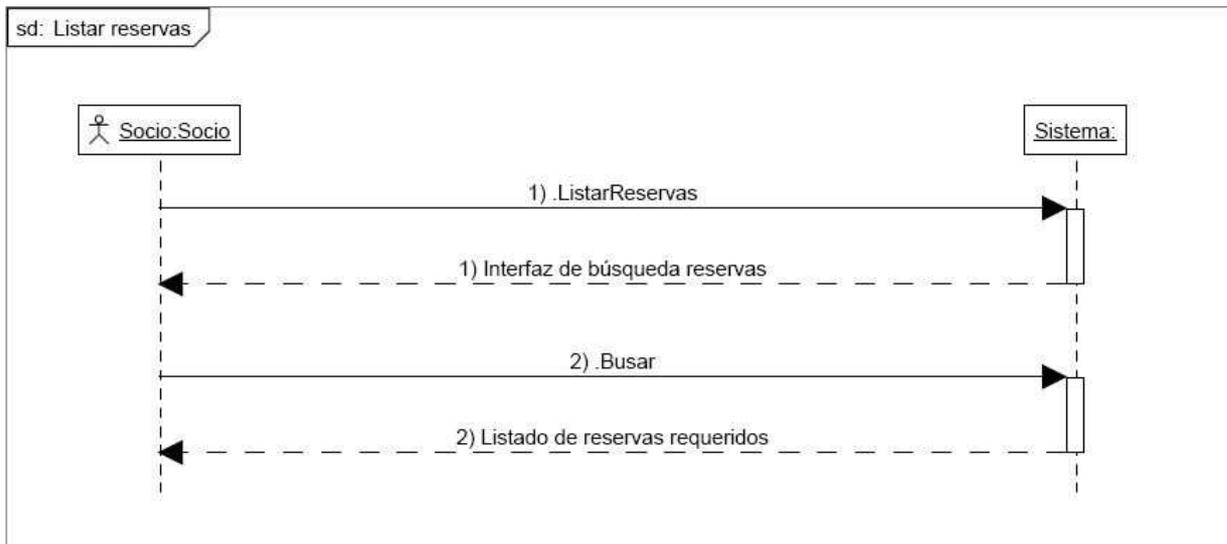


Operación	Anular_reserva()
Precondición	Socio o administrador está registrado.
Poscondición	No hay.

Operación	anular_reserva (id_reserva)
Precondición	Administrador está registrado en el sistema.
Poscondición	Datos ingresados son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos de reserva son eliminados del sistema

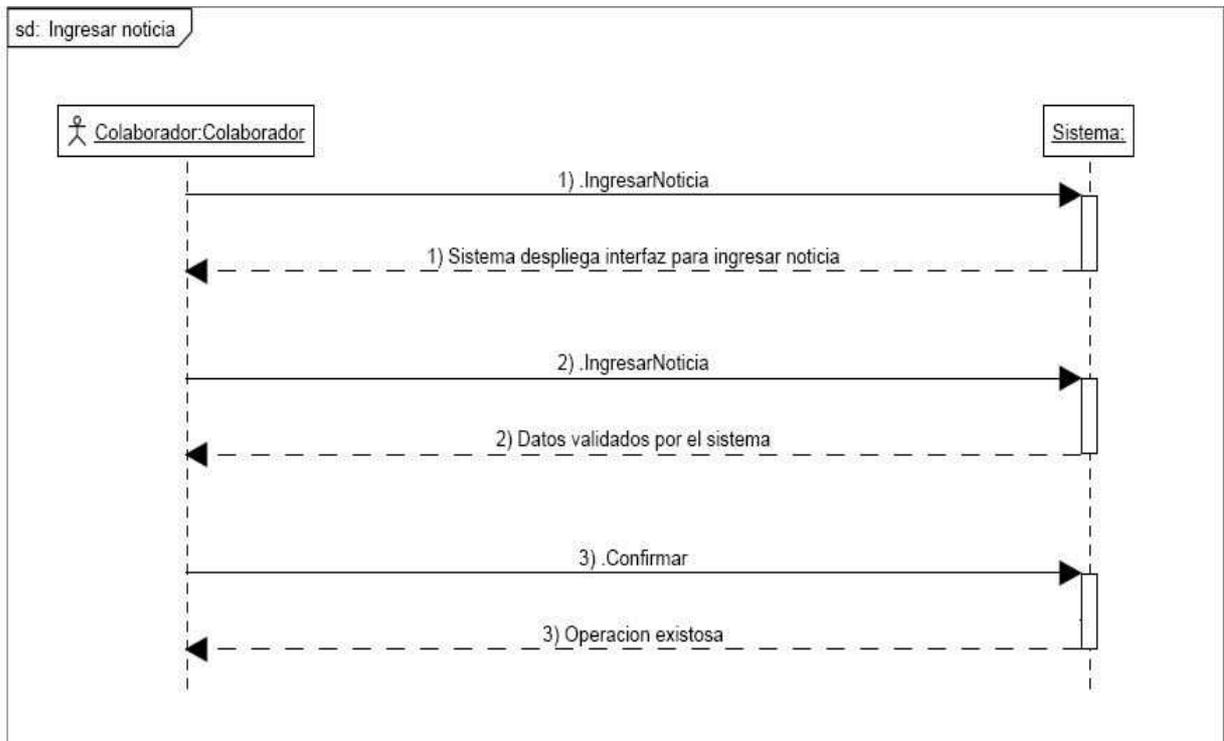
Caso de Uso	RF-11 Listar reserva de canchas	
Objetivos asociados	Mostrar lista completa y ordenada de las reservas de canchas. Esta lista es paramétrica	
Actores asociados	ACT-01 Administrador club de tenis ACT-02 Socio	
Precondición	Estar logueado e identificado	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el administrador desea listar las reservas de cancha.
	2	El sistema muestra en forma paramétrica el listado de reservas de canchas
	3	El administrador elige un parámetro para listar las reservas efectuadas (por fechas, por cancha, por socio)
	4	El sistema despliega los datos elegidos
	5	Finaliza el caso de uso
Postcondición	Lista detallada de las reservas según parámetros elegidos. Listado por pantalla o por salida de impresora.	
Secuencias alternativas	Paso	Acción
	1.1	El caso de uso también puede ser iniciado por un usuario logueado en el sistema con rol de socio; pero sólo de aquellas reservas que el mismo a hecho.
	1.2	Sigue en el paso 2.
Frecuencia esperada	Media - Alta	
Comentarios		



Operación	Listar_reservas()
Precondición	Usuario está registrado.
Poscondición	No hay.

Operación	buscar(id_cancha, fecha, rut_socio)
Precondición	No hay
Poscondición	No hay.

Caso de Uso	RF-12 Ingresar noticia	
Objetivos asociados	Registrar una nueva noticia en el sitio Web	
Actores asociados	ACT-03 Colaborador	
Precondición	1. Estar logueado e identificado	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el usuario colaborador desea ingresar una nueva noticia
	2	El sistema despliega una interfaz donde se incluyen los campos necesarios para ingresar los contenidos de la noticia.
	3	El usuario colaborador ingresa los contenidos de la noticia
	4	El sistema valida los datos ingresados para la nueva noticia
	5	El sistema registra la nueva noticia
	6	Finaliza el caso de uso
Postcondición		
Secuencias alternativas	Paso	Acción
	3.1	Datos ingresados son incorrectos
	3.1.1	El sistema indica error
	3.1.2	El administrador elige cancelar o seguir para arreglar campos no válidos.
	3.1.3	Si colaborador elige seguir, ejecución vuelve a paso 2
	3.1.4	Finalizar el caso de uso
Frecuencia esperada	Alta	
Comentarios		



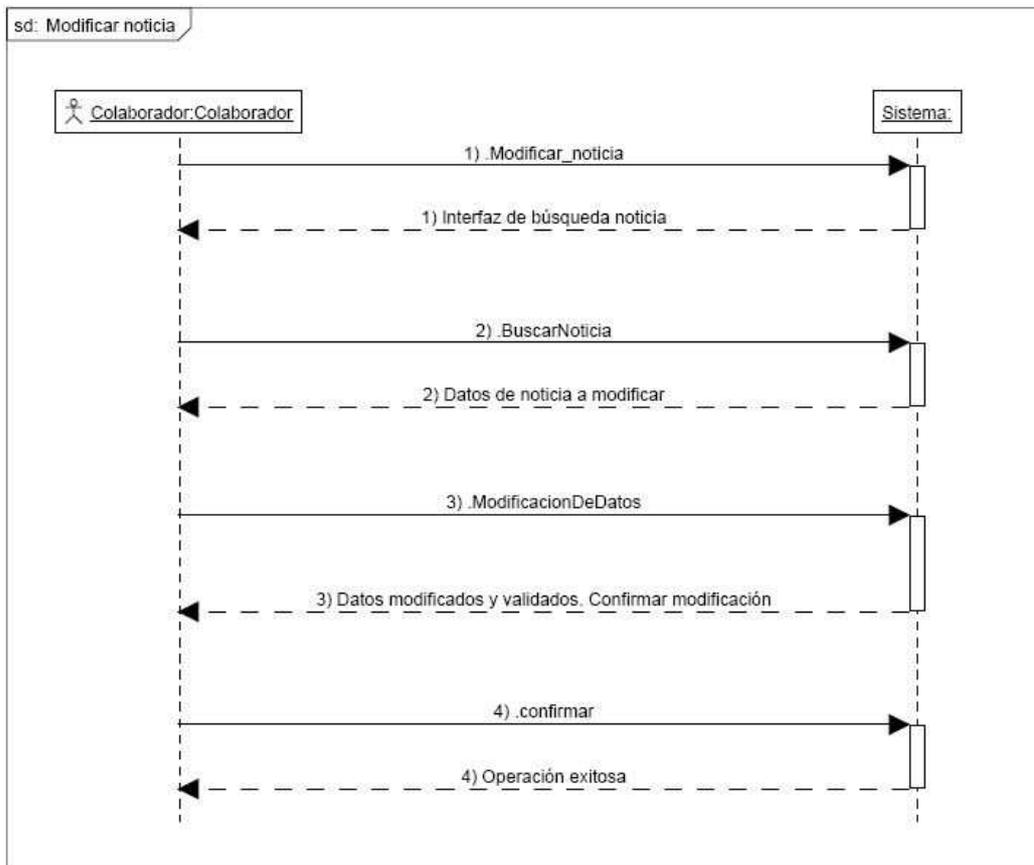
Operación	ingresar_noticia()
Precondición	Colaborador está registrado y logueado.
Poscondición	No hay.

Operación	ingresar_noticia(Fecha_publicación_noticia, Hora_publicación, autor_noticia, Contenido_noticia)
Precondición	No hay
Poscondición	Datos de la noticia son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos de la noticia son ingresados en el sistema.

Caso de Uso	RF-13 Modificar noticia	
Objetivos asociados	Actualizar los datos de una noticia, manteniendo la integridad y la unicidad de los datos.	
Actores asociados	ACT-03 Colaborador	
Precondición	La noticia ya se encuentra ingresada en el sistema. 2. El usuario colaborador se encuentra identificado y autenticado en el sistema.	
Secuencia Normal	Paso	Acción
	1	El caso de uso comienza cuando el usuario colaborador solicita al sistema modificar los datos de una noticia.
	2	El sistema despliega una interfaz con los campos de búsqueda.
	3	El usuario elige una noticia para editar
	4	El sistema despliega la información de la noticia seleccionada, la cual es modificada por el usuario colaborador
	5	El sistema despliega la información ingresada, y solicita confirmación de los datos.
	6	El usuario colaborador confirma los datos, presiona confirmar.
	7	El sistema informa el éxito de la operación
	8	Fin del caso de uso
Postcondición	Se graban las modificaciones en el registro de la base de datos y se crea un registro en una tabla de auditorias.	
Secuencias alternativas	Paso	Acción
	4.1	El sistema no valida los datos ingresados por el usuario colaborador. El control de ejecución vuelve al paso 3.
	6.1	El sistema no valida los datos ingresados. El colaborador vuelve al paso 5.

