



**Universitat Autònoma  
de Barcelona**

# ***IMPLANTACIÓ CRM EN UN CLUB DEPORTIVO***

Memoria del proyecto  
de Ingeniería Técnica en  
Informática de Sistemas

Realizado por:

**Verónica Cuesta Martínez**

Y dirigido por:

**Xavier Verge Mestre**



**Escuela de Ingeniería**

Sabadell, junio de 2011

El abajo firmante, **Xavier Verge Mestre**, profesor de la Escuela de Ingeniería de la Universidad Autónoma de Barcelona.

**CERTIFICA:**

Que el trabajo al que corresponde la presente memoria ha estado realizado bajo su dirección por **Verónica Cuesta Martínez**, y para que conste firma la presente.

Sabadell, 28 de junio de 2011

-----

Firmado: **Xavier Verge Mestre**

**UAB**  
Universitat Autònoma  
de Barcelona

El abajo firmante, **Óscar Cubillo Alonso**, de **UNIT4**

**CERTIFICA:**

Que el trabajo al que corresponde la presente memoria ha estado realizado bajo su supervisión y tutela por **Verónica Cuesta Martínez**, y para que conste firma la presente.

Sabadell, 28 de junio de 2011

-----  
Firmado: **Óscar Cubillo Alonso**

**UNIT4**  
BUSINESS SOFTWARE

## **RESUMEN**

La propuesta CRM que se pretende ofrecer consiste en una herramienta estratégica de negocio para gestionar las relaciones con los clientes. La aplicación está desarrollada bajo un prisma Siebel para un determinado club deportivo que por motivos de privacidad no se facilitará el nombre. Si que se puede indicar que su actividad se centra en el mundo del fútbol y que tiene un elevado número de socios. Esta herramienta ha de permitir estudiar todas las posibles oportunidades de negocio cubriendo la necesidad de saber en cualquier momento la circunstancia personalizada de cada cliente (socio o no socio) y de sus trámites.

El objetivo fundamental del CRM es lograr la fidelización del cliente maximizando los recursos utilizados, mejorar los servicios ofrecidos en un grado óptimo y aumentar la cuenta de resultados. Para ello se plantea implantar una aplicación totalmente configurada y adaptada a las necesidades de un club deportivo.

## **AGRADECIMIENTOS**

Agradezco cordialmente a UNIT4 y a la Escuela de Ingeniería la oportunidad que me han ofrecido para poder desarrollar el proyecto final de carrera adquiriendo la experiencia necesaria para esta nueva etapa que se me presenta y en la que se me encauza a un nuevo futuro profesional.

A Xavier Verge, como director en la parte universitaria, por su compromiso y apoyo demostrados.

A Óscar Cubillo como tutor de la empresa y, como a un gran colaborador por su soporte, orientación y ayuda con su experiencia aportada, a Xavier Cerdeño.

Y finalmente, a Oriol Civit y Marc Pla por la paciencia y apoyo también demostrados en el hito de mi aprendizaje.

## **TABLA DE CONTENIDO**

1	INTRODUCCIÓN .....	10
1.1	PRESENTACIÓN DEL PROYECTO .....	10
1.2	CONVENIO CON UNIT4 .....	10
1.3	LA EMPRESA .....	11
1.4	OBJETIVOS .....	13
1.5	ESTADO ACTUAL .....	14
1.6	MOTIVACIÓN PERSONAL .....	15
2	ESTUDIO DE VIABILIDAD .....	18
2.1	INTRODUCCIÓN .....	18
2.1.1	Tipología y palabras clave .....	18
2.1.2	Descripción .....	18
2.1.3	Objetivos del proyecto.....	18
2.1.4	Definiciones, acrónimos y abreviaciones .....	19
2.1.5	Perfiles de usuario .....	19
2.1.6	Referencias .....	20
2.1.7	Producto y documentación del proyecto .....	21
2.2	ESTUDIO DE LA SITUACIÓN ACTUAL .....	22
2.2.1	Contexto.....	22
2.2.2	Lógica del sistema y arquitectura.....	22
2.2.3	Entorno físico de trabajo.....	23
2.2.4	Diagnóstico del sistema .....	24
2.3	RECURSOS DEL PROYECTO .....	25
2.3.1	Descripción de recursos disponibles.....	25
2.3.2	Restricciones del sistema .....	26
2.3.3	Evaluación de riesgos y posibles soluciones.....	26
2.3.4	Recursos.....	27
2.4	ORGANIZACIÓN DEL PROYECTO .....	28
2.5	PLANIFICACIÓN DEL PROYECTO (MS PROJECT) .....	30
2.5.1	Etapas.....	30
2.5.2	Cronograma inicial .....	30
2.5.3	Cronograma real y desviación.....	30

2.6	PRESUPUESTO.....	31
2.7	POSIBLES ALTERNATIVAS.....	31
2.7.1	Aternativa #1.....	32
2.7.2	Alternativa #2.....	32
2.7.3	Alternativa #3.....	33
2.8	CONCLUSIONES.....	34
3	ANÁLISIS.....	36
3.1	INTRODUCCIÓN.....	36
3.2	REQUERIMIENTOS FUNCIONALES.....	36
3.3	REQUERIMIENTOS NO FUNCIONALES.....	38
3.3.1	Técnicos.....	38
3.3.2	De usabilidad.....	38
3.3.3	De seguridad.....	38
3.3.4	Espacio de almacenamiento: base de datos.....	39
3.3.5	Corporativos.....	39
3.3.6	Académicos.....	39
3.3.7	Catalogación i priorización de los requerimientos.....	40
3.4	CASOS DE USO.....	41
4	DISEÑO.....	50
4.1	METODOLOGÍA.....	50
4.2	TECNOLOGÍAS.....	50
4.3	BASE DE DATOS.....	54
4.4	MODELO DE DATOS.....	54
4.4.1	Diagrama entidad/relación.....	55
4.4.2	DISEÑO CONCEPTUAL DE LA BASE DE DATOS.....	56
4.4.3	Descripción de las tablas.....	57
4.5	INTERFACE DE USUARIO.....	59
4.5.1	Interface inicial (login).....	59
4.5.2	Menú de administrador.....	60
4.5.3	Interface de administrador.....	60
4.5.4	Menú de usuario no administrador.....	63
4.5.5	Menú de página inicial.....	63
4.5.6	Menú de contactos.....	64
4.5.7	Interface para dar de alta un nuevo contacto.....	64

4.5.8	Menú de contratos .....	65
4.5.9	Interface de los contratos .....	62
4.5.10	Menú de solicitudes de servicio .....	66
4.5.11	Interface de solicitudes de servicio .....	66
4.5.12	Menú de actividades.....	67
4.5.13	Interface de actividades.....	67
4.6	ORGANIZACIÓN DE LOS ARCHIVOS EN EL SERVIDOR .....	69
4.6.1	Permisos y control de acceso a datos .....	70
4.6.2	VISTAS/RESPONSABILIDADES .....	73
4.6.3	Lógica de negocio .....	74
5	PRUEBAS.....	90
5.1	PRUEBAS DE INTEGRACIÓN .....	90
5.2	PRUEBAS DE INSTALACIÓN.....	91
5.3	PRUEBAS DE RECUPERACIÓN .....	91
5.4	PRUEBAS DE SEGURIDAD .....	92
5.5	VALIDACIONES .....	93
6	CONCLUSIONES .....	96
6.1	ALCANCE INICIAL DEL PROYECTO .....	96
6.2	RELACIÓN DE PROBLEMAS APARECIDOS Y NO IDENTIFICADOS PREVIAMENTE.....	98
6.3	CUMPLIMIENTO DE OBJETIVOS.....	98
6.4	MODIFICACIONES EN EL ALCANCE Y ALCANCE DEFINITIVO DEL PROYECTO .....	99
6.5	RETRASOS EN LA PLANIFICACIÓN .....	99
6.6	LÍNEAS DE AMPLIACIÓN.....	100
6.7	VALORACIÓN FINAL .....	100
7	BIBLIOGRAFÍA .....	102
7.1	FUENTES ELECTRÓNICAS .....	102
7.2	FUENTES BIBLIOGRÁFICAS .....	102
8	ANEXOS.....	104



# 1

---

# Introducción

# 1 INTRODUCCIÓN

## 1.1 PRESENTACIÓN DEL PROYECTO

Este proyecto ha estado desarrollado bajo el marco del convenio entre la empresa UNIT4 y la Escuela de Ingeniería de Informática de Sabadell de la Universidad Autónoma de Barcelona (UAB).

En esta empresa de gran experiencia sectorial se desarrolla, comercializa e implanta software empresarial y ofrece servicios profesionales a las organizaciones con el fin de gestionar sus negocios y adaptarse firmemente a los grandes avances tecnológicos de la actualidad con facilidad, rapidez y de una manera rentable.

Es una multinacional que divide su organización en un área de fabricación del producto (departamento de investigación I+D+i), consultoría y auditoría, soporte y marketing.

En el desarrollo de este proyecto se estudia la posibilidad de implantar una solución CRM para un club deportivo y consiste en alcanzar la integridad de la gestión a la vez de ser una óptima solución para conseguir el objetivo más importante: la fidelización del cliente y el mejor acercamiento a proveedores y colaboradores.

A través de los procesos de negocio requeridos se desarrolla una interfaz orientada a una aplicación de gestión que contendrá una parte de *call center* dónde se recogerán incidencias telefónicas y peticiones de clientes, en torno a las diferentes responsabilidades que correspondan con diferentes usuarios, por otro lado, la aplicación ayudará a ver las gestión de la relación con los clientes y esto ayuda a comunicar la información adecuada a la persona adecuada en el momento adecuado y a estudiar oportunidades de negocio.

La tecnología utilizada en el proyecto CRM es la herramienta de desarrollo Siebel de Oracle versión 8.0. y los lenguajes de programación eScript, Visual Basic, Java y SQL/PLSQL.

## 1.2 CONVENIO CON UNIT4

Desde principios del curso 2002/2003 la Escuela y el CCS (actualmente UNIT4) firmaron un convenio de colaboración con el objetivo de elaborar conjuntamente propuestas de formación en el área de las nuevas tecnologías de la información, abiertos a todos los estudiantes, y permitiendo que los alumnos de último curso pudieran desarrollar su proyecto final de carrera incorporándose en alguno de los departamentos de la empresa.

La duración de la estada en la empresa es de aproximadamente unas 560 horas repartidas a lo largo del curso, desde noviembre del 2010 hasta julio del 2011 en

horario de media jornada, permitiendo así el correcto seguimiento del resto de asignaturas que pueda tener el alumno.

Gracias a este convenio surge la oportunidad tanto de un desarrollo profesional para el alumno en su proyecto final de carrera, como la oportunidad de un nuevo descubrimiento en la etapa de la empresa, sirviendo de mutuo beneficio y crecimiento profesional.

### 1.3 LA EMPRESA

UNIT4 Ibérica es la filial española del grupo UNIT4, uno de los primeros fabricantes internacionales de software de gestión empresarial (ERP y CRM para pymes, RRHH, BI, CPM).

UNIT4 es una empresa global de software empresarial que crea, desarrolla, comercializa y ofrece una amplia gama de productos de software de gestión y servicios relacionados que permiten que los clientes estén actualizados; sus soluciones ofrecen un apoyo funcional de negocio tanto genérico como especializado, y benefician a todos los tipos de organizaciones del sector público y privado en el mundo entero. Además, algunas de ellas están específicamente centradas en las necesidades de sectores de mercado concretos abasteciendo a necesidades multisectoriales e internacionales.



**Figura 1: Presencia internacional/mundial de Unit4**

La empresa no comenzó siendo UNIT4 Ibérica, sino que tiene sus orígenes en CCS, el antiguo Centro de Cálculo de Sabadell fundado en 1963, la primera empresa española de software y servicios para la gestión de empresas. En el 2006 una compañía Holandesa llamada Agresso compró el 100% de las acciones de CCS, y la empresa pasó a formar parte de la filial española Agresso Spain, con el nombre de

CCS Agresso, compitiendo con otros dos grandes fabricantes de ERP como lo es SAP y Microsoft.

A principios del 2010 el conjunto de empresas de UNIT4 Agresso decide por temas de marketing, entre otros, que lo mejor era unificarlas todas, cogiendo un nombre común. Este es UNIT4. UNIT4 Ibérica desarrolla, comercializa y ofrece servicios a su completo y avanzado software de gestión (ERP y CRM) tanto para pequeñas, como grandes empresas, públicas y privadas en dos centros de I+D+i en España, dónde más de 100 investigadores certificados por el Ministerio de Industria y una plantilla especializada de 600 personas desarrollan sus soluciones para este software.

Algunos de estos software son UNIT4 ekon, UNIT4 Agresso Business World y Ready (auténtico ERP para pequeñas pymes), con lo que es el fabricante líder del país con una importante cartera de clientes tanto empresas, como Administraciones Públicas y la Sanidad, que confían en estas soluciones. La compañía también dispone de oficinas, centros de desarrollo y servicios en Portugal y Guinea Ecuatorial.

La empresa, internamente, se divide en diferentes áreas según la función que se realiza entre ellas Administración, RRHH, Research&Development, Marketing&Comercial, Soporte y Professional Services que realiza funciones de consultoría.

El sector que se dedica al desarrollo de software se llama fábrica, y dada la naturaleza específica del proyecto, éste se sitúa dentro del área de Professional Services programando en entorno Oracle Siebel dónde se responsabiliza a llevar a cabo la configuración de funcionalidades con Oracle Siebel CRM a partir del diseño técnico del cliente.



**Figura 2: Central Unit4 Barberà del Vallès**

## 1.4 OBJETIVOS

El objetivo principal de este proyecto es incrementar el grado de fidelización de los clientes mejorando la relación y optimizando los servicios ofrecidos. Partiremos de una base consolidada en un club deportivo con una cartera de clientes establecida y alcanzaremos los objetivos que se especifican acorde con la línea de negocio propuesta. Para conseguirlo se implementará un sistema de CRM<sup>1</sup> que satisfaga las necesidades del negocio. Esta implementación consiste en determinar las necesidades a cubrir y posteriormente determinar la configuración, parametrización y personalización óptima de una solución estándar de CRM.

En cuanto a los objetivos técnicos y personales del proyecto son:

1. El usuario final deberá disponer de una aplicación en la que podrá **analizar cada uno de los estados del proceso de cada gestión tramitada y estudiar las oportunidades de negocio para cada cliente**, con el fin de evitar su pérdida, es más, fortalecer su relación. Mejorando las relaciones con los clientes incrementamos su fidelización y podemos aplicar tareas iterativas para incentivar sus compras o contratación de servicios diferentes a los actuales a lo largo de las temporadas del club (up selling y cross selling).
2. Agilizar la gestión de los servicios de los clientes y usuarios de la aplicación. Mejoramos la estrategia comercial, puesto que obtendremos un sistema centralizado en el que deberemos disponer de un historial de movimientos con información fiable de cada socio para conocer sus preferencias y hábitos. De esta forma se llega a un acercamiento logrando un trato personalizado con el fin de fidelizar al socio y de incrementar su nivel de satisfacción.  
Por otro lado obtendremos información actual, y lograremos mejorar la comunicación entre socios, clientes, y empleados.
3. Obtener informes periódicos de la emisión de trámites, renovaciones, servicios contratados y bajas del sistema; y, lo más importante, saber en todo momento en qué estado está cada trámite o gestión. Existe un histórico de las actividades realizadas a los trámites solicitados y una gestión documental que recoge cada archivo adjunto necesario.
4. A nivel personal, el objetivo que me propongo es alcanzar un acercamiento a la tecnología Siebel para poder realizar la implantación de un proyecto en su totalidad y adquirir el conocimiento requerido en cada una de las fases que lo conforma, de igual forma que el aprendizaje de los conceptos técnicos durante el desarrollo.

---

<sup>1</sup> Gestión de las Relaciones con los Clientes / Customer Relationship Management

## 1.5 ESTADO ACTUAL

Actualmente vivimos uno de los avances tecnológicos en los que estamos obligados a adaptarnos continuamente al cambio y, cómo no, a la utilización de sistemas de información en nuestra vida cotidiana, para gestionar, planificar o simplemente organizar nuestro trabajo diario.

La situación inicial de la cual partimos trata de una aplicación desarrollada a medida para un club deportivo de fútbol que pretende ampliar sus servicios y sus instalaciones. Pretende ofrecer pistas adicionales de baloncesto, tenis y un servicio de parking.

La aplicación se rige bajo una base de datos diseñada en Microsoft Access 2003 con numerosas tablas, formularios y sus correspondientes subformularios para recogida e introducción de datos y las consultas más comunes utilizadas en su uso diario para tramitar las incidencias de los socios. Los proyectos son gestionados mediante hojas de cálculo Excel y desarrollos propios. Todo esto afecta en problemas casi continuos y en que tareas como buscar facturas, presupuestos, albaranes o cualquier otro documento relacionado con el proyecto, se convierte en algo casi traumático, ya que es complicado disponer de la información de forma ágil y rápida al no tenerla centralizada.

A lo largo de los años se ha ido incrementando su funcionalidad con la problemática de la escalabilidad en el sistema Access. Para solventar tal efecto ofimático se ha tomado la decisión de ejecutar un proyecto de migración del sistema actual a tecnología Oracle.

A nivel empresarial, existen múltiples herramientas de sistemas de información orientadas a la gestión de las operaciones de negocio, entre otras, mencionar CRM, BI, SCM, MRP, ERP, etc... que nos permiten gestionar la información de los clientes de manera que siempre esté actualizada y disponible al instante.

Tras analizar un estudio estadístico, se plantea la solución de implantar una herramienta CRM que nos proporcione un sistema proactivo que nos ofrezca tácticas de retención, estudios de nuevas oportunidades de negocio y mantener la información de socios y servicios evitando eliminarla. Pues es de bien sabido que hoy en día invertir tiempo y dinero en la búsqueda de nuevos clientes es menos rentable que mantener la cartera actual.

En resumen, ofrecemos una solución imprescindible desde el punto de vista de la eficiencia interna para la gestión financiera, para el control y mejora de sus procesos y para la consolidación de datos y toma de decisiones. Cubrimos todas las áreas del club deportivo e incrementaremos los beneficios considerablemente.

Para que una empresa pueda tener éxito, hay dos condiciones previas que debe cumplir: tiene que ser eficiente y competitiva en los procesos internos.

## 1.6 MOTIVACIÓN PERSONAL

La motivación principal de este proyecto es la de desarrollar una aplicación en una nueva tecnología creando un proyecto desde cero, en la que se requiere gran conocimiento de los conceptos de negocio inherentes a la solución del problema más que a técnicas avanzadas de programación. Será necesario también trabajar en la planificación y gestión del proyecto y simular el comportamiento del arranque de un proyecto de consultoría.

Además la actividad vertical a la que se quiere adaptar es un call center de un club deportivo en el que se realizan actividades de fútbol principalmente, y desde un principio hasta el final del desarrollo me he visto obligada a replantearme el modelo de datos numerosas veces.

Para finalizar, añadir que el hecho de llegar a tener el prisma de gestión de procesos de negocio en el que los objetos son componentes de negocio y las relaciones no se basan bajo tablas y relaciones únicamente, me obligó a abstraerme y adquirir una visión por capas que no tenía hasta ese momento. Informalmente, fue un cambio de "chip" para lo que había estudiado hasta entonces.



# 2

---

# Estudio Viabilidad

## 2 ESTUDIO DE VIABILIDAD

### 2.1 INTRODUCCIÓN

Se trata de un proyecto con el que se pretende implementar una herramienta de gestión de la relación con los clientes de un club deportivo futbolístico, que hará posible la agilidad en los trámites administrativos y en los servicios ofrecidos a los clientes y concretamente, en el ciclo de vida de un socio y de un abono. Esto se realizará a través de una aplicación CRM que configuraremos a partir de una herramienta Siebel partiendo de los requerimientos funcionales.

Al final de este apartado estudiaremos la viabilidad del proyecto.

#### 2.1.1 Tipología y palabras clave

**Tipología:** Configuración y desarrollo.

**Palabras clave:** CRM, customización, Siebel.

#### 2.1.2 Descripción

La gestión de la relación con los clientes y la necesidad de saber en cualquier momento la circunstancia de éstos, y de sus trámites administrativos es una alternativa atractiva para la mejora de la fidelización y los servicios ofrecidos en un grado óptimo.

Con una aplicación de gestión estratégica de negocio que permita facilitar las oportunidades y ofrecer un grado de satisfacción más alto por parte de los clientes, hará que la empresa que lo implemente destaque considerablemente.

La aplicación CRM que se pretende implantar de este proyecto permitirá llevar la gestión de socios de un club deportivo y la de los proveedores que forma parte de él con el fin de obtener el máximo rendimiento de los sistemas y los recursos utilizados.

#### 2.1.3 Objetivos del proyecto

1. Mejorar la gestión de los datos de los clientes y la comunicación en los servicios ofrecidos aumentando la productividad.
2. Obtener informes periódicos y una gestión documentada de todas las actividades realizadas en cada trámite solicitado. Se guarda un historial tanto de los movimientos como de los clientes.
3. Lograr un acercamiento y ofrecer un trato personalizado integrando el cliente en todas las áreas de la empresa, es decir, centralizando sus peticiones con mayor eficiencia.

4. Automatización de procesos con el fin de evitar errores y ofrecer una gestión documental que evitará la pérdida de documentos importantes.

Implantar una aplicación totalmente configurada y adaptada a las necesidades del cliente con la finalidad de mejorar y fidelizar la relación y los servicios ofrecidos en un grado óptimo.

La propuesta CRM que se pretende ofrecer consiste en un software de gestión que permitirá estudiar todas las posibles oportunidades de negocio de un club deportivo cubriendo la necesidad de saber en cualquier momento la circunstancia personalizada de cada cliente y de sus trámites administrativos para maximizar el rendimiento de los recursos utilizados y mantener la cartera de clientes.

### 2.1.4 Definiciones, acrónimos y abreviaciones

**CRM** [Customer Relationship Management]: estrategia de negocios centralizada en identificar y administrar las relaciones en las cuentas más valiosas de una empresa con el fin de mejorar la efectividad sobre los clientes e intentar fidelizarlos.

**Siebel**: herramienta de gestión CRM que permite administrar todos los puntos de contacto del cliente.

**Oracle**: sistema de gestión de bases de datos objeto-relacional bajo arquitectura cliente/servidor.

### 2.1.5 Perfiles de usuario

En el desarrollo del proyecto hay dos partes implicadas, por una parte el desarrollador de la aplicación, y por otra el que realiza la revisión y supervisión.

Por otro lado, una vez esté desarrollado y se haya realizado la puesta en marcha se pueden diferenciar tres tipos de usuarios con perfiles diferentes:

**Usuario call center:** gestión de solicitudes, tramitación de altas, bajas y otras gestiones que el socio necesite, como cambios en el contrato de los socios, a efectos prácticos se encargará de la recogida de solicitudes entrantes del sistema. Se encarga de la comunicación con el cliente y se asegura de que los envíos lleguen en la fecha prevista.

Puesto que la aplicación pretende ofrecer un sistema centralizado que recoge todas las incidencias solicitadas, el call center se considera como el perfil de primer contacto con el cliente, que tendrá la potestad de crear las actividades oportunas y ceder la responsabilidad al usuario back office.

**Usuario back office:** realiza las actividades de consulta de la información, da soporte al call center, se encarga de realizar la mayor parte de las consultas y modificaciones a la base de datos. Tiene la responsabilidad de realizar los trámites del sistema y no necesita estar en contacto con la persona que solicita el trámite. La comunicación es bidireccional con el call center.

**Administrador:** la persona que recibirá a priori una formación para conocer el entorno y que será la responsable de llevar el control total sobre la aplicación y podrá modificar los parámetros de ésta en función de las necesidades del usuario, la conclusión es que tendrá potestad absoluta. Podrá administrar usuarios, productos y asignar las responsabilidades adecuadas a las diferentes vistas de la interface. Además se responsabilizará del mantenimiento de la base de datos.

## 2.1.6 Referencias

La customización de herramientas CRM de Siebel se regirá a partir de la **Ley Orgánica de Protección de Datos (LOPD) que obliga a todas las personas, empresas y organismos, tanto privados como públicos, que traten con datos de carácter personal, a cumplir unas normas estrictas y a aplicar ciertas medidas de seguridad para almacenar los datos personales nombrados, garantizando al titular de estas su libertad y sus derechos fundamentales en honor a la intimidad personal y familiar.**

Esto significa que el administrador de la base de datos del sistema no hará mal uso de los datos personales ni privados que se almacenen siendo totalmente consciente de la privacidad.

Por otro lado también se rige la Ley de Servicios de la Sociedad de la Información y comercio electrónico (**LSSI/LSSICE**)<sup>2</sup>:

---

<sup>2</sup> **LSSI:** Ley de Servicios de la Sociedad de la Información.  
<http://www.mityc.es/dgdsi/Lssi/Documents/ltriptico.pdf>

### 2.1.7 Producto y documentación del proyecto

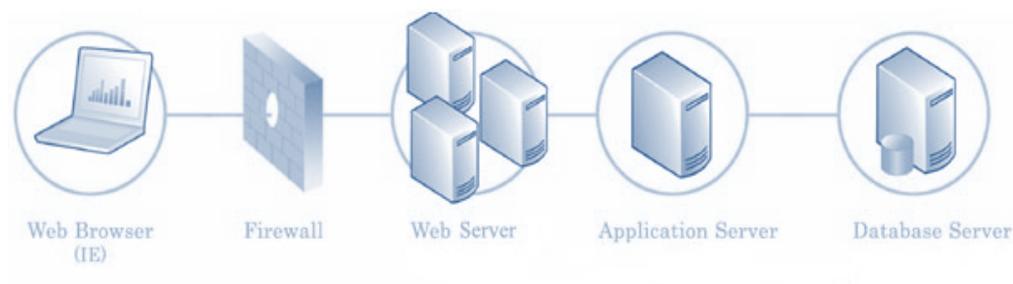
- Se entregará los archivos srf que contienen los compilados de la aplicación desarrollada junto a otros archivos de configuración.
- Se elaborará una memoria del proyecto que contendrá un estudio de la viabilidad del proyecto junto a los diseños técnicos y funcionales.
- Se adjunta la propuesta de modelo de datos inicial y con la lista de requerimientos del club deportivo y el alcance.