	7.1	El colaborador no confirma los datos ingresados.
	7.2	Fin del caso de uso.
Frecuencia esperada	Media	
Comentarios		

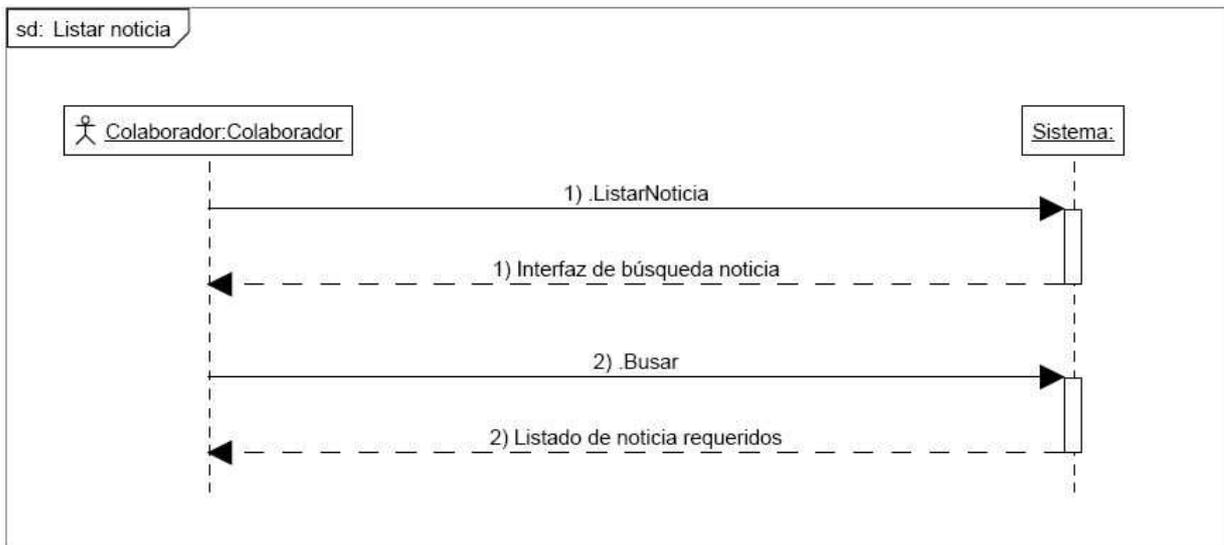


Operación	modificar_noticia()
Precondición	Colaborador está registrado.
Poscondición	No hay.

Operación	modificar(datos_modificados, login)
Precondición	Usuario ya está registrado en el sistema.
Poscondición	Datos de socio son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos son modificados e ingresados al sistema.

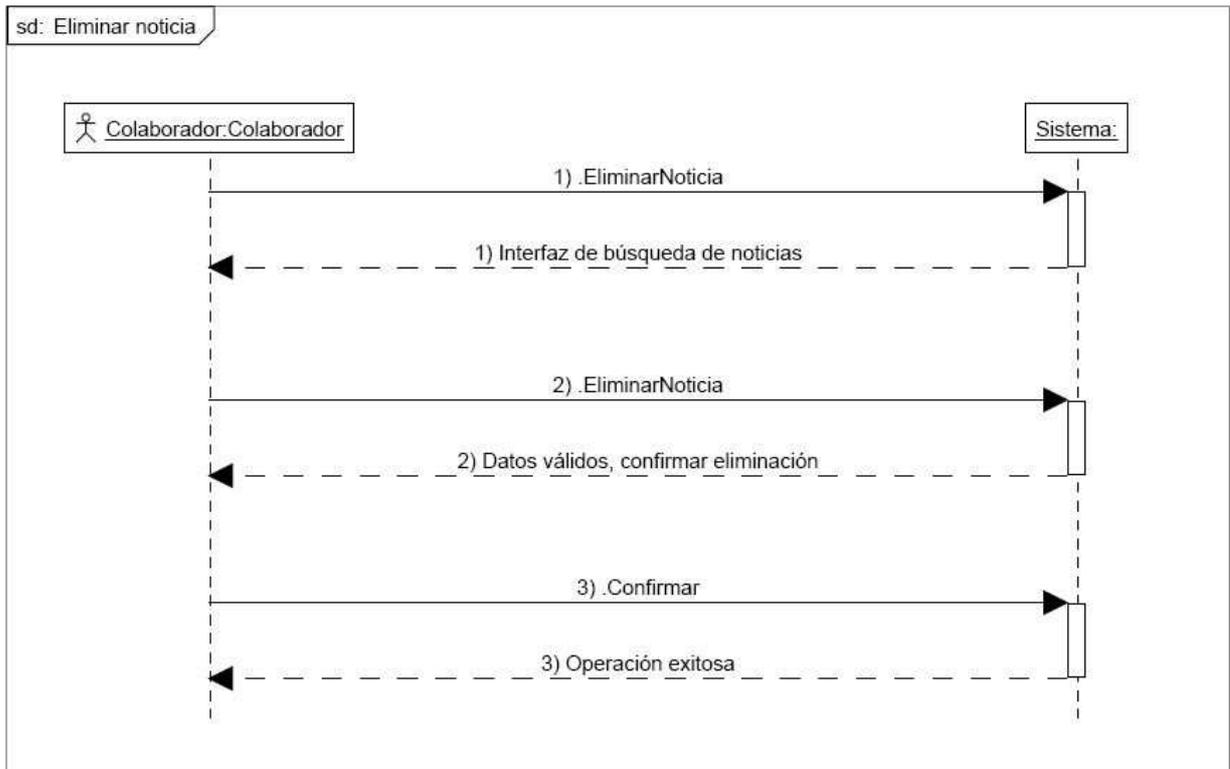
Caso de Uso	RF-14 Listar noticia	
Objetivos asociados	Mostrar lista completa y ordenada de las noticias publicadas en el sitio Web.	
Actores asociados	ACT-02	
Precondición	Estar logueado e identificado	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el usuario colaborador desea listar las noticias.
	2	El sistema muestra en forma cronológica el listado de las noticias.
	3	El usuario colaborador elige una noticia para ver sus contenidos.
	4	El sistema despliega los datos de la noticia seleccionada
	5	Finaliza el caso de uso.
Postcondición	Obtención de información de las noticias	
Secuencias alternativas	Paso	Acción
	3.1	El usuario colaborador elige cancelar el listado de noticias
	3.2	Fin del caso de uso.
Frecuencia esperada	Media	
Comentarios		



Operación	Listar_noticias()
Precondición	Usuario está registrado.
Poscondición	No hay.

Operación	buscar(id_noticia)
Precondición	No hay
Poscondición	No hay.

Caso de Uso	RF-15 Eliminar noticia	
Objetivos asociados	Eliminar un registro de una noticia desde la base de datos	
Actores asociados	ACT-03 Colaborador	
Precondición	<ol style="list-style-type: none"> 1. La noticia debe existir en la base de datos 2. Estar logueado e identificado 3. Tener los permisos para eliminación de noticias. 	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el colaborador desea eliminar una noticia.
	2	El colaborador selecciona el registro de una noticia.
	3	El sistema presenta por pantalla los datos de la noticia seleccionada y pide confirmación para la eliminación del registro.
	4	El colaborador confirma eliminación.
	5	El sistema procede a realizar la eliminación del registro desde la base de datos y muestra mensaje de eliminación del registro.
	6	Finaliza el caso de uso
Postcondición	Se elimina la membresía de la tabla; pero se guarda el registro en otra tabla histórica.	
Secuencias alternativas	Paso	Acción
	4.1	El colaborador cancela la eliminación
	4.2	Finaliza el caso de uso
Frecuencia esperada	Baja	
Comentarios		

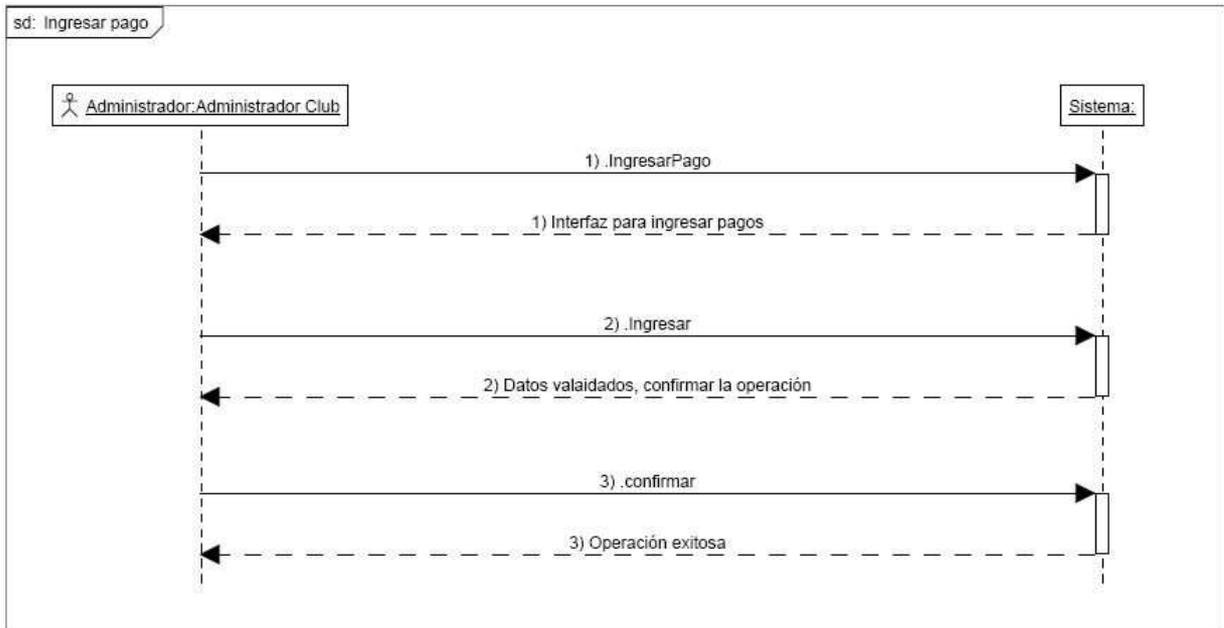


Operación	eliminar_noticia()
Precondición	Colaborador o administrador está registrado.
Poscondición	No hay.

Operación	eliminar_noticia(id_noticia)
Precondición	Colaborador está registrado en el sistema.
Poscondición	Datos ingresados son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos de noticia son eliminados del sistema

Caso de Uso	RF-16 Ingresar pago	
Objetivos asociados	Registrar un nuevo pago de mensualidad de un socio en la base de datos del Club de tenis	
Actores asociados	ACT-01 Administrador club de tenis	
Precondición	1. El pago no se encuentra ingresado en el sistema. 2. Estar logueado e identificado	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el administrador del club desea registrar un nuevo pago de mensualidad de un socio en particular.
	2	El administrador ingresa los datos para el nuevo pago: Nombre socio que paga, Fecha pago, Mes pagado, Número comprobante, Monto del pago, Forma de pago (Efectivo, Cheque, Traspaso electrónico)
	3	El sistema valida los datos ingresados para el nuevo socio
	4	El sistema registra el nuevo socio
	5	Finaliza el caso de uso
Postcondición		
Secuencias alternativas	Paso	Acción
	3.1	Datos ingresados son incorrectos
	3.1.1	El sistema indica error
	3.1.2	El administrador elige cancelar o seguir para arreglar campos no válidos.
	3.1.3	Si administrador elige seguir, la ejecución vuelve a paso 2
	3.1.4	Finaliza el caso de uso
Frecuencia esperada	Alta	
Comentarios		



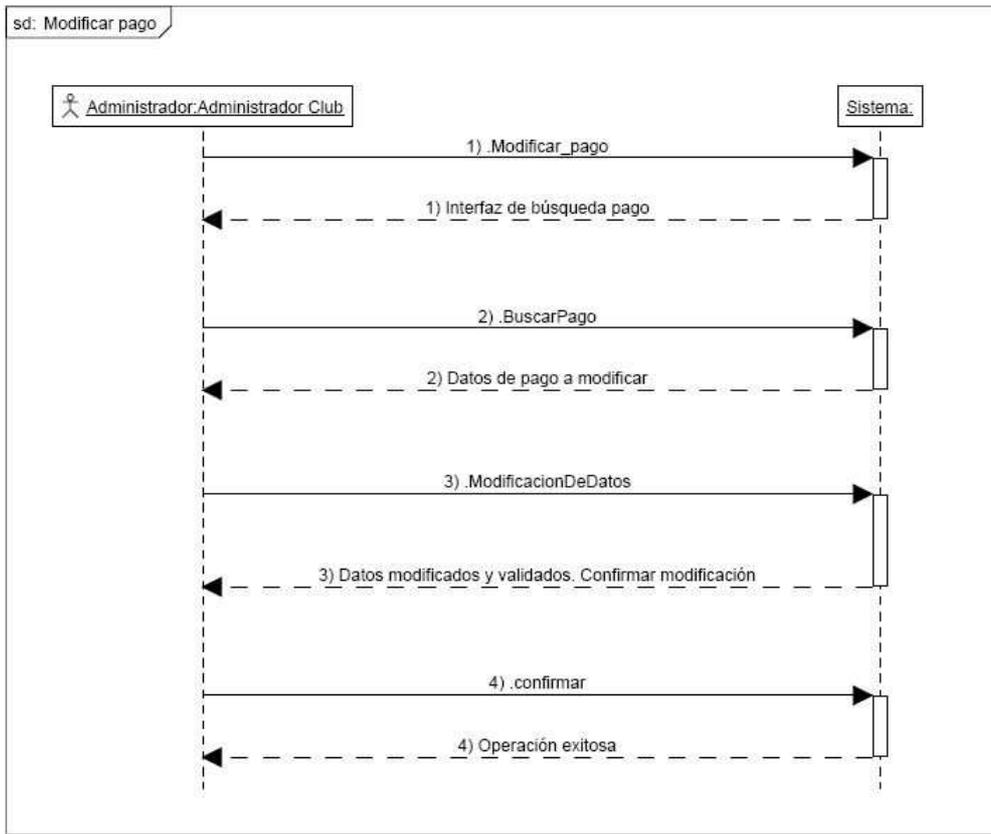
Operación	ingresar_pago(Nombre_socio, Fecha_pago, Mes_pagado, Número_comprobante, Monto_pago, formas_pago)
Precondición	Administrador está registrado y logueado.
Poscondición	No hay.