## 2.2 ESTUDIO DE LA SITUACIÓN ACTUAL

### 2.2.1 Contexto

Actualmente existen muchas aplicaciones y programas que ayudan en la gestión administrativa y social de un club deportivo. De igual forma, el mantenimiento y supervisión del volumen de información que se mueve diariamente en cualquier trámite implica una gran base de datos detrás. Aún así la mayoría de estas aplicaciones no disfrutan de la accesibilidad y escalabilidad necesarias que ayudaría a una actualización fiable y ágil, o bien, el acceso de un servicio que se necesite, y lo más importante, disponer en cualquier momento de información actualizada.

### 2.2.2 Lógica del sistema y arquitectura



El sistema se rige bajo una arquitectura web que consiste en un cliente web (navegador), un servidor web, un gateway name server y una empresa que puede soportar uno o más servidores Siebel.

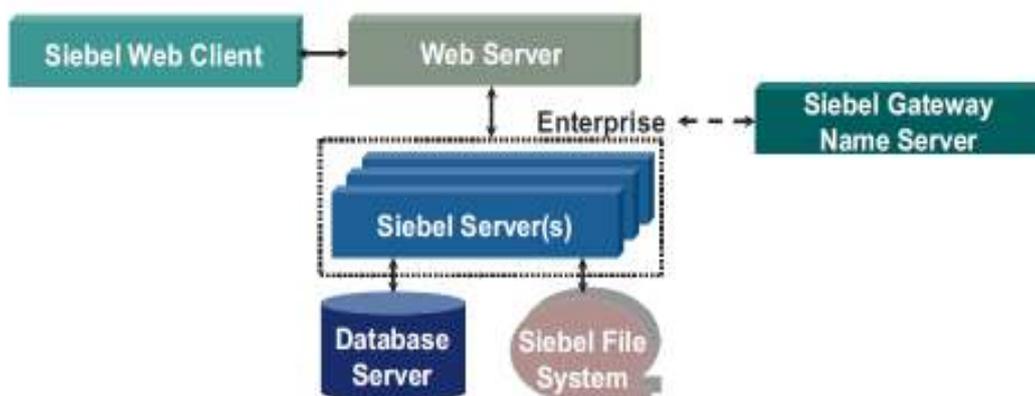


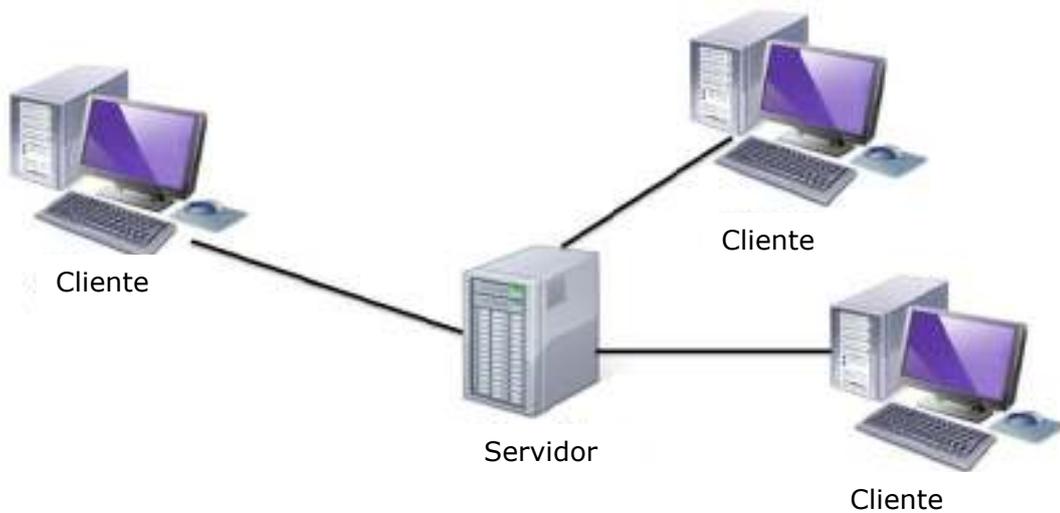
Figura 4 Arquitectura

### 2.2.3 Entorno físico de trabajo

La aplicación Siebel que se pretende instalar consta de un cliente (browser) conectado a través de un navegador compatible, en este caso Internet Explorer 7.0, y un servidor con las siguientes especificaciones:

SERVIDOR	CLIENTE
Memoria RAM: 3 GB	Memoria RAM: 3 GB
Procesador: 2,66 GHz	Procesador Core 2 Duo T8100 2,10 GHz
Windows Server 2003 Enterprise Edition Service Pack 2	Windows XP Professional Service Pack 3

**Figura 5 Especificaciones Servidor/Cliente**



**Figura 6 Clientes/Servidor**

## 2.2.4 Diagnóstico del sistema

Existe una serie de problemas que podemos encontrar en la aplicación, entre ellos:

- Que en un momento dado se caiga algún servicio de Siebel ya sea servidor o del *gateway*, y la aplicación no funcione vía web.
- Otro problema originado por la caída de un servicio viene relacionado con la base de datos, Oracle. Suponiendo que éste caiga, no podremos acceder a realizar las consultas y *queries* para devolver información.
- Que se produzca un error de comunicaciones y se pierda la conexión con el servidor, con lo que la aplicación dejaría de funcionar para todos los clientes.

Por lo tanto sería conveniente en un futuro tomar algunas medidas considerando los posibles errores arriba comentados, con el fin de mejorar el rendimiento del sistema.

- Optimización del hardware del servidor para evitar fallos de servicios con tal de garantizar el buen funcionamiento y rendimiento un 100% del tiempo.
- Intentar adaptar la tecnología e infraestructuras más avanzadas

## 2.3 RECURSOS DEL PROYECTO

### 2.3.1 Descripción de recursos disponibles

Los recursos que se requieren para el desarrollo de este proyecto se dividen en humanos y materiales, por un lado los recursos de los que se disponen son los humanos y los necesarios serían los recursos materiales para poder llevarlo a cabo.

<b>Recursos Humanos</b>		<b>Horas dedicadas</b>	<b>Tarifa [Ingreso]</b>	<b>Beneficio [I-C]</b>
<i>Director de proyecto</i>		<i>38,4 horas</i>	<i>65€/hora</i>	<i>49,67 €/h</i>
<i>Consultor</i>		<i>308 horas</i>	<i>45 €/hora</i>	<i>32,39 €/h</i>
<i>Técnico</i>		<i>352 horas</i>	<i>35 €/hora</i>	<i>25,07 €/h</i>
<i>Comercial</i>		<i>4 horas</i>	<i>35 €/hora</i>	<i>26,22 €/h</i>
<i>Proyectista</i>		<i>649 horas</i>	<i>10 €/hora</i>	<i>5,3 €/h</i>
<b>Recursos Materiales</b>	<b>Coste</b>			
<i>Servidor</i>	<i>2000 €</i>			
<i>Gateway Server</i>	<i>3000 €</i>			
<i>Proyector</i>	<i>400 €</i>			

**Figura 7 Recursos disponibles y necesarios**

### 2.3.2 Restricciones del sistema

1. La aplicación Siebel 8.0 del cliente se ha de conectar al servidor a través de un navegador o browser compatible. En este caso se trata de Internet Explorer 7.0. Mozilla Firefox usa un modo de interactividad bajo y hace que los botones y controles no funcionen correctamente.
2. En cuanto a la compatibilidad del sistema operativo no hay una restricción específica, no obstante, mencionar que la mayoría de clientes que se conectan al sistema utilizan un sistema operativo Windows XP profesional, o Windows 7.
3. Se prescinde directamente de las comunicaciones a través de una red local de 10/100/1000 bits/seg que conecte todos los sistemas de la infraestructura.

### 2.3.3 Evaluación de riesgos y posibles soluciones

A continuación se contempla una serie de riesgos a tener en cuenta durante el desarrollo del proyecto, y, se indican las posibles medidas que podrían adoptarse para evitarlos (soluciones):

**R1:** Falta de alguna tarea necesaria no contemplada en el plan de proyecto.

*GAP ANALYSIS* (riesgo) gestión del cambio adecuado para cubrir desviaciones: herramienta de negocio utilizada para comparar el rendimiento actual con el potencial (análisis de desviación y modificación de hábitos para la mejora de los recursos)

**R2:** Planificación temporal optimista: No se acaba en la fecha prevista, no se cumplen los objetivos y afecta atrasando el tiempo de finalización y a la calidad del proyecto causando pérdidas económicas.

**R3:** Presupuesto poco ajustado que afectaría a la Viabilidad y al Plan de proyecto.

Mayor dedicación para solventar la inexperiencia inicial a la hora de desarrollar el proyecto.

**R4:** Abandono del proyecto antes de la finalización: en cualquier fase. Pérdidas económicas, frustración.

Definir calendario reuniones de seguimiento quincenales con el director de proyecto y los usuarios.

**R5:** El posible rechazo de la aceptación del proyecto por parte del usuario.

Formación a la mayoría de usuarios e involucración de éstos en el proceso de definición del proyecto.

### 2.3.4 Recursos

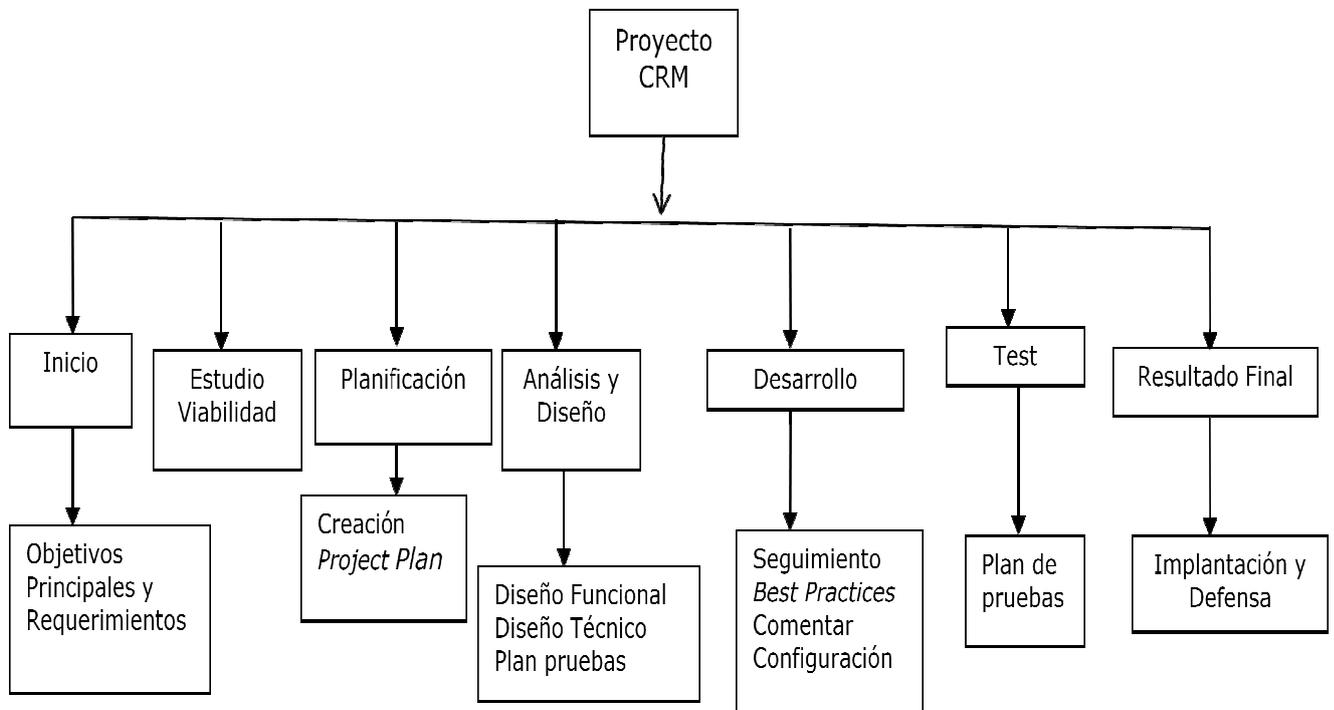
- Recursos materiales:  
Hardware: Servidor, Gateway Server, Ordenador Portátil,  
Software: Microsoft Office Word 2007 Microsoft Office Project 2007,  
Windows Xp SP3, Windows 2003 Server,  
Imagen: Proyector.
  
- Recursos humanos: Projectista, director de proyecto, consultor  
(analista/programador) y técnico.