Operación	ingresar_pago(Fecha_publicación_noticia, Hora_publicación, autor_noticia, Contenido_noticia)
Precondición	No hay
Poscondición	Datos de la noticia son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos del pago de mensualidad son ingresados en el sistema.

Caso de Uso	RF-17 Modificar pago	
Objetivos asociados	Actualizar los datos de un pago de mensualidad, manteniendo la integridad y la unicidad de los datos.	
Actores asociados	ACT-01 Administrador club de tenis	
Precondición	El pago de la mensualidad ya se encuentra ingresada en el sistema. 2. El actor se encuentra identificado y autenticado en el sistema.	
Secuencia Normal	Paso	Acción
	1	El caso de uso comienza cuando el usuario administrador solicita al sistema modificar los datos de pago de mensualidad.
	2	El sistema despliega una interfaz con los campos de búsqueda.
	3	El usuario elige un pago de mensualidad para editar
	4	El sistema despliega la información del pago de la mensualidad seleccionada, la cual es modificada por el usuario administrador.
	5	El sistema despliega la información ingresada, y solicita confirmación de los datos.
	6	El usuario administrador confirma los datos, presiona confirmar.
	7	El sistema informa el éxito de la operación
	8	Fin del caso de uso
Postcondición	Se graban las modificaciones en el registro de la base de datos y se crea un registro en una tabla de auditorias.	
Secuencias alternativas	Paso	Acción
	4.1	El sistema no valida los datos ingresados por el usuario administrador. El control de ejecución vuelve al paso 3.
	6.1	El sistema no valida los datos ingresados. El administrador vuelve al paso 5.

	7.1	El administrador no confirma los datos ingresados.
	7.2	Fin del caso de uso.
Frecuencia esperada	Baja	
Comentarios		

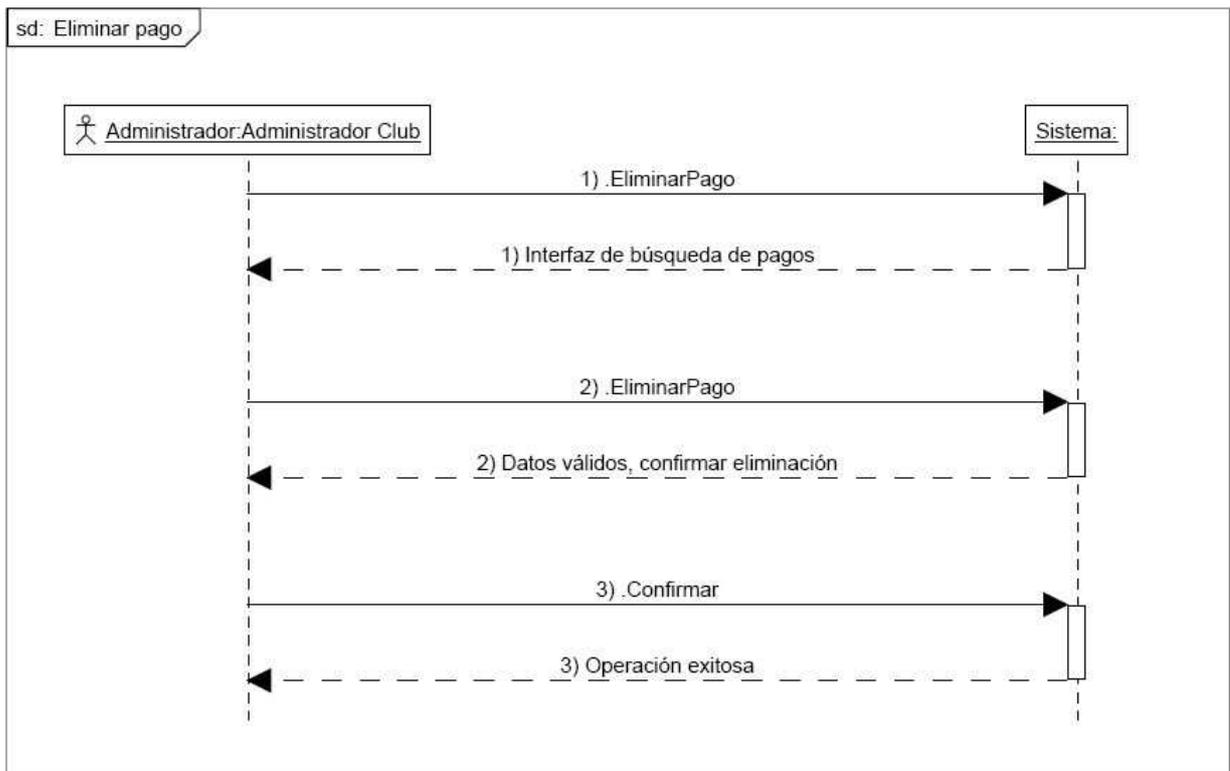


Operación	modificar_pago()
Precondición	Administrador está registrado.
Poscondición	No hay.

Operación	modificar(datos_modificados, login)
Precondición	Usuario ya está registrado en el sistema.
Poscondición	Datos de socio son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos del pago de mensualidad son modificados e ingresados en el sistema.

Caso de Uso	RF-18 Eliminar pago	
Objetivos asociados	Eliminar un registro de un pago desde la base de datos	
Actores asociados	ACT-01 Administrador club de tenis	
Precondición	1. El pago de mensualidad debe existir en la base de datos 2. Estar logueado e identificado	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el administrador desea eliminar un pago.
	2	El administrador selecciona el registro de un pago.
	3	El sistema presenta por pantalla los datos del pago seleccionado y pide confirmación para la eliminación del registro.
	4	El administrador confirma eliminación.
	5	El sistema procede a realizar la eliminación del registro desde la base de datos y muestra mensaje de eliminación del registro.
	6	Finaliza el caso de uso
Postcondición	Se elimina el pago de mensualidad de la tabla; pero se guarda el registro de la eliminación en 1 atabla de auditoria.	
Secuencias alternativas	Paso	Acción
	4.1	El administrador cancela la eliminación
	4.2	Finaliza el caso de uso
Rendimiento	Paso	Cota de tiempo
Frecuencia esperada	Baja	
Comentario		

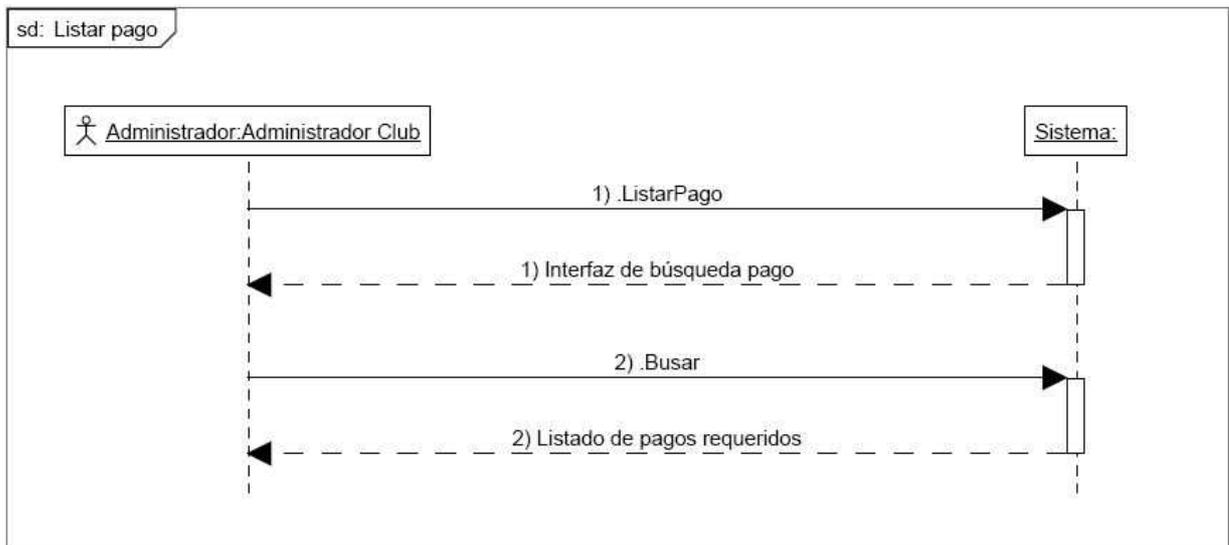


Operación	eliminar_pago()
Precondición	Administrador está registrado.
Poscondición	No hay.

Operación	eliminar_pago(id_pago)
Precondición	Administrador está registrado en el sistema.
Poscondición	Datos ingresados son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos del pago de mensualidad son eliminados del sistema.

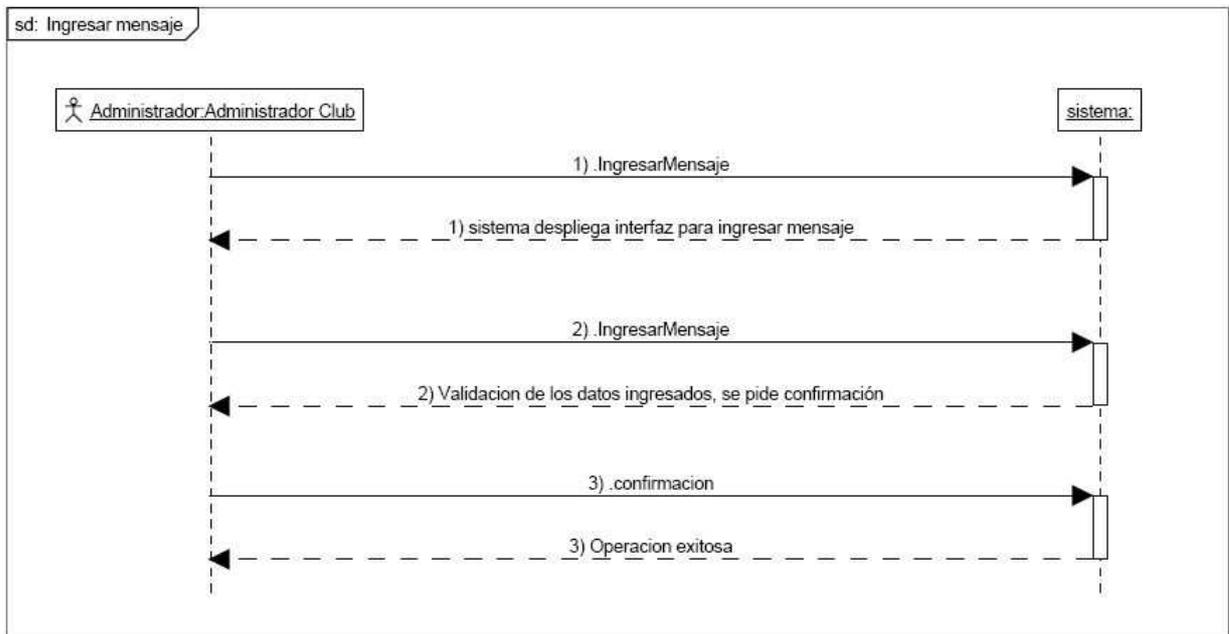
Caso de Uso	RF-19 Listar pago	
Objetivos asociados	Mostrar lista completa, ordenada y parametrizada de los pagos de mensualidades que los socios han hecho al club.	
Requisitos asociados		
Actores asociados	ACT-01 Administrador club de tenis ACT-02 Socio	
Precondición	1. Estar logueado e identificado	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el actor desea listar pagos de acuerdo a parámetros establecidos. (Por fecha, por socio, por año).
	2	El sistema pide ingreso de los parámetros de búsqueda.
	3	El actor elige un parámetro de búsqueda
	4	El sistema despliega los datos elegidos según los parámetros.
	5	Finaliza el caso de uso
Postcondición		
Secuencias alternativas	Paso	Acción
	4.1	El sistema deniega la entrega del listado, debido a que el actor no posee los privilegios para ver esta información. Se muestra un mensaje en pantalla informando de la negación de entrega de la información
	4.2	El control de ejecución vuelve al paso 2
	4.3	Finaliza el caso de uso
Frecuencia esperada	Alta	
Comentarios		



Operación	Listar_pago()
Precondición	Usuario está registrado.
Poscondición	No hay.