Es importante mencionar que puesto que se trata de una simulación de un proyecto real, en este caso todos los recursos humanos que han intervenido a excepción del director de proyecto, han sido la misma persona, que será la responsable de la implementación total del proyecto, desde la viabilidad hasta la realización del juego de pruebas.

## 2.4 ORGANIZACIÓN DEL PROYECTO

A continuación se definen las principales fases del Proyecto Siebel que deben respetarse para optimizar el esfuerzo/recursos y alcanzar de este modo un resultado óptimo y lo más ajustado a los objetivos iniciales:

1. **Iniciación:** Formación Siebel, requerimientos básicos y objetivos principales del proyecto.
2. **Estudio Viabilidad:** Estudio de la situación actual del sistema, usuarios participantes y diagnóstico, restricciones económicas.  
En esta fase se estiman costes, tiempos, riesgos y alternativas y se determina la viabilidad del proyecto o no en función de la envergadura del sistema.
3. **Planificación:** Gestión y distribución de tareas y sub-tareas con los recursos materiales y humanos dedicados en ellas sobre un diagrama de tiempo. La herramienta utilizada es Microsoft Project.
4. **Análisis:** Especificación del sistema delimitando su alcance: casos de uso, diagramas de contexto, modelo de datos, tablas y relaciones.
5. **Diseño.**
6. **Desarrollo e implantación de la aplicación:** Formación necesaria a cada uno de los usuarios que formarán parte de la implantación en función de sus perfiles y preparación de pruebas de implantación.
7. **Testeo y bug fixing.** Para garantizar que todos los requerimientos han sido satisfechos es necesario el planteamiento de un plan de pruebas en el que todos ellos sean testeados.
  - a. **Pruebas de implantación para comprobar cualquier detalle de diseño interno y comportamiento del sistema bajo las condiciones más extremas:**
    - Pruebas de unidad
    - Pruebas de integración
    - Pruebas de sistema
  - b. **Pruebas de aceptación:** Realizadas por y para los usuarios y con el objetivo de validar formalmente que el sistema se ajusta a sus necesidades: UAT (user acceptance testing): proceso para obtener la confirmación de que un sistema cumple las expectativas requeridas. Para ello los usuarios de sistema realizan las pruebas con los desarrolladores.
  - c. **Corrección de los errores encontrados**
8. **Conclusiones y defensa del proyecto:** cierre del proyecto.



**Figura 8 Organización del proyecto**

## 2.5 PLANIFICACIÓN DEL PROYECTO (MS PROJECT)

### 2.5.1 Etapas

1. Iniciación (diagnóstico y preparación)
2. Estudio de viabilidad del proyecto (planificación)
3. Análisis
4. Diseño.
5. Desarrollo de la aplicación.
6. Testeo (pruebas de aceptación para validar que el sistema cumple con el funcionamiento esperado para el usuario final y documentación de las pruebas).
7. Conclusiones y defensa del proyecto.

### 2.5.2 Cronograma inicial

<b>Tarea</b>	<b>Descripción</b>	<b>Horas</b>
<b>1</b>	Formación	40
<b>2</b>	Investigación y Viabilidad	60
<b>3</b>	Análisis	60
<b>4</b>	Diseño	110
<b>5</b>	Desarrollo de la aplicación	210
<b>6</b>	Pruebas unitarias	40
<b>7</b>	Documentación	40
<b>8</b>	Defensa del proyecto	10
<b>Total</b>		<b>560</b>

**Figura 9 Cronograma inicial**

## 2.6 PRESUPUESTO

<b>COSTES DIRECTOS</b>					
<b>Recursos Humanos</b>	<b>Horas dedicadas</b>	<b>Tarifa [Ingreso]</b>	<b>Ingresos totales</b>	<b>Coste anual (U4) Salario-S.S-Dieta</b>	<b>Costes totales [horas*coste]</b>
Director de proyecto	38,4 horas	65€/hora	2496 €	31,36 €/hora	1204 €
Consultor	308 horas	45 €/hora	13860 €	26,27 €/hora	8091 €
Técnico	352 horas	35 €/hora	12320 €	21,17 €/hora	7452 €
Comercial	4 horas	35 €/hora	140 €	19 €/hora	76 €
Proyectista	649 horas	4,70 €/hora	3050 €	4,70 €/hora	3050 €
			<b>31866 €</b>	<b>19873 €</b>	
<b>Beneficio = Ingreso – Coste</b>				<b>11993 €</b>	
<b>Recursos Materiales</b>				<b>Coste</b>	
Servidor				2000 €	
Gateway Server				3000 €	
Ordenador portátil				600 €	
Proyector				400 €	
<b>Total coste materiales</b>				<b>6000 €</b>	
<b>COSTES INDIRECTOS</b>					
Alquiler de la oficina (incluido el coste de electricidad, agua, y climatización)					
Servicio de limpieza y mantenimiento					
Material oficina,					
Salario personal administrativo					
Salario personal recursos humanos					

Figura 10 Costes, ingresos y beneficios

## 2.7 POSIBLES ALTERNATIVAS

### 2.7.1 Alternativa #1

Adquirir una solución CRM diferente, como puede ser por ejemplo Salesforce:  
<http://www.salesforce.com/>



Figura 11 Salesforce CRM

Gestión de las ventas, del servicio de atención al cliente, y cualquier tipo de aplicaciones comerciales a través de un navegador de Internet basándose en la arquitectura multiusuario.

### 2.7.2 Alternativa #2

Adquirir otra posible solución CRM diferente y de libre distribución, como puede ser SUGARCRM: <http://www.sugarcrm.com/crm/>. Una aplicación orientada a diferentes tipos de negocios diseñada para facilitar la gestión de ventas, oportunidades, contactos de negocio,...



Figura 12 SugarCRM

### 2.7.3 Alternativa #3

Otra posible estrategia de negocio basada en la optimización de procesos de servicio al cliente, es SAP CRM que ofrece soporte a los departamentos de atención al cliente, y en general tiene una adaptabilidad eficiente a los cambios del sector: <http://www.sap.com/solutions/business-suite/crm/index.epx>



Figura 13 Sap CRM software

## 2.7.4 Evaluación de alternativas

Para realizar un completo análisis comparativo de las alternativas deberíamos conocer los costes asociados a las licencias, si existen costes por servicios adicionales, los costes de alojamiento cuando sea necesario y los de mantenimiento para poder calcular tanto el CAPEX (Costes de capital o inversión a realizar) como el OPEX (Gastos de operación o que es lo que cuesta mantenerlo operativo), comparando estos indicadores financieros con las prestaciones y necesidades del negocio a satisfacer se debería escoger alguna de las alternativas. En este caso no se ha realizado dado que al ser un proyecto en empresa y disponer de la herramienta Oracle-Siebel CRM la decisión era obvia.

## 2.8 CONCLUSIONES

Analizando el presupuesto estimado que, a priori parece ajustado, alcanzable y asequible para el desarrollo de este proyecto que parte desde cero, se puede considerar que es un proyecto sin fecha de caducidad y su rendimiento a largo plazo puede ser muy alto ya que la mejor estrategia de negocio es siempre ofrecer un grado de satisfacción fidelizando las relaciones con tus clientes y teniendo una trazabilidad de en qué estado está cada uno de sus trámites y por finalizar analizando los pros y contras de éste, llegamos a la conclusión de que es viable.

### **A favor:**

- Adaptabilidad vertical al modelo de negocio deportivo.
- Mejora del sistema de comunicación interno entre usuarios de la empresa y externo para la relación cliente-club.
- Coste asequible y rendimiento a largo plazo.
- Posibilidad de mejora con nuevas prestaciones en un futuro (escalable y concurrente) con facilidad de uso y gestión documental.
- Optimización del retorno de la inversión en los balances de gestión.
- Calidad de la entrada datos reduciendo el margen de error humano.
- Aumento del número de socios.
- Agilidad en los servicios ahorrando tiempo a los empleados.
- Reduce la preocupación del informático a la hora de corregir errores puesto que se trata de un sistema totalmente centralizado en el servidor.

### **En contra:**

- La configuración de los datos del negocio afectan al rendimiento de la aplicación CRM.
- Coste de mantenimiento una vez esté implantado.
- Necesidad de un breve período de formación inicial.

# 3

---

# Análisis

## 3 ANÁLISIS

### 3.1 INTRODUCCIÓN

A partir de la entrevista con el director del club deportivo se priorizan los requerimientos básicos adecuados a la aplicación deseada y se han seleccionado las funcionalidades extraídas del documento "Requerimientos - CRM Club Deportivo" como alcance de este proyecto. Cada funcionalidad ha sido enfocada bajo un prisma Siebel para que la implementación del proyecto pueda resultar en un paquete completo, dentro de sus limitaciones.

### 3.2 REQUERIMIENTOS FUNCIONALES

Con los requerimientos funcionales describiremos el comportamiento deseado de la aplicación que se debe desarrollar y configurar generando una lista detallada con cada una de las funcionalidades:

Mencionar que la mayoría de requerimientos funcionales se pueden considerar como trámites dentro de un escenario cíclico como es el proceso de la vida de un socio.

Requerimientos	Módulo	Descripción	Criticidad
RF1: Solicitud de alta / nuevo carnet de socio	Socio Empleado	Creación de nuevos contactos con los datos personales y bancarios correspondientes.	Media
RF2: Validación de datos	Socio	Integridad de los datos de los clientes para evitar duplicados.	Alta
RF3: Aceptación	Empleado	Una vez validado el nuevo socio se le asignará una clave numérica o código.	Alta
RF4.1: Actualización foto del carnet de socio	Empleado Socio Solicitudes de servicio Contratos	Solicitud de cambio de foto. Realizar foto nueva y eliminación de la antigua en la base de datos.	Media
RF4.2: Duplicado abono/carnet de socio	Socio Empleado Solicitud de servicio Contrato	Solicitud de duplicado de abono ó carnet de socio a través de una previa validación.	Media
RF4.3: Cambio de asiento	Socio Empleado Solicitud de servicio Contrato	Solicitud de cambio de asiento. de abono ó carnet de socio a través de una previa validación.	Media
RF4.4: Carnet y diploma socio difunto	Socio Empleado Solicitud de servicio Actividad Contrato	Solicitud de carnet de socio difunto. Validación de certificado y cese de carnet a familiares.	Media

RF4.5:Renovaciones abonos/carnets	Socio Empleado Solicitud de servicio Actividad Contrato	Solicitud de renovación de abono o carnet. Verificación del socio y de pago al corriente y activación del trámite.	Media
RF4.6:Registro bajas abono	Socio Empleado Solicitud de servicio Actividad Contrato	Solicitud de baja de un abono. Verificación del socio y pagos. Formalización de baja y encuesta de satisfacción.	Alta
RF4.7: Petición envío revista	Socio Empleado Solicitud de servicio Actividad Contrato	Solicitud de envío de revista. Validación de socio. Activación servicio envíos.	Media
RF5: Registro bajas usuarios	Socio Empleado Solicitud de servicio Socio Actividad Contrato	Solicitud de baja de un usuario. Verificación del socio y pagos. Formalización de baja y encuesta de satisfacción	Alta
RF6: Solicitud abono	Abono Contrato Socio Empleado Solicitud de servicio Actividad	Generación de nuevos abonos	Alta

**Figura 14 Requerimientos funcionales**

## 3.3 REQUERIMIENTOS NO FUNCIONALES

Las restricciones que nos impone el club deportivo a la hora de desarrollar la aplicación son las siguientes:

### 3.3.1 Técnicos

Como único requerimiento no funcional se considerará la utilización de la plataforma Siebel.

<b>Plataforma Siebel 8.0</b>	
Siebel Tools Oracle	Herramienta de desarrollo para modificar y almacenar los ficheros que controlarán el comportamiento de la presentación del negocio.
Siebel Enterprise	Cliente servidor web.
Conexión red local 10/100/1000 bits/segundo fiable	
Servidor de aplicaciones de Siebel	
Servidor web IIS (Internet Information Services) que ofrece servicios HTTP/S	
Gateway Server	
Base de datos del servidor Oracle(Database server)	

**Figura 15** Requerimientos no funcionales técnicos

### 3.3.2 De usabilidad

Consistirá en un programa sencillo con una interface Siebel amigable y totalmente adaptada a las necesidades de un club deportivo, con lo que este punto no será excesivamente preocupante ya que se realizará un breve período de formación a los usuarios. Por otro lado se ha de tener en cuenta que existirá un mantenimiento de la aplicación una vez esté implantada.

### 3.3.3 De seguridad

El acceso al programa será controlado por el administrador y éste creará los perfiles de usuario en función del rol de cada uno, con la accesibilidad correspondiente, por ello se han diferenciado dos tipos de usuarios que gestionarán trámites: un usuario *call center* como toma de primer contacto y *back office* para la realización de verificaciones y gestiones.

En la interface de usuarios final se dispondrán de diferentes vistas que tendrán asignadas responsabilidades diferentes en función de las actividades de cada usuario. No obstante, será el usuario administrador el que tenga acceso a toda la aplicación y el que asigne las vistas a las responsabilidades adecuadas, como también administre los usuarios y los productos.

### 3.3.4 Espacio de almacenamiento: base de datos

El servidor de la base de datos principal estará dotado de un amplio disco duro en el que tendrá suficiente cabida para guardar todos los datos relacionados con los clientes y para realizar las copias de backup.

- ❑ **Lenguajes de desarrollo:** PL/SQL, eScript lenguaje propietario de SIEBEL.
- ❑ **Base de datos:** Oracle.
- ❑ **Sistemas Operativos:** se garantizará el funcionamiento en los sistemas operativos MS Windows y HP-UX (en cualquiera de sus distribuciones).
- ❑ **Servidor de aplicaciones:** Oracle Application Server (servidor lógico).  
En cualquier caso, deberá ser lo más independiente posible del entorno Hardware y del Sistema Operativo que la soporte.
- ❑ **Oracle Application Server (OAS),** como servidor de aplicaciones para los distintos entornos.  
Ofrece funciones de carga de trabajo distribuidas para optimizar el rendimiento.

### 3.3.5 Corporativos

- Estilo de la aplicación.
- Color azul de la interface.
- Formato de la aplicación.
- Fuente: estilo, color, tamaño.
- Diseño de la aplicación, pantallas, vistas y logo.
- Técnico 90 % disponible en la aplicación. ( 90 % disponible)
- Mantenimiento.
- Seguridad.
- Escalabilidad.
- Idioma de la aplicación.
- Coste.

### 3.3.6 Académicos

- Requerimientos personales de aprendizaje para el desarrollador
- Curso de formación inicial de *Siebel 8.1.x Core Consultant*
  - a. *Siebel 8.1.x Installation and System Administration*
  - b. *Siebel 8.1.x Tools*
  - c. *Siebel 8.1.x Business Automation*

### 3.3.7 Catalogación i priorización de los requerimientos.

Requisitos Funcionales	RF1	RF2	RF3	RF4.1	RF4.2	RF4.3	RF4.4	RF4.5	RF4.6	RF4.7	RF5	RF6
Esencial	X	X	X	X	X	X	X		X	X	X	X
Opcional												
Condicional								X				

Requisitos No Funcionales	RNF1	RNF2	RNF3	RNF4	RNF5	RNF6
Esencial	X		X	X		X
Opcional		X				
Condicional					X	

Figura 16 Priorización requerimientos funcionales

	RF1	RF2	RF3	RF4.1	RF4.2	RF4.3	RF4.4	RF4.5	RF4.6	RF4.7	RF5	RF6
O1		X						X	X	X		
O2			X	X		X	X				X	
O3					X							X
O4	X											

	RNF1	RNF2	RNF3	RNF4	RNF5	RNF6
O1				X		X
O2	X					
O3		X			X	
O4			X			

Figura 17 Relación entre los requerimientos y los objetivos

### 3.4 CASOS DE USO

En esta etapa se realizan los documentos UML establecidos según las especificaciones funcionales previamente seleccionadas.

La lista de casos de uso es la siguiente:

**RF1: Caso de uso Alta de socio.**

**RF2: Cado de uso Validación de los datos del socio.**

**RF3: Caso de uso Aceptación.**

Estos tres casos de uso se pueden considerar como uno que trata el alta de un nuevo socio.

**Actores:** cliente, empleado *call center* (cc), empleado *back office* (bo), socio.

**Descripción:** un cliente nuevo se quiere dar de alta en el club deportivo y se pone en contacto con el empleado *call center* que creará una actividad al *back office* para que valide los datos. Una vez validados se le dará de alta y se le asignará un número de socio.

1. El cliente solicita un alta en el club.
2. El empleado call center gestiona la solicitud de alta del futuro socio creando la petición correspondiente y dejándola en estado pendiente.
3. El empleado back office tramita la solicitud validando que los datos sean reales y el cliente pasa a un estado "activo" siendo ya considerado como nuevo socio.
4. El empleado call center cumplimenta la información personal y bancaria del cliente y realiza la validación aceptando la gestión del alta solicitada y asignándole un número de socio identificativo.
5. El nuevo socio queda activo en el sistema.

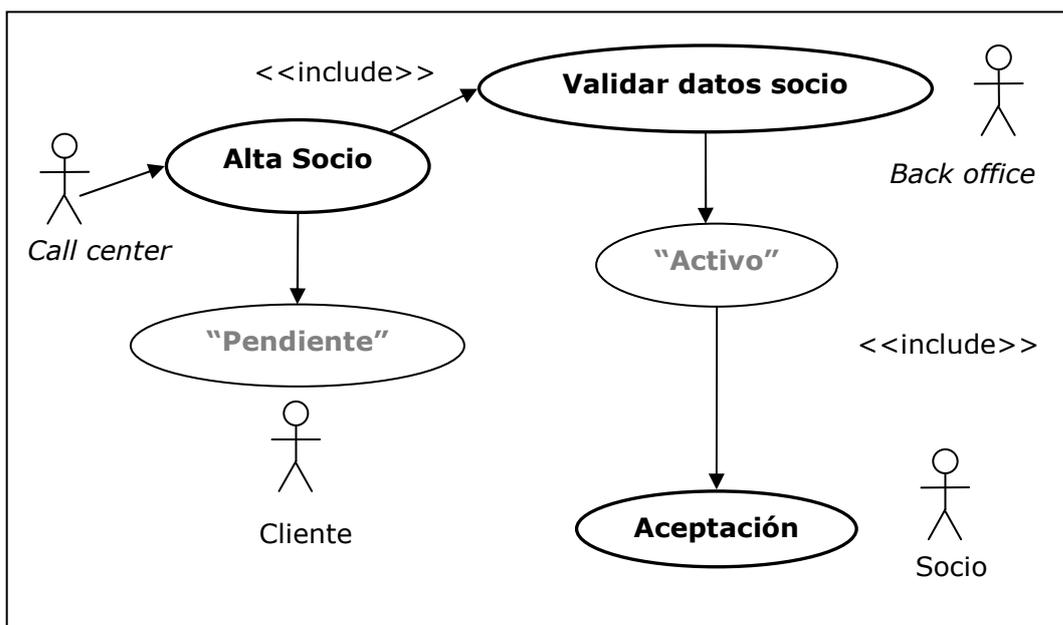


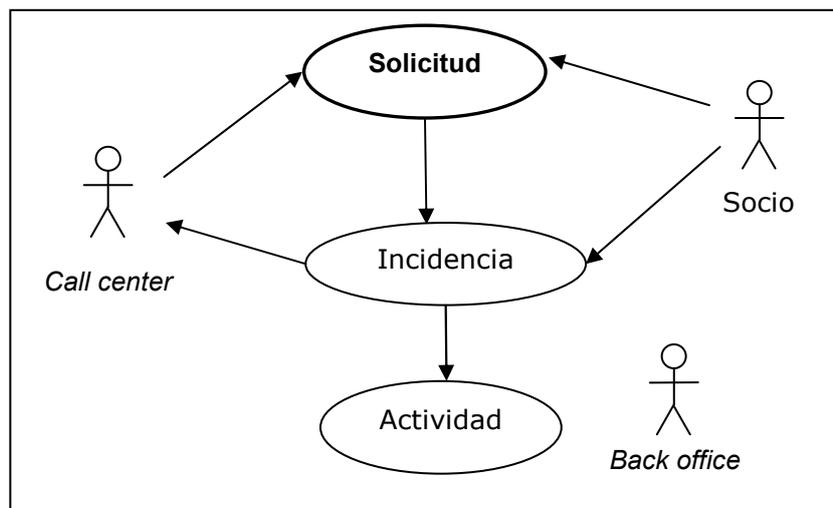
Figura 18 Casos de uso: alta/validación/aceptación

**Curso alternativo:** Si el usuario introduce un número de DNI incorrecto el sistema avisará y no dejará guardar el registro. Por otro lado si se intenta introducir un usuario ya existente también está controlado, pues es la forma de evitar duplicados.

#### **RF4: Caso de uso Solicitudes de servicio**

**Actores:** socio, *call center*, *back office*.

**Descripción:** Todos los posibles trámites que un socio pueda solicitar son considerados como solicitudes de servicio. De hecho, cualquier petición es centralizada bajo una creación de ellas. Diferenciaremos entre un listado de solicitudes que listaremos más abajo.



**Figura 19 Caso de uso solicitudes**

La lista de solicitudes que se tendrán en cuenta en este proyecto son los siguientes:

##### **RF4.1: Actualización de la foto del carnet de socio**

1. El socio solicita la actualización de la foto del carnet al empleado *call center*.
2. El *back office* verifica la acreditación del socio y realiza la nueva fotografía.
3. Elimina la fotografía antigua y la actualiza.

##### **RF4.2: Duplicado de abono(s)/carnet de socio**

1. El socio solicita el duplicado de un abono/carnet al empleado *call center*
2. El *back office* acredita la validación de los datos del socio y de los pagos al corriente.
3. Anula el abono anterior y solicita una denuncia en caso de robo o pérdida del carnet.
4. Realiza la actualización del contrato y el duplicado.

5. El *call center* pone en comunicado al socio en el momento que esté finalizado el trámite.

#### **RF4.3: Cambio de localidad/asiento**

1. El socio solicita un cambio de localidad.
2. El *call center* confirma la titularidad del socio del club y crea una actividad de verificación para el empleado *call center*.
3. Se verifican los datos del cliente y que los pagos estén al corriente.
4. El *call center* entrega la solicitud de cambio de nombre al socio. Se buscarán asientos libres y se ofrecerán.
5. Una vez elegido el nuevo asiento el *back office* se dispone a actualizar el contrato.

#### **RF4.4: Carnet y diploma de socio difunto**

1. Un familiar del socio solicita el carnet y diploma del socio difunto. El *back office* verifica el certificado de defunción del socio.
2. Una vez validados el *call center* entrega el carnet y diploma junto a una carta de condolencia.
3. El *back office* elimina el estado "activo" del socio difunto y registra sus datos en una tabla auxiliar de socios inactivos.

#### **RF4.5: Renovaciones de abonos/carnets**

1. El sistema alerta al empleado de la fecha de renovación de los abonos 15 días antes.
2. El *call center* se pone en contacto con el socio para ofrecer una renovación automática de su abono/ carnet, o bien ofertarle otros tipos de abonos más amplios.
3. El socio autoriza la renovación automática del abono enviando acreditación de número de socio y fotocopia del DNI.
4. El empleado *back office* renueva el abono del socio una temporada más actualizando el contrato pertinente.

#### **RF4.6: Registro de bajas de abonos**

1. El cliente solicita la baja del abono.
2. El *back office* del club valida el socio y verifica el abono contratado.
3. El *call center* formaliza la baja entregando al socio una solicitud de baja y una encuesta de satisfacción para cubrir necesidades futuras.
4. El *back office* gestiona la baja del abono actualizando el contrato del socio.

#### **RF4.7: Petición de envío de revista**

1. El cliente solicita el envío de la revista mensual en su domicilio al empleado *call center*.
2. El *back office* verifica la acreditación del socio y mira si los pagos están al corriente en el servidor.

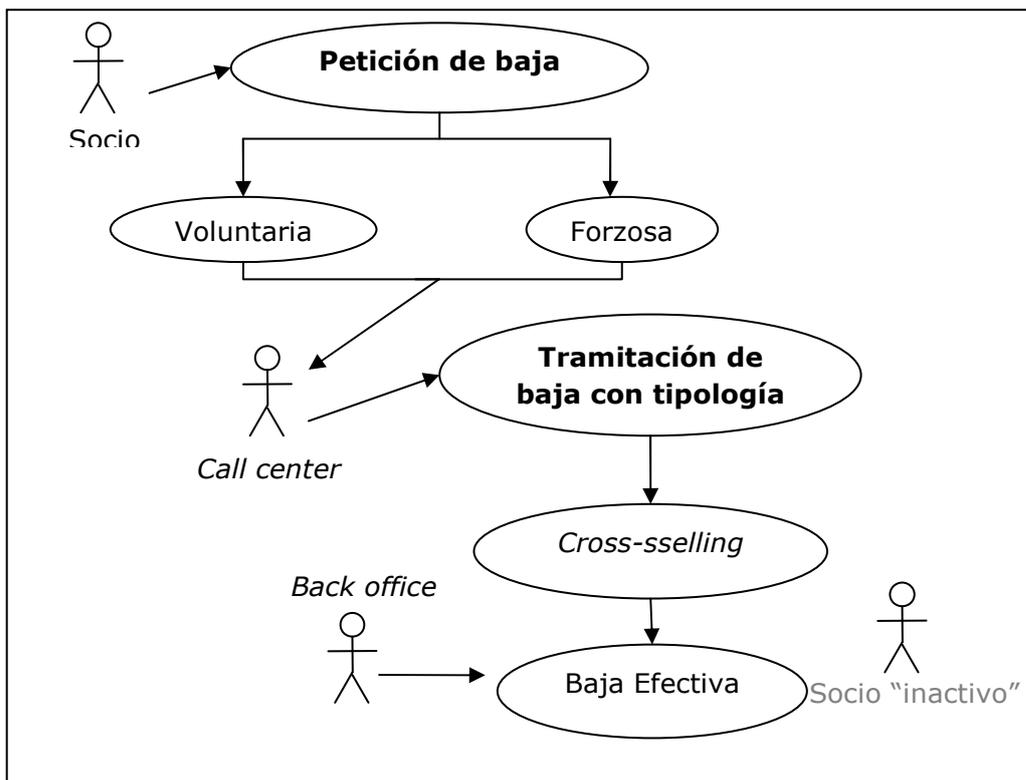
3. El *call center* formaliza la petición comprobando que el domicilio sea el que figura en el contrato y gestiona el alta del servicio.
4. El socio recibirá la revista de forma periódica en su domicilio.