Operación	buscar(id_pago, fecha_pago, rut_socio)
Precondición	No hay
Poscondición	No hay.

Caso de Uso	RF-20 Configurar mensaje	
Objetivos asociados	Registrar un nuevo mensaje en la base de datos del Club de tenis	
Actores asociados	ACT-01 Administrador club de tenis	
Precondición	1. El mensaje no se encuentra ingresado en el sistema. 2. Estar logueado e identificado	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el administrador del club desea registrar un nuevo mensaje.
	2	El administrador ingresa los datos del mensaje: Fecha de creación, Hora de creación, Contenido mensaje.
	3	El sistema valida los datos ingresados para el nuevo mensaje.
	4	El sistema registra el nuevo mensaje.
	5	Finaliza el caso de uso
Postcondición		
Secuencias alternativas	Paso	Acción
	3.1	Datos ingresados son incorrectos
	3.1.1	El sistema indica error
	3.1.2	El administrador elige cancelar o seguir para arreglar campos no válidos.
	3.1.3	Si administrador elige seguir, la ejecución vuelve a paso 2
	3.1.4	Finalizar el caso de uso
Frecuencia esperada	Media	
Comentarios		

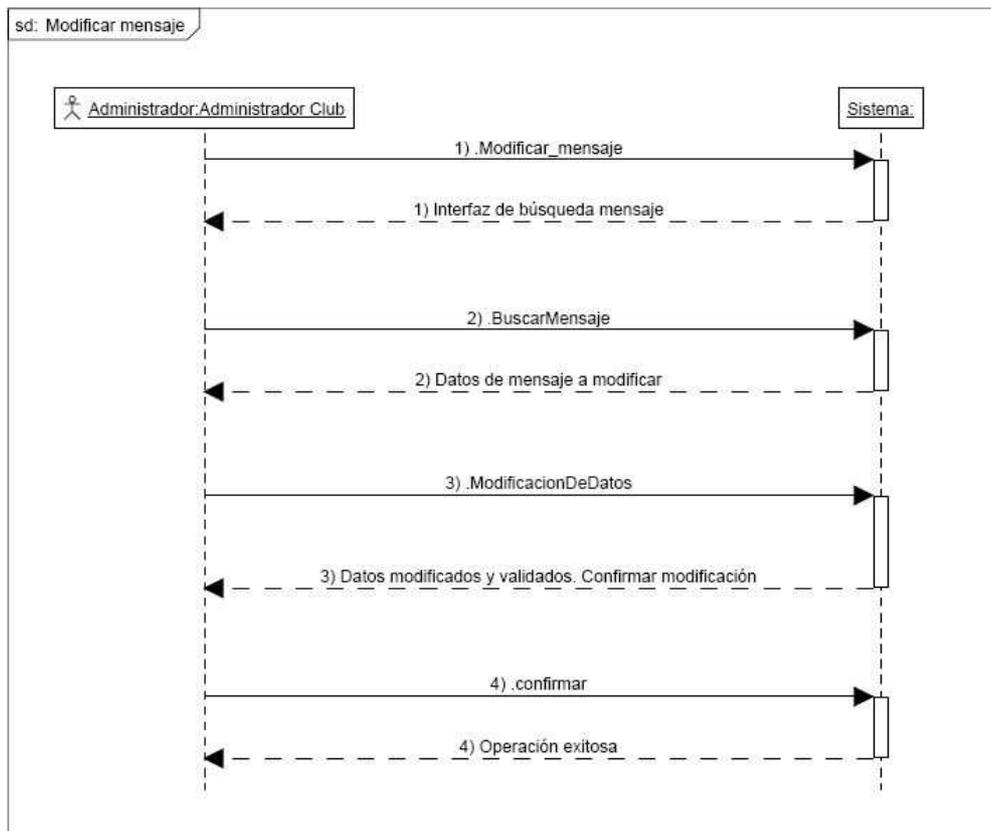


Operación	ingresar_mensaje()
Precondición	Colaborador está registrado y logueado.
Poscondición	No hay.

Operación	ingresar_mensaje(Fecha_ creación, Hora_ creación, Contenido_mensaje)
Precondición	No hay
Poscondición	Datos del mensaje son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos del mensaje son ingresados en el sistema.

Caso de Uso	RF-21 Modificar mensaje	
Objetivos asociados	Actualizar los datos de un mensaje, manteniendo la integridad y la unicidad de los datos.	
Actores asociados	ACT-02 Colaborador	
Precondición	El mensaje ya se encuentra ingresado en el sistema. 2. El actor se encuentra identificado y autenticado en el sistema.	
Secuencia Normal	Paso	Acción
	1	El caso de uso comienza cuando el usuario colaborador solicita al sistema modificar los datos de un mensaje.
	2	El sistema despliega una interfaz con los campos de búsqueda.
	3	El usuario elige un mensaje para editar.
	4	El sistema despliega la información del mensaje seleccionado, la cual es modificada por el usuario colaborador.
	5	El sistema despliega la información ingresada, y solicita confirmación de los datos.
	6	El usuario colaborador confirma los datos, presiona confirmar.
	7	El sistema informa el éxito de la operación.
	8	Fin del caso de uso.
Postcondición		
Secuencias alternativas	Paso	Acción
	4.1	El sistema no valida los datos ingresados por el usuario colaborador. El control de ejecución vuelve al paso 3.
	6.1	El sistema no valida los datos ingresados. El colaborador vuelve al paso 5.
	7.1	El colaborador no confirma los datos ingresados.
	7.2	Fin del caso de uso.
Frecuencia esperada	Media	

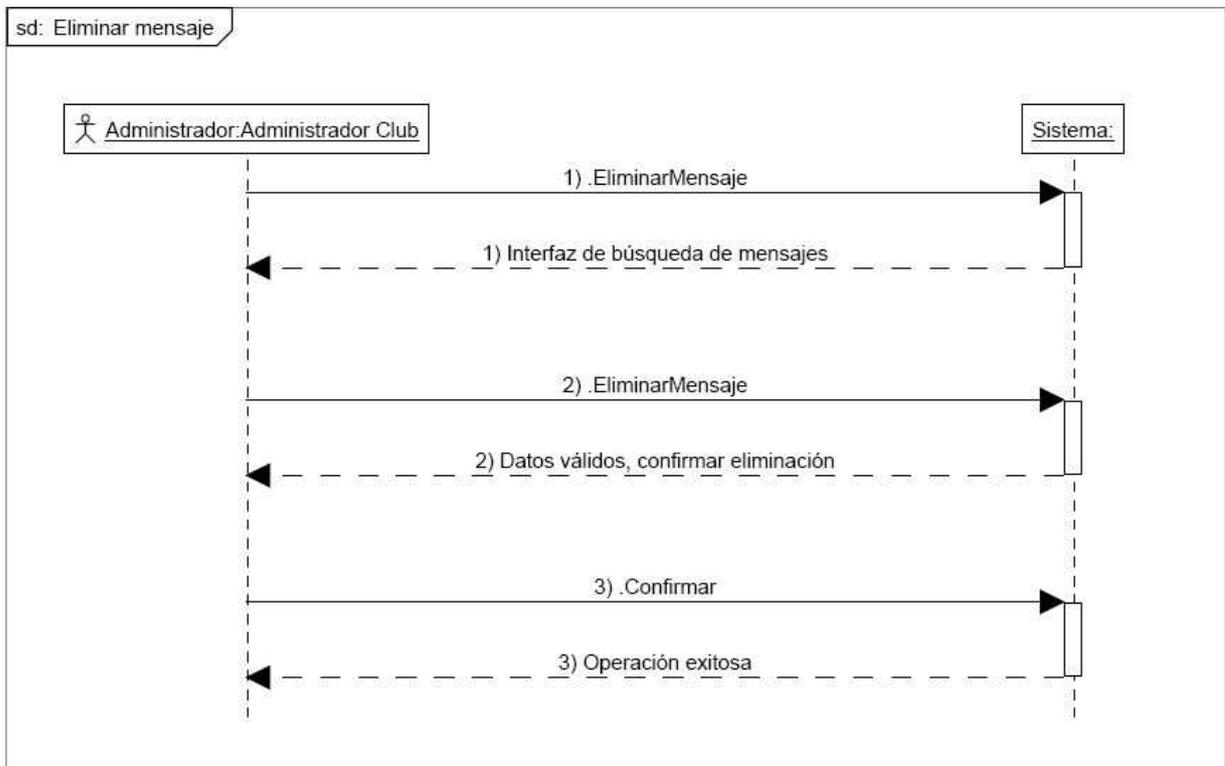


Operación	modificar_mensaje()
Precondición	Administrador está registrado.
Poscondición	No hay.

Operación	modificar(datos_modificados, login)
Precondición	Usuario ya está registrado en el sistema.
Poscondición	Datos de socio son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos del mensaje son modificados e ingresados en el sistema.

Caso de Uso	RF-22 Eliminar mensaje	
Objetivos asociados	Eliminar un registro de un mensaje desde la base de datos	
Actores asociados	ACT-03 Colaborador	
Precondición	1. Mensaje debe existir en la base de datos 2. Estar logueado e identificado	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el colaborador desea eliminar un mensaje.
	2	El colaborador selecciona el registro de un mensaje
	3	El sistema presenta por pantalla los datos del mensaje seleccionado y pide confirmación para la eliminación del registro.
	4	El colaborador confirma eliminación.
	5	El sistema procede a realizar la eliminación del registro desde la base de datos y muestra mensaje de eliminación del registro.
	6	Finaliza el caso de uso
Postcondición	Se elimina un mensaje de la tabla; pero se guarda un registro de la eliminación en una tabla de auditorias.	
Secuencias alternativas	Paso	Acción
	4.1	El colaborador cancela la eliminación
	4.2	Finaliza el caso de uso
Frecuencia esperada	Baja	
Comentarios		

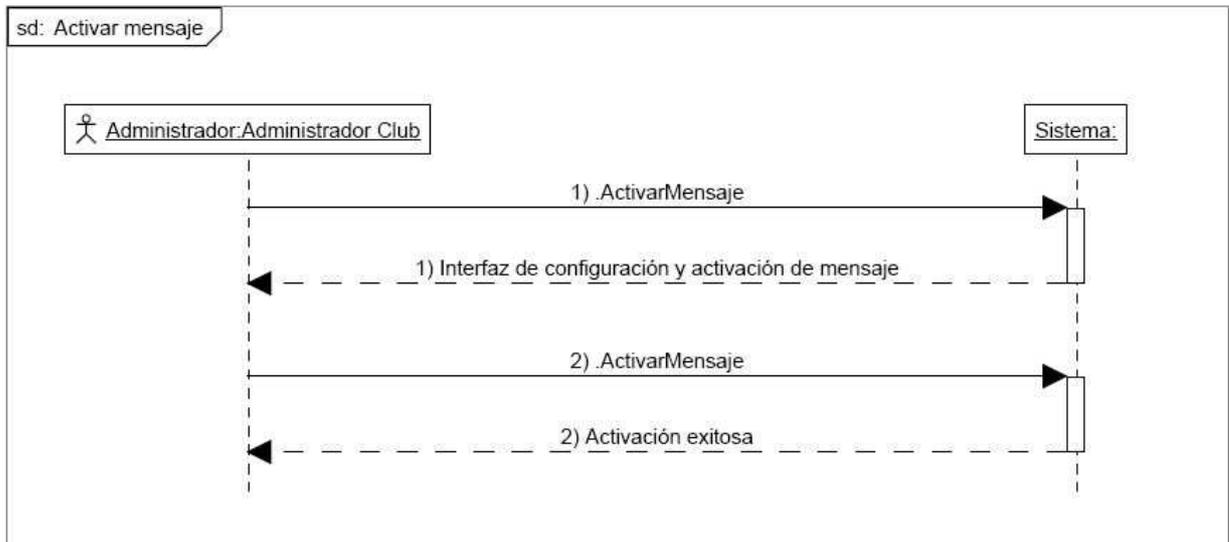


Operación	eliminar_mensaje()
Precondición	Administrador está registrado.
Poscondición	No hay.