### **RF5: Caso de uso Registro de Bajas**

**Actores:** socio, *call center*, *back office*.

**Descripción:** Registro de una baja de un socio.

1. El socio solicita la baja del club rellenando la solicitud de bajas pertinente.
2. El *back office* verifica la acreditación del socio y mira si los pagos están al corriente en el servidor.
3. El *call center* del club gestiona la baja y entrega una encuesta de satisfacción para registrar el motivo e intentar ofrecerle un servicio adicional con tal de retenerlo.
4. El *back office* se encargará de actualizar los datos de los contratos del socio y de cambiar el estado "activo" para formar parte de una tabla auxiliar de socios inactivos.



**Figura 20** Caso de uso registro de baja

#### **RF5.1: Registro de tipologías de Bajas**

El socio solicita la baja del club mediante el formulario de bajas.

El empleado asigna la tipología de la baja correspondiente (Voluntaria/Forzosa)

El empleado tramita la gestión de la baja en el servidor.

El empleado intenta realizar Cross-selling al cliente.  
El empleado realiza la baja efectiva en el servidor.

*En marketing, se llama venta cruzada (del inglés **Cross-selling**) a la táctica mediante la cual un vendedor intenta vender productos complementarios a los que consume o pretende consumir un cliente. Su objetivo es aumentar los ingresos de una compañía.*

### **RF5.2: Registro de motivos de Bajas**

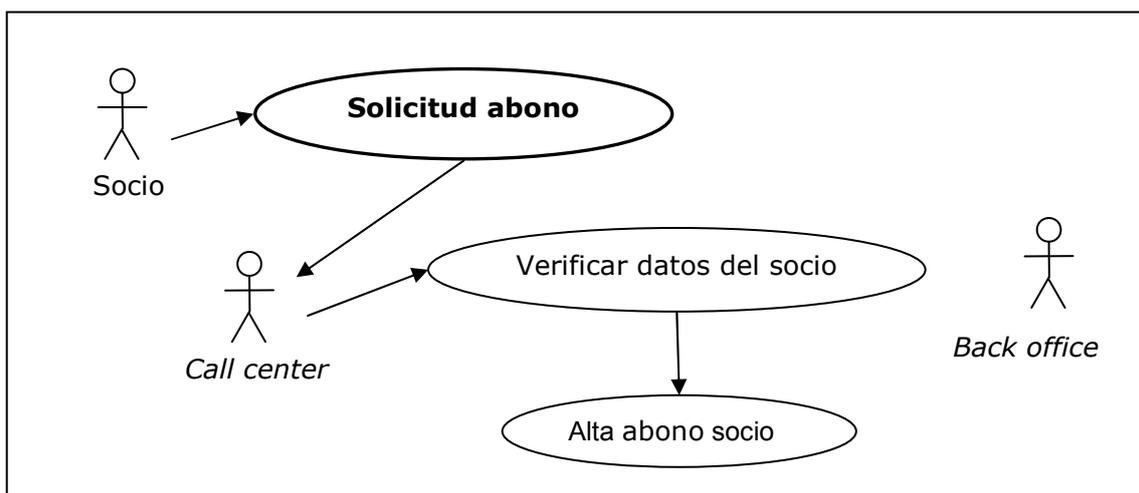
El socio solicita la baja del club a través del formulario de bajas.  
El empleado contra argumenta su voluntad para evitar perder al cliente.  
El empleado intenta realizar cross-selling al cliente.  
El empleado tramita la gestión de la baja en el servidor.  
El empleado registra el motivo de la baja del socio.  
El empleado realiza la baja efectiva en el servidor.  
El socio pasa de estar activo a estar registrado en la tabla auxiliar de socios inactivos con el motivo correspondiente a la baja en el servidor.  
El empleado comunica la baja al socio.

### **RF6: Solicitud de abono**

**Actores:** socio, call center, back office.

**Descripción:** Solicitud de un nuevo abono.

1. El cliente solicita un nuevo abono.
2. El *back office* del club valida el socio y verifica los abonos contratados junto a los pagos al corriente.
3. El *call center* solicita los datos necesarios y ofrece las diferentes posibilidades de abonos existentes que mejor se ajusten.
4. El back office gestiona el alta del abono actualizando el contrato del socio.



**Figura 21 Caso de uso solicitud abono**

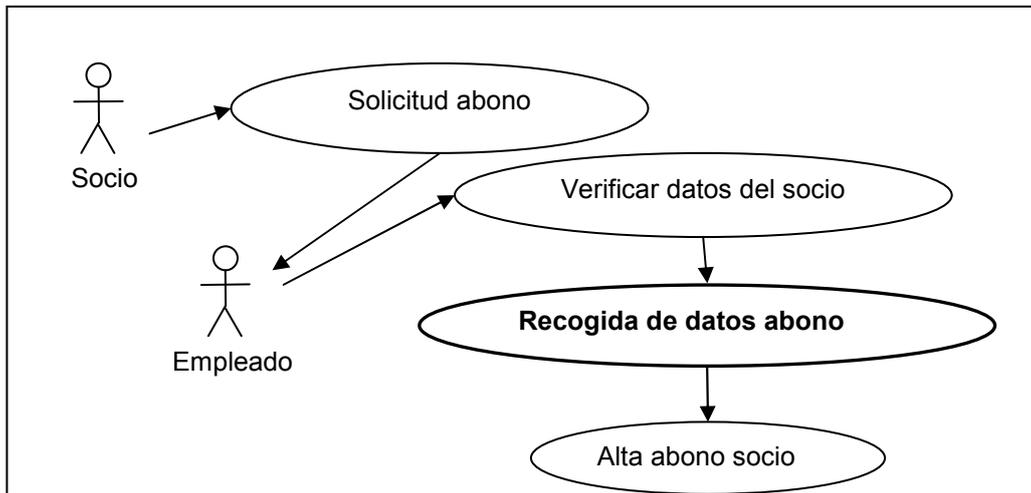
**RF7: Datos abono**

El socio solicita un abono.

El empleado verifica la acreditación del socio y mira que los pagos estén al corriente en el servidor.

El empleado recoge los datos personales, de contacto y tipo de abono deseado.

El empleado da de alta el abono que corresponda en el servidor y lo registra en el contrato del socio.



**Figura 22 Caso de uso datos abono**

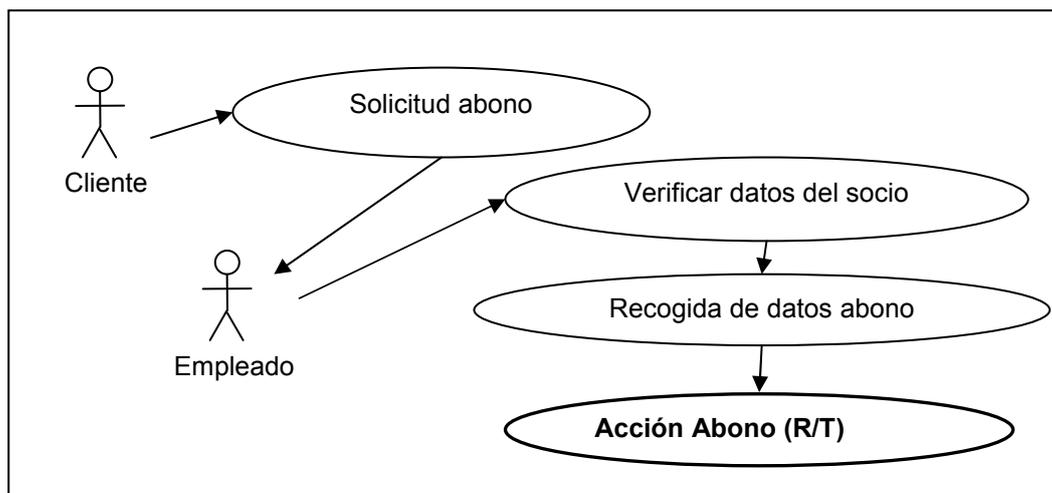
**RF8: Acciones abono**

El cliente solicita un abono.

El empleado call center verifica la acreditación del cliente.

El empleado recoge los datos personales, de contacto y tipo de abono deseado.

El empleado gestiona la acción correspondiente al abono: renovación ó trámite y la registra en el contrato del socio.



**Figura 23 Caso de uso acciones abono**

**RF9: Registro baja abono**

El socio solicita la baja de un abono.

El empleado le proporciona el formulario de baja correspondiente.

El socio entrega la solicitud junto a los datos acreditativos (DNI/carnet de socio).

El empleado realiza la baja del abono al socio y éste queda registrado en la tabla auxiliar de productos inactivos.

El empleado anula el abono del contrato del socio.



# 4

---

# Diseño

## 4 DISEÑO

### 4.1 METODOLOGÍA

En este apartado se explicarán las tecnologías utilizadas para el desarrollo de la aplicación. Explicaré de cada una de ellas las ventajas e inconvenientes y por último haremos una representación exhaustiva de la estructura interna de la base de datos, de las vistas configuradas y de la interface de usuario que representa la capa de presentación con las correspondientes capturas de pantalla.

A partir de los requerimientos del club deportivo se desarrolla una primera versión del diseño funcional para alcanzar los objetivos deseados. Previamente se han seleccionado las entidades y relaciones que formarán parte de la aplicación a desarrollar, es decir, los objetos de negocio, y se han estudiado las posibilidades gráficas para desarrollar la funcionalidad más importante de la aplicación que son sus procesos de negocio.

### 4.2 TECNOLOGÍAS

Las principales tecnologías utilizadas para el desarrollo de este proyecto son:

#### 1. Lenguajes de programación:

- Siebel eScript
- Siebel Tools
- Siebel 8.0
- JavaScript
- PL/SQL

#### 2. Gestión de base de datos:

- Oracle 10G

#### 3. Gestión y administración del servidor Windows:

- Windows 2003 Server
- Optimización de los servicios para mejora del rendimiento

#### **4. Lógica de negocio: los procesos rigen la funcionalidad más importante de la aplicación.**

##### **4.2.1 SIEBEL eSCRIPT**

Siebel eScript es un lenguaje de scripting o programación que los desarrolladores de aplicaciones usan para escribir simples escripts para extender las aplicaciones de Siebel. JavaScript, un lenguaje popular usado principalmente en sitios web, es su idioma maestro.

Siebel eScript es compatible con ECMAScript, que es la aplicación estándar de JavaScript definido por el estándar ECMA-262. El motor de Siebel ST eScript, el valor predeterminado de secuencias de comandos del motor en Siebel 8.0, compatible con ECMAScript Edition 4.

Siebel eScript proporciona acceso a las llamadas de sistema local a través de dos objetos, y Clib SElib, para que Puede utilizar las llamadas de programación al estilo C a ciertas partes del sistema operativo local.

Esta capacidad permite a los programadores para escribir archivos en el disco duro local y realizar otras tareas que la norma JavaScript no puede.

Siebel recomienda considerar la codificación como último recurso. Siebel Tools ofrece muchas maneras de configurar la aplicación Siebel sin necesidad de programación, y estos métodos deben ser agotados antes de intentar escribir código propio, por las siguientes razones:

- El uso de Siebel Tools es más fácil que escribir código
- Más importante aún, el código no puede sobrevivir una actualización. Las personalizaciones creadas directamente en Siebel Tools se actualizan automáticamente cuando se actualiza la aplicación Siebel, pero el código no es tocado, y es posible que tenga que revisarse a raíz de una actualización.
- Por último, la configuración declarativa a través de los resultados de Siebel Tools ofrece un mejor rendimiento que la aplicación de la misma funcionalidad a través de código.

##### **4.2.2 JAVASCRIPT**

JavaScript es un lenguaje basado en objetos. Desarrollado inicialmente como LiveScript por Netscape, nace como complemento del lenguaje HTML, permitiendo la creación de limitadas aplicaciones para este entorno. Lo más habitual es encontrarnos scripts en este lenguaje en formularios que necesiten de alguna verificación, en la creación de ventanas de diálogo, o tratamiento de eventos: haciendo un clic o pasando por encima de un gráfico por ejemplo.

Es por estos motivos que se dice que está basado en objetos ya que nuestro navegador entenderá como objeto a ese gráfico en el que el usuario le ha provocado un evento y dónde nosotros le programamos una respuesta.