Operación	eliminar_mensaje(id_pago)
Precondición	Administrador está registrado en el sistema.
Poscondición	Datos ingresados son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos del mensaje son eliminados del sistema.

Caso de Uso	RF-23 Activar mensaje	
Objetivos asociados	Activar el o los mensajes requeridos para que sean enviados a los socios que corresponda.	
Actores asociados	ACT-01 Administrador club de tenis	
Precondición	El mensaje debe existir en la base de datos. El actor se encuentra identificado y autenticado en el sistema	
Secuencia Normal	Paso	Acción
	1	El caso de uso comienza cuando el usuario administrador solicita al sistema activar un mensaje
	2	El sistema despliega una interfaz con los campos de búsqueda.
	3	El usuario elige un mensaje para activar.
	4	El sistema despliega la información del mensaje seleccionado, la cual puede ser modificada por el usuario administrador.
	5	El usuario activa el mensaje
	6	El sistema informa el éxito de la operación.
	7	Fin del caso de uso.
Postcondición		
Secuencias alternativas	Paso	Acción
	5.1	El usuario elige cancelar la operación
	5.2	El control de ejecución del programa vuelve al paso 2
	5.3	Fin del caso de uso.
Frecuencia esperada	Baja - Media	
Comentarios		

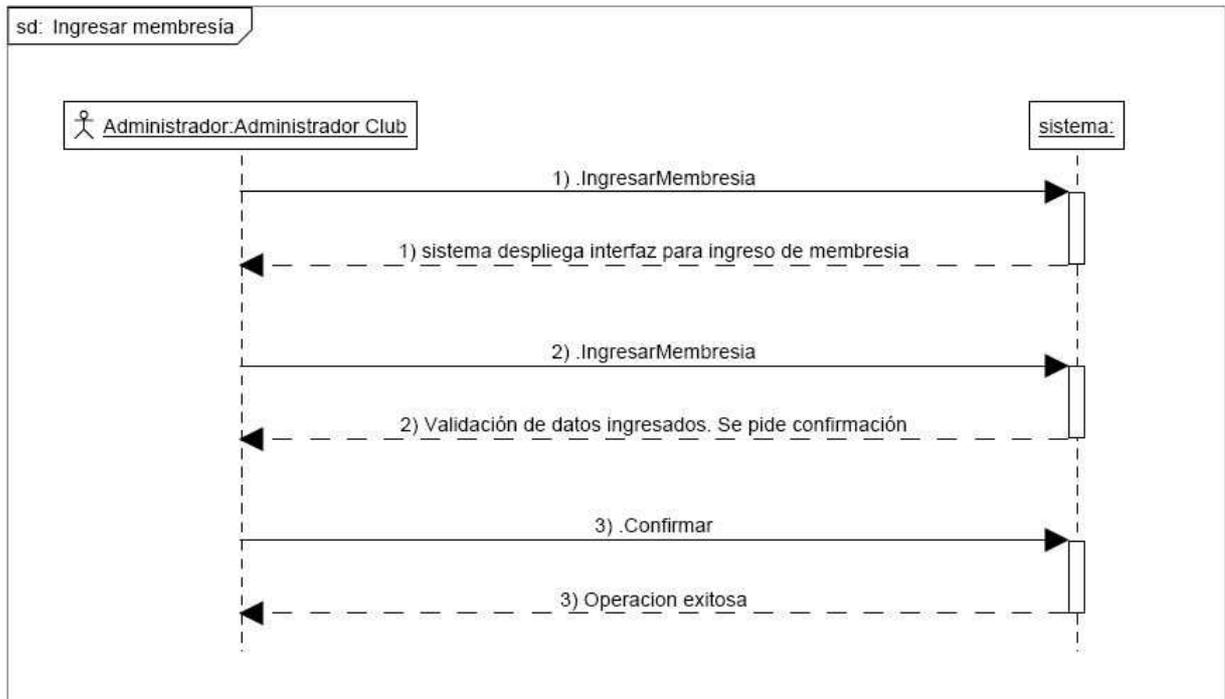


Operación	Activar_mensaje()
Precondición	Administrador está registrado y logueado.
Poscondición	No hay.

Operación	Activar_mensaje()
Precondición	No hay
Poscondición	Datos del mensaje son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos de activación del mensaje son ingresados en el sistema.

Caso de Uso	RF-24 Ingresar membresía	
Objetivos asociados	Registrar una nueva membresía en la base de datos del Club de tenis	
Actores asociados	ACT-01 Administrador club de tenis	
Precondición	1. La membresía no se encuentra ingresada en el sistema. 2. Estar logueado e identificado	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el administrador del club desea registrar una nueva membresía.
	2	El administrador ingresa los datos para la nueva membresía: Fecha de creación, Valor membresía, Descripción de la membresía, Observaciones
	3	El sistema valida los datos ingresados para el nuevo socio
	4	El sistema registra el nuevo socio
	5	Finaliza el caso de uso
Postcondición		
Secuencias alternativas	Paso	Acción
	3.1	Datos ingresados son incorrectos
	3.1.1	El sistema indica error
	3.1.2	El administrador elige cancelar o seguir para arreglar campos no válidos.
	3.1.3	Si administrador elige seguir, la ejecución vuelve a paso 2
	3.1.4	Finaliza el caso de uso
Frecuencia esperada	Baja	
Comentarios		



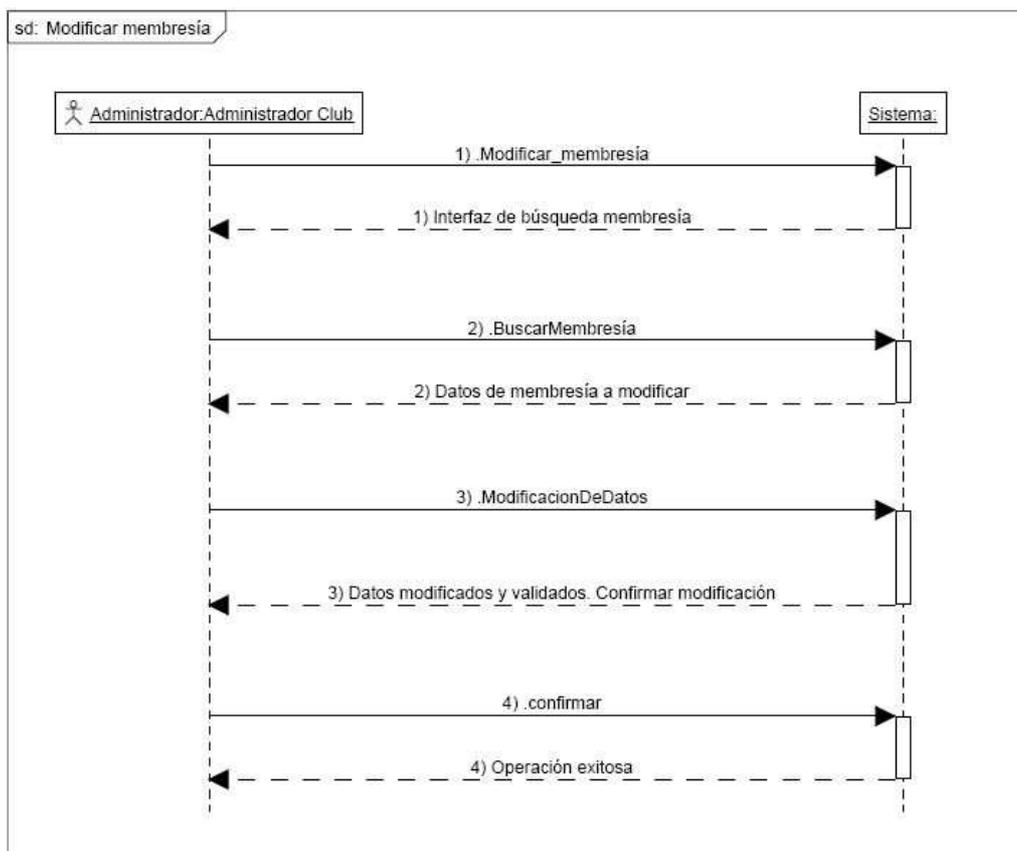
Operación	ingresar_membresía()
Precondición	Administrador está registrado y logueado.
Poscondición	No hay.

Operación	ingresar_membresía(Fecha_creación, Valor_membresía, Descripción_membresía, Observaciones)
Precondición	No hay
Poscondición	Datos de la membresía son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos de la membresía son ingresados en el sistema.

Caso de Uso	RF-25 Modificar membresía	
Objetivos asociados	Actualizar los datos de una membresía, manteniendo la integridad y la unicidad de los datos.	
Actores asociados	ACT-01 Administrador club de tenis	
Precondición	La membresía ya se encuentra ingresada en el sistema. 2. El actor se encuentra identificado y autenticado en el sistema.	
Secuencia Normal	Paso	Acción
	1	El caso de uso comienza cuando el usuario administrador solicita al sistema modificar los datos de una membresía.
	2	El sistema despliega una interfaz con los campos de búsqueda.
	3	El usuario elige una membresía para editar
	4	El sistema despliega la información de la membresía seleccionada, la cual es modificada por el usuario administrador.
	5	El sistema despliega la información ingresada, y solicita confirmación de los datos.
	6	El usuario administrador confirma los datos, presiona confirmar.
	7	El sistema informa el éxito de la operación
	8	Fin del caso de uso
Postcondición	Se graban las modificaciones en el registro de la base de datos y se crea un registro en una tabla de auditorias.	
Secuencias alternativas	Paso	Acción
	4.1	El sistema no valida los datos ingresados por el usuario administrador. El control de ejecución vuelve al paso 3.
	6.1	El sistema no valida los datos ingresados. El administrador vuelve al paso 5.

	7.1	El administrador no confirma los datos ingresados.
	7.2	Fin del caso de uso.
Frecuencia esperada	Baja	
Comentarios		

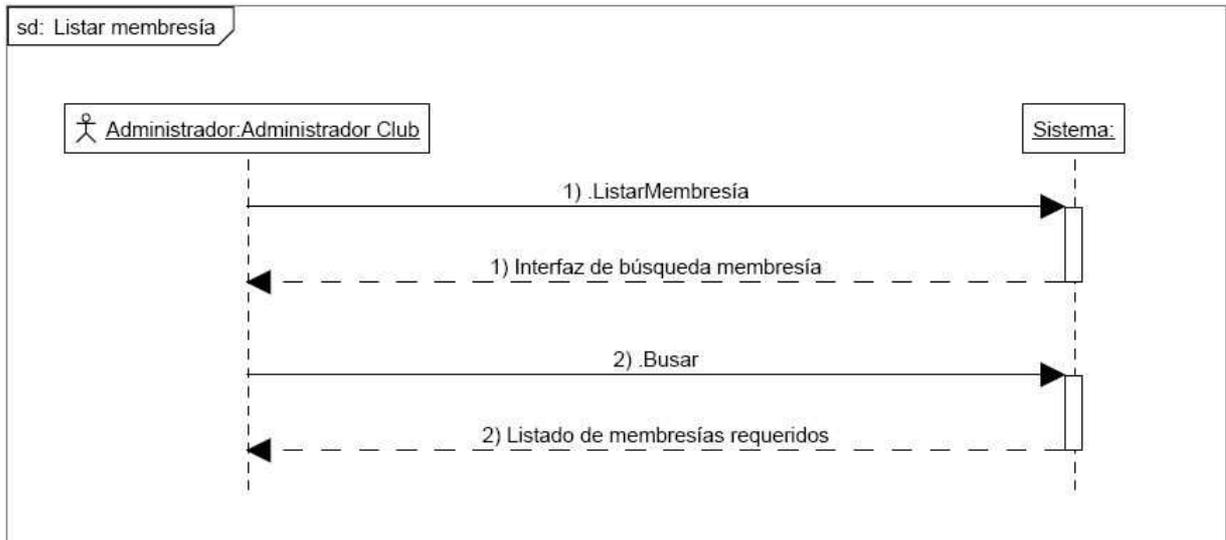


Operación	modificar_membresía()
Precondición	Administrador está registrado.
Poscondición	No hay.

Operación	modificar(datos_modificados, login)
Precondición	Usuario ya está registrado en el sistema.
Poscondición	Datos de membresía son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos son modificados e ingresados al sistema.

Caso de Uso	RF-26 Listar membresía	
Objetivos asociados	Mostrar lista completa y ordenada de las membresías que el club posee.	
Actores asociados	ACT-01 Administrador club de tenis	
Precondición	1. Estar logueado e identificado	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el administrador desea listar las membresías que el club posee.
	2	El sistema muestra el listado de reservas de las membresías
	3	El administrador elige una membresía
	4	El sistema despliega los datos elegidos
	5	Finaliza el caso de uso
Postcondición		
Secuencias alternativas	Paso	Acción
Frecuencia esperada	Baja - Media	
Comentarios		

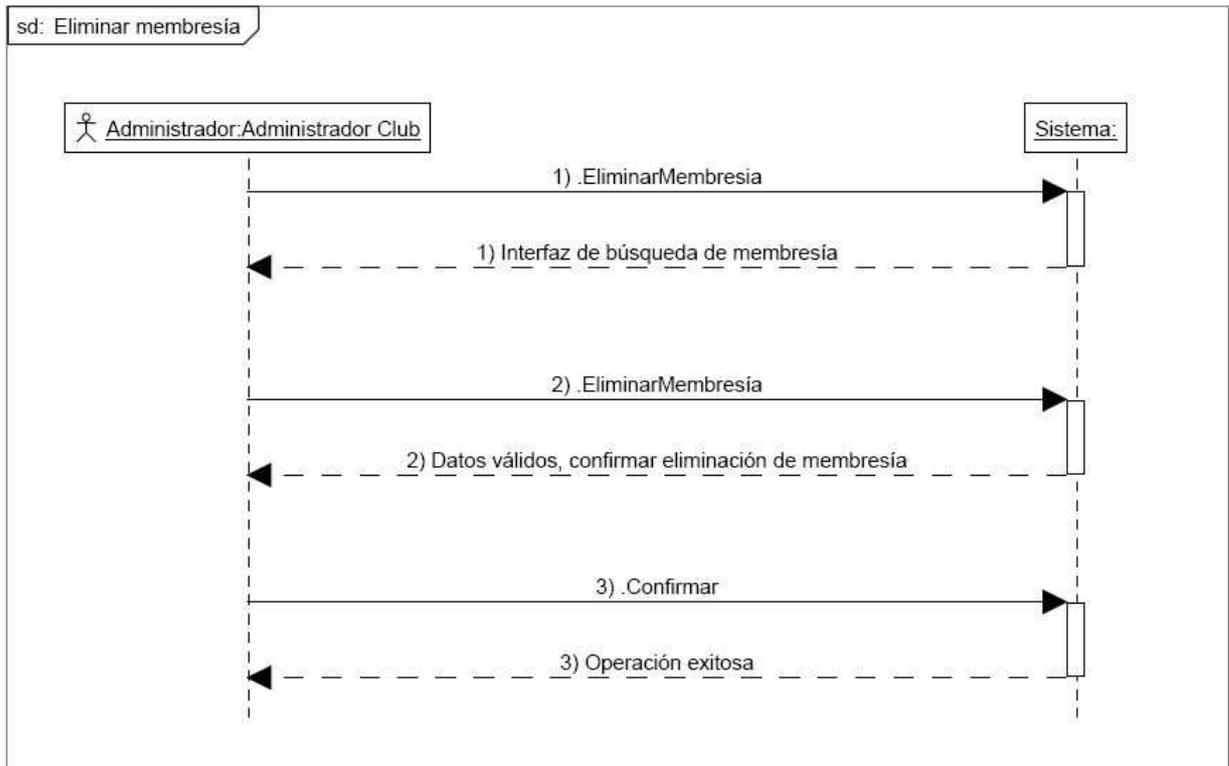


Operación	Listar_membresía()
Precondición	Usuario está registrado.
Poscondición	No hay.

Operación	buscar(id_ membresía)
Precondición	No hay
Poscondición	No hay.

Caso de Uso	RF-27 Eliminar membresía	
Objetivos asociados	Eliminar un registro de una membresía desde la base de datos	
Actores asociados	ACT-01 Administrador club de tenis	
Precondición	1. La membresía debe existir en la base de datos 2. Estar logueado e identificado	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el administrador desea eliminar una membresía.
	2	El administrador selecciona el registro de una membresía
	3	El sistema presenta por pantalla los datos de la membresía seleccionada y pide confirmación para la eliminación del registro.
	4	El administrador confirma eliminación.
	5	El sistema verifica que no haya registros de socios enlazados con la membresía a eliminar.
	6	El sistema procede a realizar la eliminación del registro desde la base de datos y muestra mensaje de eliminación del registro.
	7	Finaliza el caso de uso
Postcondición	Se elimina la membresía de la tabla; pero se guarda el registro en otra tabla histórica.	
Secuencias alternativas	Paso	Acción
	4.1	El administrador cancela la eliminación
	4.2	Finaliza el caso de uso
	5.1	El sistema detecta registros de socios que están enlazados a la membresía a eliminar. Envía un mensaje de negación de la eliminación de la membresía.
	5.2	El control de ejecución vuelve al paso 2

	5.3	Finaliza el caso de uso
Rendimiento	Paso	Cota de tiempo
Frecuencia esperada	Muy baja	
Comentarios		

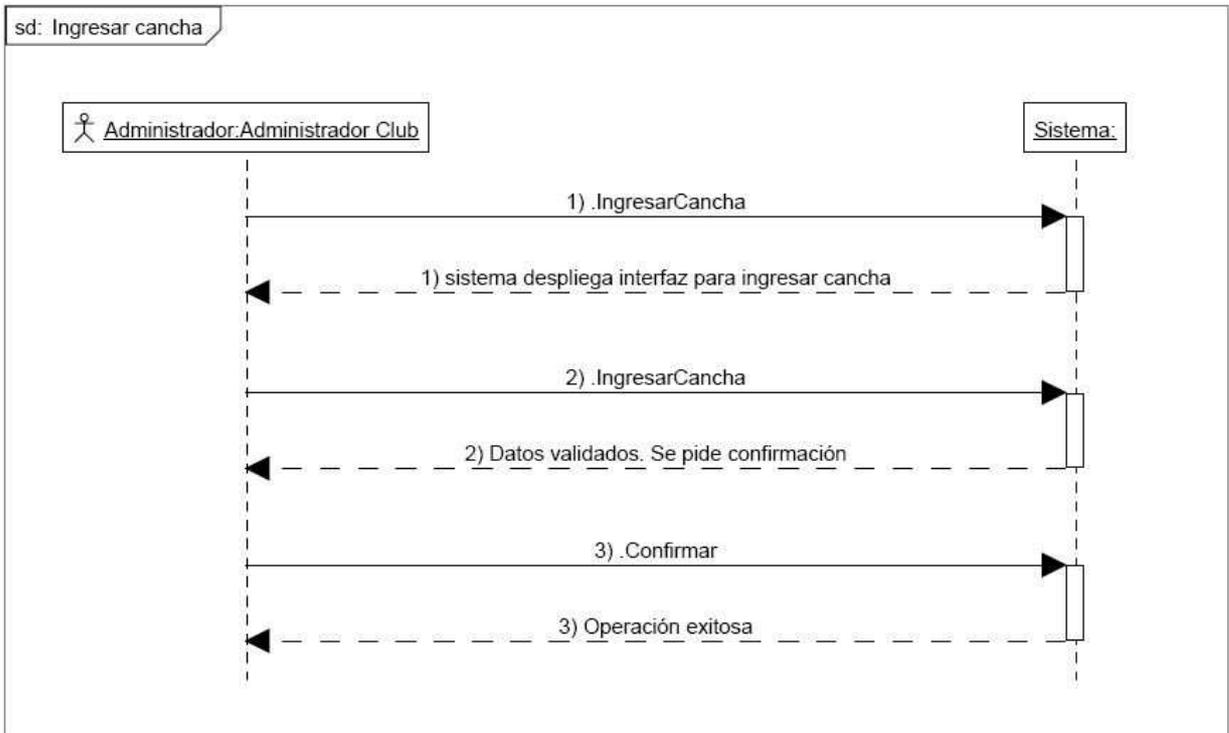


Operación	eliminar_membresia()
Precondición	Administrador está registrado.
Poscondición	No hay.

Operación	eliminar_membresia(id_membresia)
Precondición	Administrador está registrado en el sistema.
Poscondición	Datos ingresados son validados.

Operación	Confirmar()
Precondición	No hay.
Poscondición	Datos de la membresía son eliminados del sistema.

Caso de Uso	RF-28 Ingresar nueva cancha	
Objetivos asociados	Registrar una nueva cancha de tenis en la base de datos del Club de tenis.	
Requisitos asociados		
Actores asociados	ACT-01 Administrador club de tenis	
Precondición	1. La cancha de tenis no se encuentra ingresada en el sistema. 2. Estar logueado e identificado	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el administrador del club desea registrar una nueva cancha de tenis.
	2	El administrador ingresa los datos de identificación de la nueva cancha : Tipo de superficie, Fotografía, Posee Iluminación artificial, Posee galerías, Disponibilidad, Observaciones
	3	El sistema valida los datos ingresados para la nueva cancha
	4	El sistema registra la nueva cancha
	5	Finaliza el caso de uso
Postcondición		
Secuencias alternativas	Paso	Acción
	3.1	Datos ingresados son incorrectos
	3.1.1	El sistema indica error
	3.1.2	El administrador elige cancelar o seguir para arreglar campos no válidos.
	3.1.3	Si administrador elige seguir, la ejecución vuelve a paso 2
	3.1.4	Finaliza el caso de uso
Frecuencia esperada	Muy Baja	

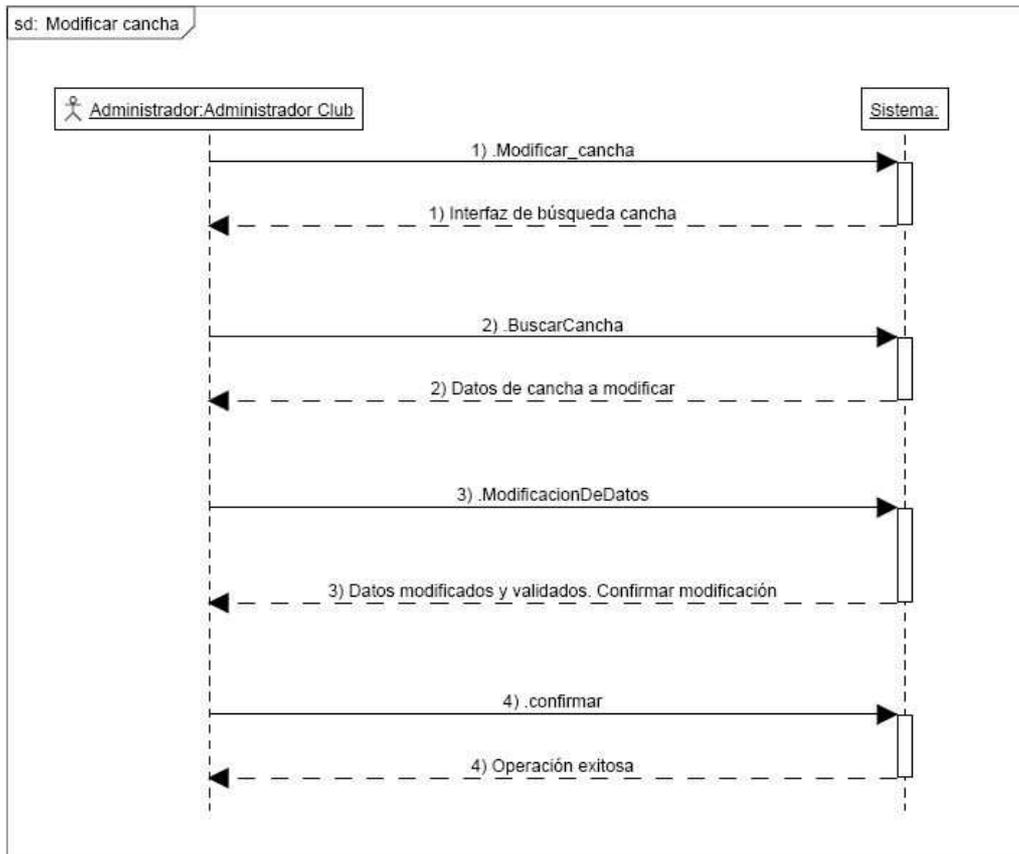


Operación	ingresar_cancha()
Precondición	Administrador está registrado y logueado.
Poscondición	No hay.