### **Características:**

- ❑ Interpretado (no compilado) desde el navegador del usuario
- ❑ Basado en objetos. Sólo los usa, carece de herencias y de clases
- ❑ El código está integrado en el HTML. Queda oculto al usuario, pero puede formar parte del mismo archivo.
- ❑ Enlace dinámico. Los objetos a los que hace referencia deben existir en tiempo de ejecución.
- ❑ No requiere la presencia de servidor, ya que mantiene un funcionamiento local.
- ❑ La aplicación reside en el cliente.

### **Ventajas:**

- ❑ El lenguaje Javascript aporta mucho de cara a la interacción con el usuario.
- ❑ Amplía las posibilidades del HTML, y puede tratar situaciones como la modificación de cadenas de textos introducidas por el usuario para dejarlas en un formato concreto, etc....
- ❑ Su uso es muy generalizado, y existen millones de manuales y aplicaciones que pueden ser utilizadas directamente en una página HTML.

### **Inconvenientes:**

- ❑ Existen determinadas incompatibilidades dependiendo del navegador que se utilice, y por tanto, el resultado que vemos en un navegador puede no ser el mismo que vemos en alguno otro diferente.

## **4.2.3 PL/SQL**

PL/SQL (Procedural Language/Structured Query Language) es un lenguaje de programación incrustado en Oracle que está incorporado en servidores de bases de datos y forma parte de una de las herramientas de Oracle.

Los programas se pueden almacenar en la base de datos como nuevos objetos y se ejecutan en el servidor para ahorrar recursos a los clientes.

Soporta todas las consultas, ya que la manipulación de datos que se usa es la misma que en SQL, incluyendo nuevas **características:**

- ❑ El manejo de variables.
- ❑ Estructuras modulares.
- ❑ Estructuras de control de flujo y toma de decisiones.
- ❑ Control de excepciones.

### **Ventajas:**

- ❑ La reducción del número de accesos y volumen de información intercambiada entre el cliente y el gestor de base de datos.

- ❑ El uso de PL/SQL también tiene la ventaja de encapsular en el servidor todos los accesos a base de datos, evitando tener que embeber sentencias SQL en el resto del código fuente.
- ❑ La ejecución de procedimientos y funciones, al realizarse en el servidor, proporcionan un extra de seguridad.
- ❑ Reduce el tráfico de red (mejor rendimiento) y gestión de errores.

#### **Inconvenientes:**

- ❑ Las aplicaciones que implican la lógica de negocio amplia y procesamiento podría colocar una carga excesiva en el servidor.
- ❑ Mantenimiento y agilidad deficiente.
- ❑ Cuellos de botella

### **4.2.4 ORACLE DATABASE**

Oracle es un sistema de gestión de base de datos objeto-relacional (o ORDBMS por el acrónimo en inglés de Object-Relational Data Base Management System), desarrollado por Oracle Corporation.

Oracle está considerado como el motor de base de datos relacional más usado a nivel mundial. Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando las siguientes **características**:

- ❑ Soporte de transacciones
- ❑ Estabilidad
- ❑ Escalabilidad
- ❑ Soporte multiplataforma

Su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySql o Firebird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo GNU/Linux.

#### **Ventajas:**

- ❑ Multiplataforma
- ❑ Consolidación de aplicaciones de negocio en redes de bases de datos rápidas, fiables y escalables.
- ❑ Maximiza la disponibilidad y elimina la redundancia de datos inactivos.
- ❑ Rendimiento más rápido.
- ❑ Protección con seguridad la información.
- ❑ Oracle soporta todas las funciones que se esperan de un servidor "serio": un lenguaje de diseño de bases de datos muy completo (PL/SQL) que permite implementar diseños "activos", con triggers y procedimientos almacenados, con una integridad referencial declarativa bastante potente.
- ❑ Permite el uso de particiones para la mejora de la eficiencia.
- ❑ El software del servidor puede ejecutarse en multitud de sistemas operativos.
- ❑ Oracle es la base de datos con mas orientación hacía INTERNET.

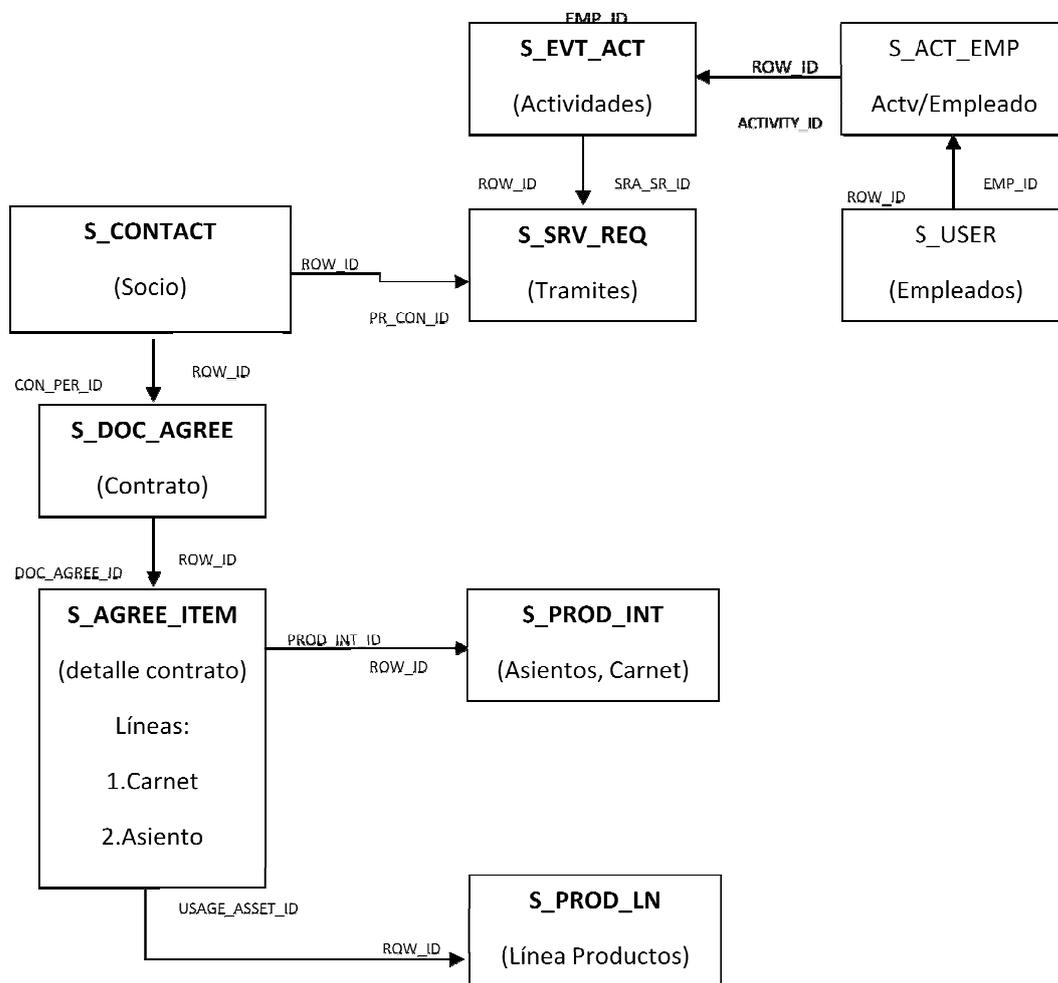
**Inconvenientes:**

- ❑ El mayor inconveniente de Oracle es su precio.
- ❑ Un Oracle mal configurado puede ser desesperantemente lento.
- ❑ Elevado coste en formación.

**4.3 BASE DE DATOS**

Durante este capítulo mostraremos la estructura de la base de datos, entrando en detalle de los diferentes campos y tablas que se hacen servir en la aplicación de Siebel CRM.

**4.4 MODELO DE DATOS**



**Figura 24 Modelo de datos**

### 4.4.1 Diagrama entidad/relación

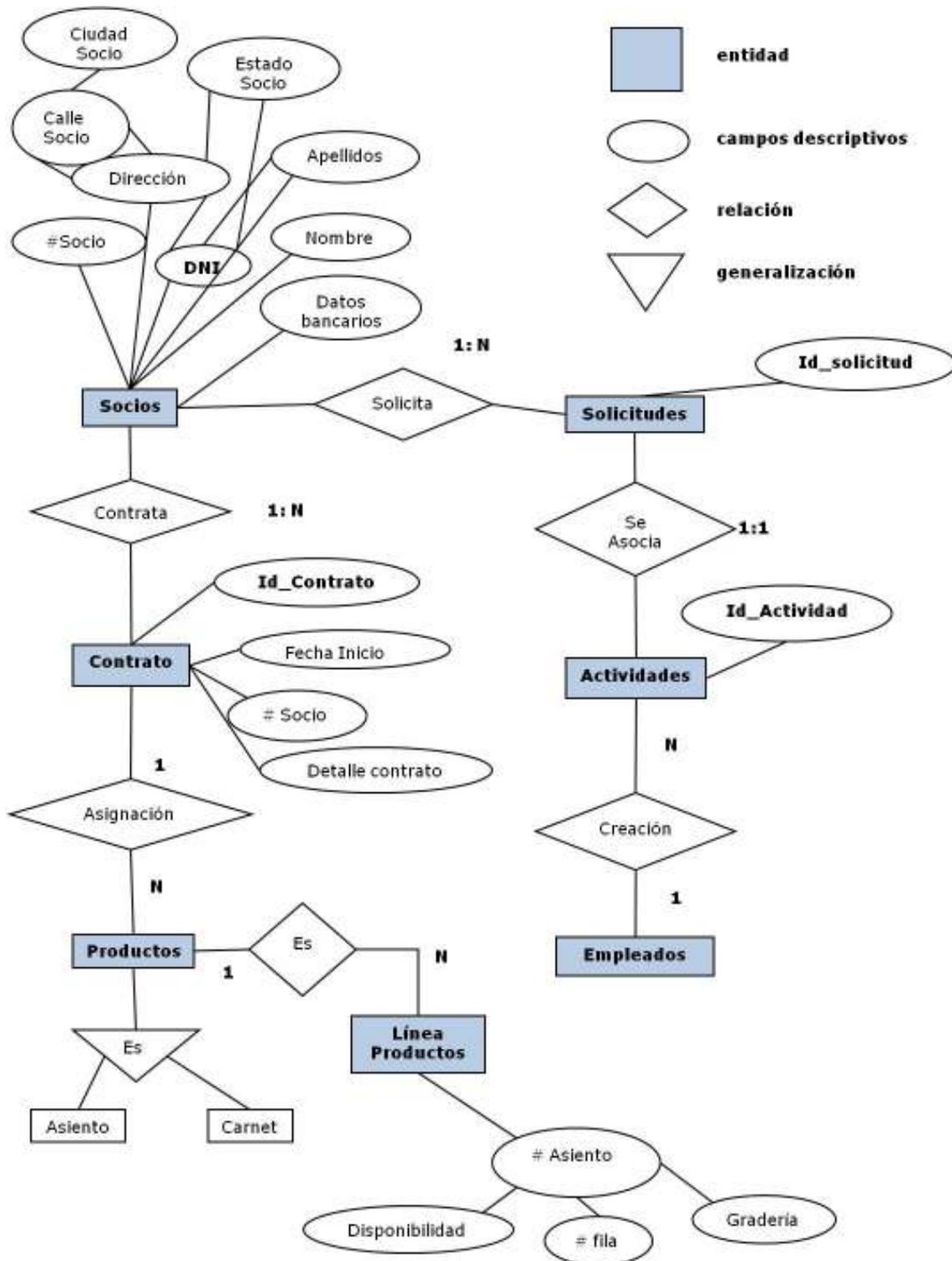


Figura 25 Diagrama Entidad/Relación

Disponemos de dos tipos de actores: empleados y socios. Los empleados pueden crear nuevos contactos.

Entre los empleados podemos diferenciar diferentes perfiles que tendrán acceso a la gestión de trámites según su responsabilidad, y por lo tanto un empleado tendrá acceso a unas vistas específicas.

Los datos del contacto son los siguientes: Nombre, Apellidos, DNI, Dirección, Número de Socio, Motivo de Baja (únicamente obligatorio si el estado de socio está relacionado con la baja), etc.) y los datos bancarios utilizados para domiciliar el pago de todos los productos contratados con el Club.

Los clientes se identifican por DNI. El club almacena el nombre, apellidos, calle, ciudad y datos bancarios de sus clientes, asigna un número de socio a cada uno de ellos y un estado tanto para la baja como para el alta.

Los empleados se identifican por DNI. El club almacena el nombre, apellidos, calle, ciudad, número telefónico, correo electrónico, departamento y posición de sus empleados, y asigna un número de empleado a cada uno de ellos.

Los socios activos tendrán disposición para realizar una serie de trámites, los cuales serán considerados como incidencias por parte de los empleados y relacionados con el socio correspondiente. El Empleado, una vez trabaje en esas incidencias, creará una Actividad, conectada con la incidencia que toque, la cual registrará todo el trabajo realizado en el cliente que corresponda.