Operación	ingresar_cancha(Tipo_superficie, Fotografía Iluminación, galerías, Disponibilidad, Observaciones)
Precondición	No hay
Poscondición	Datos de la cancha son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos de la cancha son ingresados en el sistema.

Caso de Uso	RF-29 Modificar cancha	
Objetivos asociados	Cambiar datos del registro de una cancha de tenis	
Actores asociados	ACT-01 Administrador club de tenis	
Precondición	1. Estar logueado e identificado	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el administrador desea modificar los datos de una cancha.
	2	El administrador selecciona el registro de una cancha.
	3	El sistema presenta por pantalla los datos de la cancha de tenis elegida.
	4	El administrador modifica los campos editables y/o datos deseados
	5	El sistema valida los cambios realizados y pide confirmación al administrador para ejecutar las modificaciones.
	6	El administrador confirma los cambios.
	7	El sistema registra las modificaciones del registro en la BD
	8	Finaliza el caso de uso
Postcondición		
Secuencias alternativas	Paso	Acción
	5.1	Los datos ingresados son incorrectos
	5.2	El sistema presenta mensaje de error en la validación de los datos.
	5.3	El control de ejecución vuelve al paso número 3
	6.1	El administrador cancela la modificación de los datos
	6.2	Finaliza el caso de uso.
Frecuencia esperada	Baja	
Comentarios		

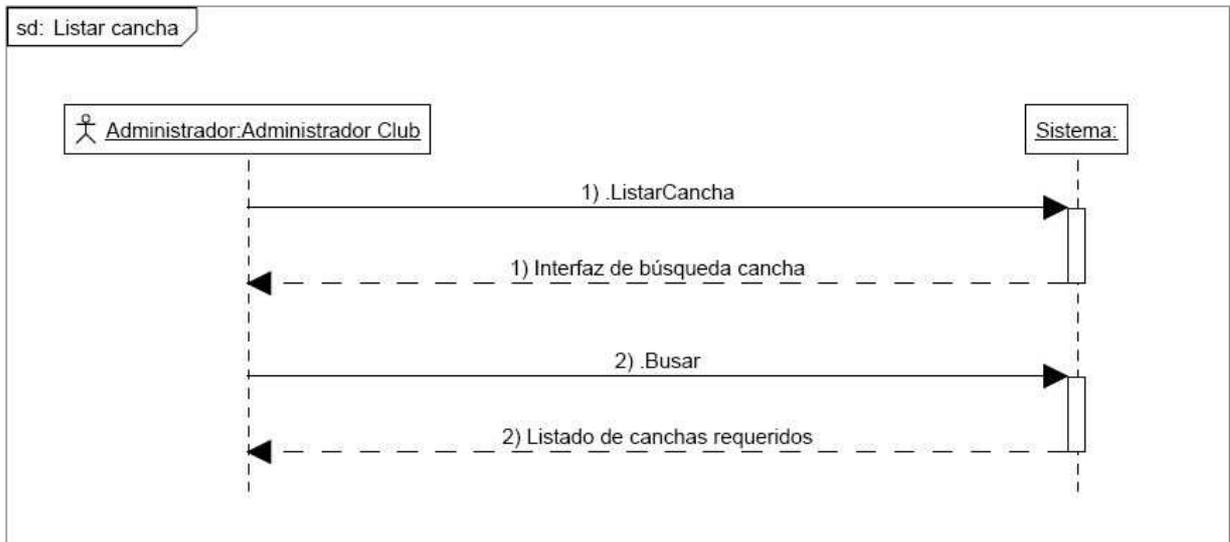


Operación	modificar_cancha()
Precondición	Administrador está registrado.
Poscondición	No hay.

Operación	modificar(datos_modificados, login)
Precondición	Usuario ya está registrado en el sistema.
Poscondición	Datos de la cancha son validados.

Operación	confirmar()
Precondición	No hay.
Poscondición	Datos son modificados e ingresados all sistema.

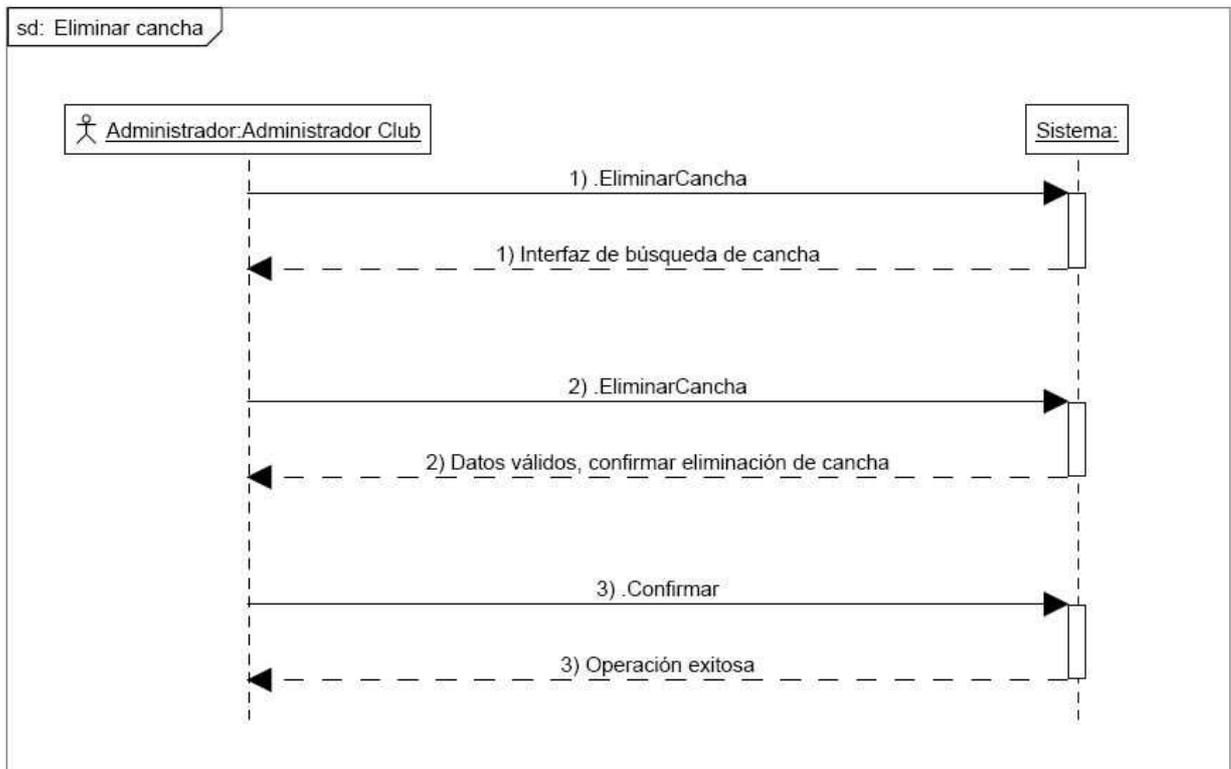
Caso de Uso	RF-30 Listar cancha	
Objetivos asociados	Mostrar lista completa y ordenada de las canchas del club desde donde se puede obtener la información parametrizada de ellos.	
Actores asociados	ACT-01 Administrador club de tenis ACT-02 Socio	
Precondición	1. Estar logueado e identificado	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando un actor desea listar las canchas de tenis que el club posee.
	2	El sistema muestra en forma ordenada numéricamente el listado de las canchas.
	3	El actor elige una cancha para ver sus datos en detalle
	4	El sistema despliega los datos específicos de la cancha elegida.
	5	Finaliza el caso de uso.
Postcondición		
Secuencias alternativas	Paso	Acción
	4.1	El actor ACT-02 puede pedir reserva de la cancha elegida.
	4.2	El control de ejecución pasa al caso de uso RF-08.
	4.3	Finaliza el caso de uso
Frecuencia esperada	Alta	
Comentarios		



Operación	Listar_cancha()
Precondición	Usuario está registrado.
Poscondición	No hay.

Operación	buscar(id_cancha)
Precondición	No hay
Poscondición	No hay.

Caso de Uso	RF-31 Eliminar cancha	
Objetivos asociados	Eliminar registro de una cancha	
Actores asociados	ACT-01 Administrador club de tenis	
Precondición	1. La cancha se encuentra ingresada en el sistema. 2. Estar logueado e identificado.	
Secuencia Normal	Paso	Acción
	1	El caso de uso se inicia cuando el administrador desea eliminar una cancha.
	2	El administrador selecciona el registro de una cancha
	3	El sistema presenta por pantalla los datos de la cancha seleccionada y pide confirmación para la eliminación del registro.
	4	El administrador confirma eliminación.
	5	El sistema procede a realizar la eliminación del registro desde la base de datos y muestra mensaje de eliminación del registro.
	6	Finaliza el caso de uso
Postcondición	Se elimina la cancha; pero se guarda el registro en otra tabla histórica Y se mantienen los registros de otras tablas asociadas a esta a través de algún campo específico, como las reservas históricas de esta cancha.	
Secuencias alternativas	Paso	Acción
	4.1	El administrador cancela la eliminación
	4.2	Finaliza el caso de uso
Rendimiento	Paso	Cota de tiempo
Frecuencia esperada	Muy baja	
Comentarios		



Operación	eliminar_cancha()
Precondición	Administrador está registrado.
Poscondición	No hay.

Operación	eliminar_cancha(id_cancha)
Precondición	Administrador está registrado en el sistema.
Poscondición	Datos ingresados son validados.

Operación	Confirmar()
Precondición	No hay.
Poscondición	Datos de la cancha son eliminados del sistema.

ANEXO C: CODIFICACION DE CLASES

CODIFICACION DE LA CLASE SOCIOS CAPA CONTROLADOR

```
class SociosController extends ApplicationController {
    var $name = 'Socios';
    var $helpers = array('Html', 'Form', 'javascript');
    function beforeFilter() {
        $this->checkSession();
    }
    function index() {
        $this->checkSession();
        $this->Socio->recursive = 0;
        $this->set('socios',$this->Socio->findAll(null, array(), 'nombresocio ASC') );
    }
    function listasocios() {
        $this->checkSession();
        $this->Socio->recursive = 0;
        $this->set('socios',$this->Socio->findAll(null, array(), 'nombresocio ASC') );
    }
    function view($id = null) {
        $this->checkSession();
        if (!$id) {
            $this->flash('Id no válido para socio', '/socios/index');
        }
        $this->set('socio', $this->Socio->read(null, $id));
    }
    function view_admin($id = null) {
        $this->checkSession();
        if (!$id) {
            $this->flash('Id no válido para socio', '/socios/index');
        }
        $this->set('socio', $this->Socio->read(null, $id));
    }
}
```

```
function add() {
    $this->checkSession();
    if (empty($this->data)) {
        $this->set('membresias', $this->Socio->Membresia->generateList());
        $this->render();
    } else {
        $this->cleanUpFields();
        if ($this->Socio->save($this->data)) {
            $this->Session->setFlash('El nuevo socio ha sido ingresado correctamente');
            $this->redirect('/socios/index');
        } else {
            $this->set('membresias', $this->Socio->Membresia->generateList());
        }
    }
}

function edit($id = null) {
    $this->checkSession();
    if (empty($this->data)) {
        if (!$id) {
            $this->flash('Id no válido', '/socios/index');
        }
        $this->data = $this->Socio->read(null, $id);
        $this->set('membresias', $this->Socio->Membresia->generateList());
    } else {
        $this->cleanUpFields();
        if ($this->Socio->save($this->data)) {
            $this->flash('Datos del socio han sido actualizados correctamente.',
'/socios/index/');
        } else {
            $this->set('membresias', $this->Socio->Membresia->generateList());
        }
    }
}
}
```

```
function edit_socio($id = null) {  
    $this->checkSession();  
    $this->set('socio', $this->Socio->read(null, $id));  
    if (empty($this->data)) {  
        if (!$id) {  
            $this->flash('Id no válido', '/socios/view/'.$id);  
        }  
        $this->data = $this->Socio->read(null, $id);  
    } else {  
        $this->cleanUpFields();  
        if ($this->Socio->save($this->data)) {  
            $this->flash('Datos del socio han sido actualizados correctamente.', '/socios/view/'.$id);  
        } else {  
            $this->set('membresias', $this->Socio->Membresia->generateList());  
        }  
    }  
}  
  
function delete($id = null) {  
    $this->checkSession();  
    if (!$id) {  
        $this->flash('Id no válido', '/socios/index');  
    }  
    if ($this->Socio->del($id)) {  
        $this->flash('Socio con Id: '.$id.' borrado', '/socios/index');  
    }  
    }  
}
```

CODIFICACION DE LA CLASE SOCIOS CAPA MODELO

```
class Socio extends AppModel {
    var $name = 'Socio';
    var $displayField = 'nombresocio';
var $validate = array(
    'rutsocio' => array('required' => VALID_NOT_EMPTY,'length' => array( 'rule' => 'validateLength',
    'min' => 5, 'max' => 100 ),'validRut' => array('rule'=>'validateRut')),
    'nombresocio' => VALID_NOT_EMPTY,
    'apellido1socio' => VALID_NOT_EMPTY,
    'direccionsocio' => VALID_NOT_EMPTY,
    'ciudad socio' => VALID_NOT_EMPTY,
    'fechanacimiento' => VALID_NOT_EMPTY,
    'fechaingresosocio' => VALID_NOT_EMPTY,
    'fonofijo1socio' => VALID_NOT_EMPTY,
    'fonomovilsocio' => VALID_NUMBER,
    'emailsocio' => VALID_EMAIL,
    'loginsocio' => '/^{4,12}$/',
    'passwordsocio' => '/^{4,12}$/',);
var $belongsTo = array(
    'Membresia' =>      array('className' => 'Membresia',
                                'foreignKey' => 'membresia_id',
                                'conditions' => "",
                                'fields' => "",
                                'order' => "",
                                'counterCache' => ""
                                ),
    );
var $hasMany = array(
    'Reserva' =>      array('className' => 'Reserva',
                                'foreignKey' => 'socio_id',
                                'conditions' => "",
                                'fields' => "",
```

```
        'order' => "",
        'limit' => "",
        'offset' => "",
        'dependent' => "",
        'exclusive' => "",
        'finderQuery' => "",
        'counterQuery' => "" ),
'Pago' =>    array('className' => 'Pago',
                'foreignKey' => 'socio_id',
                'conditions' => "",
                'fields' => "",
                'order' => "",
                'limit' => "",
                'offset' => "",
                'dependent' => "",
                'exclusive' => "",
                'finderQuery' => "",
                'counterQuery' => ""
            ),
'Image' =>    array('className' => 'Image',
                'foreignKey' => 'socio_id',
                'conditions' => "",
                'fields' => "",
                'order' => "",
                'counterCache' => "" ), );
    }
```

ANEXO D: PLAN DE PRUEBAS

Caso de prueba 1	Sólo letras- ingresar Socio-nombre
Propósito	Determinar que en los textbox de ingreso de datos sólo registren letras alfanuméricas del alfabeto español.
PRE-requisitos	Usuario registrado, Inicio de sesión, perfil válido, ingresar datos de un Socio válido
Datos de Prueba	TxtNombre={ números, letras, caracteres especiales, copiar un número y pegar en txtNombre, vacío }
Pasos	<ol style="list-style-type: none"> 1. visitar Login 2. teclear username 3. teclear password 4. hacer click en Entrar 5. visitar interfaz ingresar Socio 6. teclear nombre 7. hacer click en guardar
Notas y Preguntas	

Caso de prueba 2	Validación de Rut-formato
Propósito	Determinar que en textbox donde se ingrese el Rut valide el formato de este incluyendo las letras permitidas
PRE-requisitos	Estar situado en una interfaz de búsqueda o ingreso de datos.
Datos de Prueba	TxtRut={ números, letras, caracteres especiales, copiar un número y pegar en txtRut, vacío, cadenas con espacios al inicio y final }
Pasos	<ol style="list-style-type: none"> 1. teclear un rut en el textbox 2. generar un evento de “pérdida de foco”
Notas y Preguntas	

Caso de prueba 3	Sólo números en campos de teléfonos
Propósito	Restringir que en los textbox de ingreso de datos correspondientes a números telefónicos (cod_area, teléfono fijo, teléfono móvil) solo registren caracteres numéricos en el string.
PRE-requisitos	Usuario registrado, Inicio de sesión, perfil valido, ingresar datos de un Socio valido.
Datos de Prueba	TxtCod_Area={ números, letras, caracteres especiales, vacío } TxtTelefonoFijo={ números, letras, caracteres especiales, vacío } TxtTelefonoMovil={ números, letras, caracteres especiales, vacío }

Pasos	<ol style="list-style-type: none">1. visitar Login2. teclear username3. teclear password4. hacer click en aceptar5. visitar interfaz Ingresar Socio.6. teclear cod_area o telefonofijo o telefonomovil7. hacer click en ingresar
Notas y Preguntas	

Caso de prueba 4	Sólo letras en otros campos
Propósito	Restringir que en los textbox de ingreso de datos correspondientes a números telefónicos sólo se pueda registrar caracteres numéricos.
PRE-requisitos	Usuario registrado, Inicio de sesión, perfil valido, ingresar datos de un Socio valido.
Datos de Prueba	TxtNombre={números, letras, caracteres especiales, vacío}; TtxtaPaterno={números, letras, caracteres especiales, vacío} TtxtaMaterno={números, letras, caracteres especiales, vacío}
Pasos	<ol style="list-style-type: none"> 1. visitar Login 2. teclear username 3. teclear password 4. hacer click en aceptar 5. visitar interfaz Ingresar Socio 6. teclear cod_area o teléfono fijo o teléfono móvil 7. hacer click en ingresar
Notas y Preguntas	Este caso es aplicable a cualquier interfaz de ingreso o modificación

Caso de prueba 5	Validación de password al ingresar un usuario
Propósito	Validar que la password ingresada y su confirmación sean iguales
PRE-requisitos	Otro usuario registrado, Inicio de sesión, perfil valido, cargar interfaz ingresar juez
Datos de Prueba	txtPassword1={*****}; txtPassword2={*****};
Pasos	<ol style="list-style-type: none"> 1. visitar Login 2. teclear username 3. teclear password 4. hacer click en aceptar 5. visitar interfaz Ingresar Socio 6. teclear cod_area o telefono fijo o telefono movil 7. hacer click en ingresar
Notas y Preguntas	Esta prueba es aplicable también a la interfaz ingresar asistente judicial, así mismo como las modificaciones de juez y/o asistente judicial.

Caso de prueba 6	Validación criterios de ordenamiento.
Propósito	Validar que los datos sean listados de acuerdo al criterio en que han sido seleccionado para su ordenamiento.
PRE-requisitos	Otro usuario registrado, Inicio de sesión, perfil válido, cargar interfaz Listar Reservas
Datos de Prueba	
Pasos	<ol style="list-style-type: none"> 1. visitar Login 2. teclear username 3. teclear password 4. hacer click en aceptar 5. visitar interfaz Listar Reservas. 6. seleccionar criterio de ordenamiento. 7. hacer click en siguiente.
Notas y Preguntas	Esta prueba es aplicable también a la interfaz listar Socio, listar pagos, listar noticias.

Caso de prueba 7	Validación de datos Modificados
Propósito	Validar que en los textbox de modificación de datos correspondientes a números telefónicos (cod_area, telefonofijo, telefonomovil) se modifiquen los números correspondientes y que en los textbox de modificación de datos correspondientes a caracteres (ciudad, email, dirección) se modifiquen los caracteres correspondientes.
PRE-requisitos	Otro usuario registrado, Inicio de sesión, perfil valido, cargar interfaz Modificar Socio.
Datos de Prueba	<p>TxtCod_Area={números, letras, caracteres especiales, vacío}</p> <p>TxtTelefonoFijo={números, letras, caracteres especiales, vacío}</p> <p>TxtTelefonoMovil={números, letras, caracteres especiales, vacío}</p> <p>TxtNombre={números, letras, caracteres especiales, vacío};</p> <p>TxtaPaterno={números, letras, caracteres especiales, vacío}</p> <p>TxtaMaterno={números, letras, caracteres especiales, vacío}</p>
Pasos	<ol style="list-style-type: none"> 1. visitar Login 2. teclear username 3. teclear password 4. hacer click en aceptar 5. visitar interfaz Modificar Socio. 6. teclear Rut Socio válido. 7. hacer click en siguiente. 8. teclear cod_area o telefonofijo o telefonomovil o ciudad o email o dirección o estado a modificar. 9. hacer click en Modificar. 10. visitar interfaz Ver detalle Socio.
Notas y Preguntas	Esta prueba es aplicable también a la interfaz modificar juez y/o asistente judicial.

Caso de prueba 8	Validación cambio de estado de mensajes
Propósito	Validar que un mensaje cambia su estado activo a inactivo dentro del sistema.
PRE-requisitos	Otro usuario registrado, Inicio de sesión, perfil valido, cargar interfaz Cambiar estado de mensaje.
Datos de Prueba	
Pasos	<ol style="list-style-type: none">1. visitar Login2. teclear username3. teclear password4. hacer click en aceptar5. visitar interfaz Cambiar Juez Activo.6. seleccionar cambiar estado.7. seleccionar juez de la lista.8. hacer click en continuar.9. confirmar cambio de estado.10. hacer click en aceptar.
Notas y Preguntas	

ANEXO E: EXTRACTO DEL REGLAMENTO SOBRE CONCESIÓN DE PERSONALIDAD JURÍDICA A CORPORACIONES Y FUNDACIONES QUE INDICA (DOCUMENTO-110, 1979, MINISTERIO DE JUSTICIA).

De las Corporaciones

Artículo 2.- Las Corporaciones podrán constituirse por instrumento privado reducido a escritura pública. Dicho instrumento privado deberá ser firmado por todos los constituyentes, individualizados con su Rol Único Nacional o Tributario, contendrá el acta de constitución, los estatutos por los cuales ha de regirse y el poder que se confiere a la persona a quien se encarga la reducción a escritura pública de dicha acta, como asimismo la tramitación de la solicitud de aprobación de los estatutos y la aceptación de las modificaciones que el Presidente de la República proponga introducirles. Sin embargo, para los efectos de conceder personalidad jurídica a las corporaciones que se sujeten a un estatuto tipo aprobado por el Ministerio de Justicia, se estará a lo dispuesto en el artículo 29 de este Reglamento.

Artículo 6.- Las corporaciones no podrán proponerse fines sindicales o de lucro, ni aquéllos de las entidades que deban regirse por un estatuto legal propio. Sin perjuicio de lo anterior, se les permitirá fomentar, practicar y desarrollar, por todos los medios a su alcance, cualquiera obra de progreso social o de beneficio para la comunidad y colaborar con las instituciones legalmente constituidas en todo lo que tienda al cumplimiento de sus fines.

Artículo 7.- La corporación deberá contar con medios económicos que garanticen el cumplimiento de sus fines, circunstancia que acreditará ante el Ministerio de Justicia, mediante declaración jurada notarial que presten el Presidente y el Secretario del Directorio, o en general, con instrumentos, tales como depósitos a plazo, vale-vista, libretas de ahorro u otro de similar naturaleza. Los medios económicos de una corporación pueden consistir en cuotas de ingreso o de incorporación ordinaria o extraordinaria debiendo las de ingreso o incorporación y las ordinarias, fijarse en asamblea general ordinaria de socios y las extraordinarias, en asamblea general extraordinaria, a propuesta del directorio. En todo caso en los estatutos de la corporación deberá señalarse el valor mínimo y máximo de estas cuotas, pudiendo ser expresados en una unidad económica reajutable de actual vigencia. Las cuotas de incorporación sólo podrán establecerse respecto de cada socio por una sola vez. Las cuotas extraordinarias sólo podrán destinarse o invertirse en los fines que motivaron su establecimiento.