Los asientos y carnets de socio son considerados como líneas de productos que se conectan con el cliente a través de un contrato. Cada socio tendrá un único contrato, y cada contrato se podrá relacionar con múltiples Productos.

#### **4.4.2 DISEÑO CONCEPTUAL DE LA BASE DE DATOS**

##### **Entidades:**

**SOCIO** (DNI, Nombre, Apellidos, Dirección [ciudad socio, calle socio], Datos bancarios, [entidad, oficina, DC, número cuenta] Número de socio, motivo baja, estado socio)

**CONTRATO** (Id contrato, Número socio, Fecha Inicio, Descripción)

**PRODUCTOS (ABONO/CARNET)** (Id producto, Tipo, Descripción, Número contrato)

**Tabla extendida Productos: ABONO**

**Tabla extendida Productos: CARNET**

**SOLICITUDES** (Id solicitud, Tipo, Número socio, Estado, Observaciones)

**Tabla extendida de Solicitudes: BAJAS (tipo, motivos)**

**ACTIVIDADES** (Id\_actividad, Descripción, Número empleado,)

**EMPLEADOS** (DNI-e, Nombre, Apellidos, Dirección [ciudad empleado, calle empleado], Número empleado, Número telefónico, Dirección electrónica, Posición, Departamento, Supervisor)

## Relaciones:

**CONTRATACIÓN (SOCIO/CONTRATO):** Relación 1 -N entre cliente y contrato.

Expresa que un cliente puede tener varios contratos.

**ASIGNACIÓN (CONTRATO/PRODUCTOS):** Relación 1-N entre contrato y productos.

Expresa que en un contrato pueden participar diferentes productos.

**TRAMITAR (SOCIO/SOLICITUDES):** Relación 1-N entre socio y solicitudes.

Expresa que un único socio puede llegar a solicitar numerosos trámites, por ejemplo: solicitar un cambio de localidad y peticionar el envío de la revista a la vez.

**CREACIÓN (EMPLEADOS/ACTIVIDADES):** Relación 1-N entre empleados y actividades.

Expresa que un empleado puede realizar N actividades (incidencias) a su cargo, y cada actividad tiene que ir asociada a un único socio.

**ASOCIACIÓN (ACTIVIDADES/SOLICITUDES):** Relación 1-1 entre la actividad y el trámite.

Expresa que una actividad está asociada a un trámite determinado, es decir cada incidencia que el empleado tramite tendrá asociada una actividad.

### 4.4.3 Descripción de las tablas

En Siebel, no es necesaria la creación de las tablas y columnas que forman parte de cada objeto trabajado, es decir, lo que se hace es mapear cada campo que nos interese para realizar la función correspondiente.

La relación de objetos y tablas que existe es la siguiente:

OBJETO ( <i>Business component</i> )	TABLA	DESCRIPCIÓN
<b>XXX Account</b>	S_ORG_EXT	Organización/Cuentas
<b>XXX Action</b>	S_EVT_ACT	Actividades
<b>XXX Actividades/Empleado</b>	S_ACT_EMP	Intersección actividades/empleados
<b>XXX Activity</b>	S_EVT_ACT	Actividades
<b>XXX Agreement Attachment</b>	S_DOC_AGREE_ATT	Archivos adjuntos de contratos
<b>XXX Agreement Item</b>	S_AGREE_ITEM	Líneas de contrato
<b>XXX Agreement Item.Line Number (Sequence)</b>	S_AGREE_ITEM	Líneas de contrato
<b>XXX Contact</b>	S_CONTACT	Contactos
<b>XXX Contact Attachment</b>	S_CONTACT_ATT	Archivos adjuntos de contactos
<b>XXX Contacts</b>	S_USER	Usuarios Siebel
<b>XXX Partner Message</b>	S_BRDCST_MSG	Broadcast message
<b>XXX Personal Payment</b>	S_PTY_PAY_PRFL	Party payment profile
<b>XXX SI Employe External</b>	S_PARTY	
<b>XXX Salutation</b>		
<b>XXX Service Agreement</b>	S_DOC_AGREE	Contratos
<b>XXX Service Request</b>	S_SRV_REQ	Solicitudes de servicio
<b>XXX Service Request Attachment</b>	S_SR_ATT	Archivos adjuntos de solicitudes de servicio

Figura 26 Relación objetos de negocio/tablas

W	Name	Table	Search Specification	Changed	Project	Cache Data	Class	Data Source
	PCB Account	S_ORG_EXT		✓	PCB		CSBPCAccount	
	PCB Action	S_EVT_ACT	{{(Template Flag) =>	✓	PCB		CSBPCAction	
	PCB Actividades/Empleado	S_ACT_EMP		✓	PCB		CSBPCActEmp	
	PCB Activity	S_EVT_ACT		✓	PCB		CSBPCActivity	
	PCB Agreement Attachment	S_DOC_AGREE_ATT		✓	PCB		CSBPCDocAgreeAtt	
	PCB Agreement Item	S_AGREE_ITEM		✓	PCB		CSBPCAgreeItem	
	PCB Agreement Item Line Number (Sequence)	S_AGREE_ITEM		✓	PCB		CSBPCAgreeItemSeq	
	PCB Contact	S_CONTACT		✓	PCB		CSBPCContact	
	PCB Contact Attachment	S_CONTACT_ATT		✓	PCB		CSBPCContactAtt	
	PCB Contacts	S_USER		✓	PCB		CSBPCContact	
	PCB Partner Message	S_BRDCST_MSG	{{(No Broadcasting) =>	✓	PCB		CSBPCPartMsg	
	PCB Personal Payment	S_PTY_PAY_PRFL	{Payment Profile Type	✓	PCB		CSBPCPayPrfl	
	PCB Service Request	S_SRV_REQ		✓	PCB		CSBPCServiceReq	
	PCB SI Employee External	S_PARTY	Id = LoginId	✓	PCB		CSBPCParty	
	PCB Salutation			✓	PCB		CSBPCSalutation	
	PCB Service Agreement	S_DOC_AGREE		✓	PCB		CSBPCDocAgree	
	PCB Service Request Attachment	S_SR_ATT		✓	PCB		CSBPCServiceReqAtt	

Figura 27 Siebel Tools: Relación componentes de negocio/tablas

## 4.5 INTERFACES DE USUARIO

En este capítulo veremos todas las interfaces posibles para las diferentes responsabilidades de los usuarios que utilizarán la aplicación. Desde el usuario administrador que tendrá total acceso a los usuarios de *call center* y *back office*.

### 4.5.1 Interface inicial (login)

Desde este punto el usuario podrá acceder a la aplicación, introduciendo su nombre de usuario y la contraseña.



Figura 28 Selección de perfil

The image shows the login screen for Siebel Call Center 8.0. The top banner features the Oracle logo and 'Siebel Customer Relationship Management'. Below this, the text 'Siebel Call Center 8.0' is displayed. The main area contains a login form with the following elements: a text box for 'User ID', a text box for 'Password', a dropdown menu for 'Connect to' (set to 'Local'), and a checkbox for 'Enable Siebel QuickStart'. At the bottom right are 'OK' and 'Cancel' buttons.

Figura 29 Login

Si introducimos el usuario y password correctamente accederemos a la aplicación, en caso contrario se vuelve a pedir la introducción de los datos.



**Figura 30 Login incorrecto**

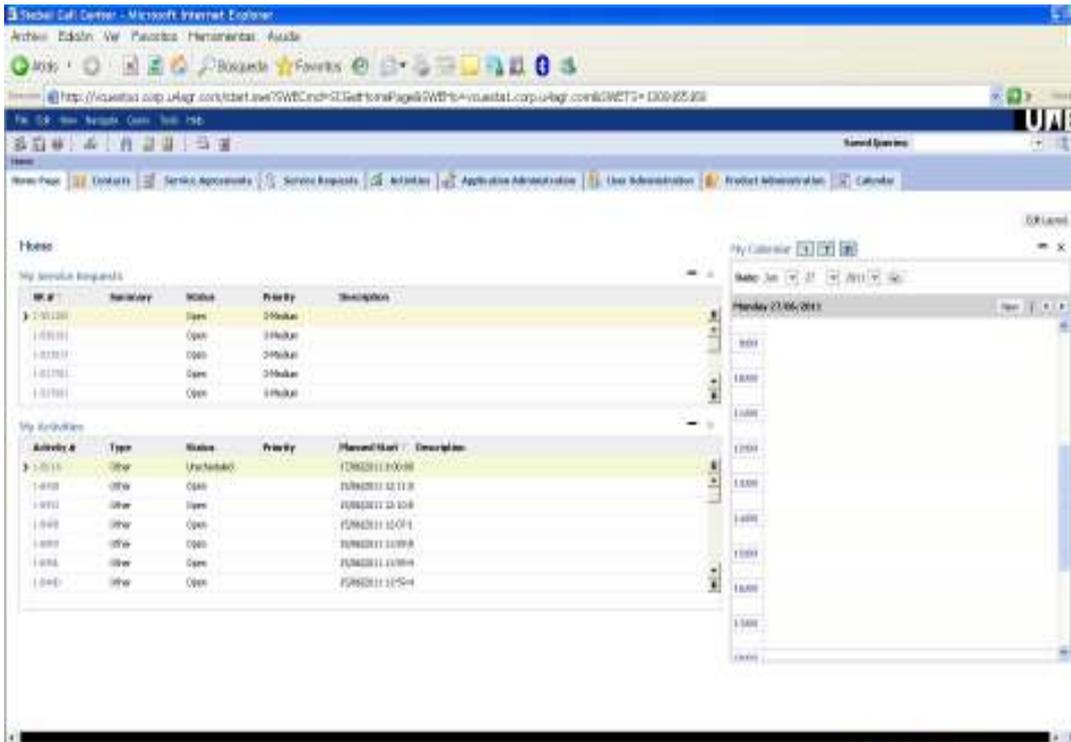
#### **4.5.2 Menú de administrador**

El menú inicial de la aplicación es el siguiente en caso de que entremos como administrador. El usuario administrador tendrá total acceso a todas las vistas de la aplicación. En la parte derecha tenemos el menú de administración de la aplicación (dónde se asignarán las vistas y responsabilidades), administración de usuarios (crear, editar y eliminar) y administración de los productos ofrecidos por el club.

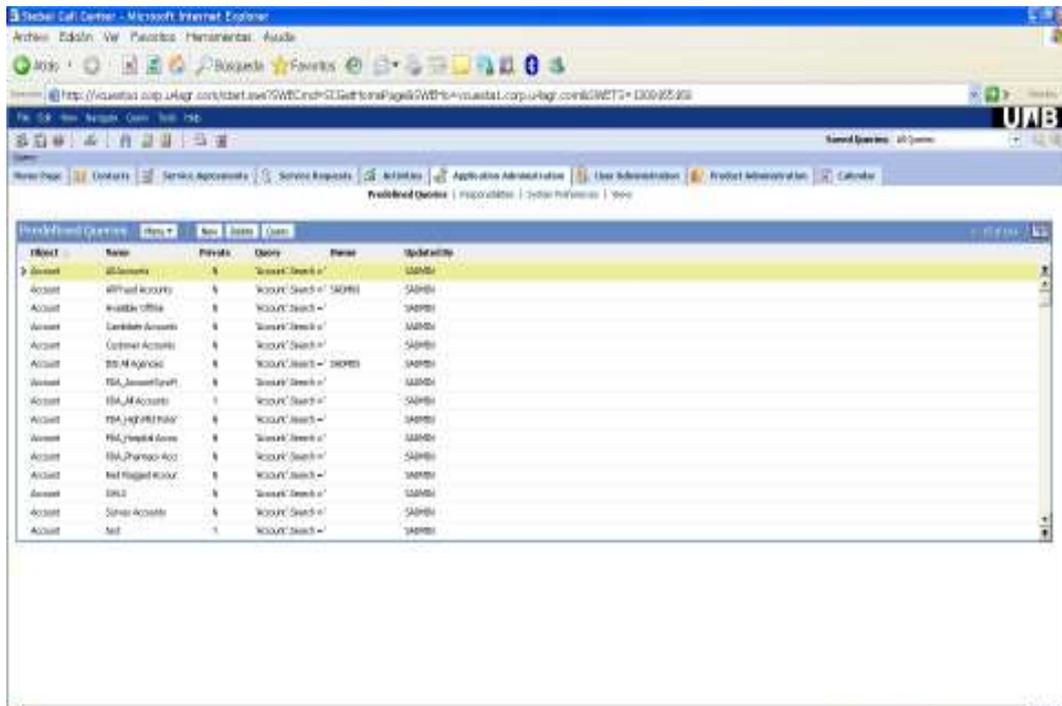
En el marco principal encontramos un listado de actividades y solicitudes pendientes con su estado y un calendario.

#### **4.5.3 Interface de administrador**

Las vistas a las que podemos tener acceso siendo un usuario administrador son las siguientes:



**Figura 31 Vista inicial del usuario administrador**



**Figura 32 Vista de administración de la aplicación**

**Implantación CRM en un club deportivo**  
 Ingeniería Técnica en Informática de Sistemas  
 Escuela de Ingeniería de Sabadell

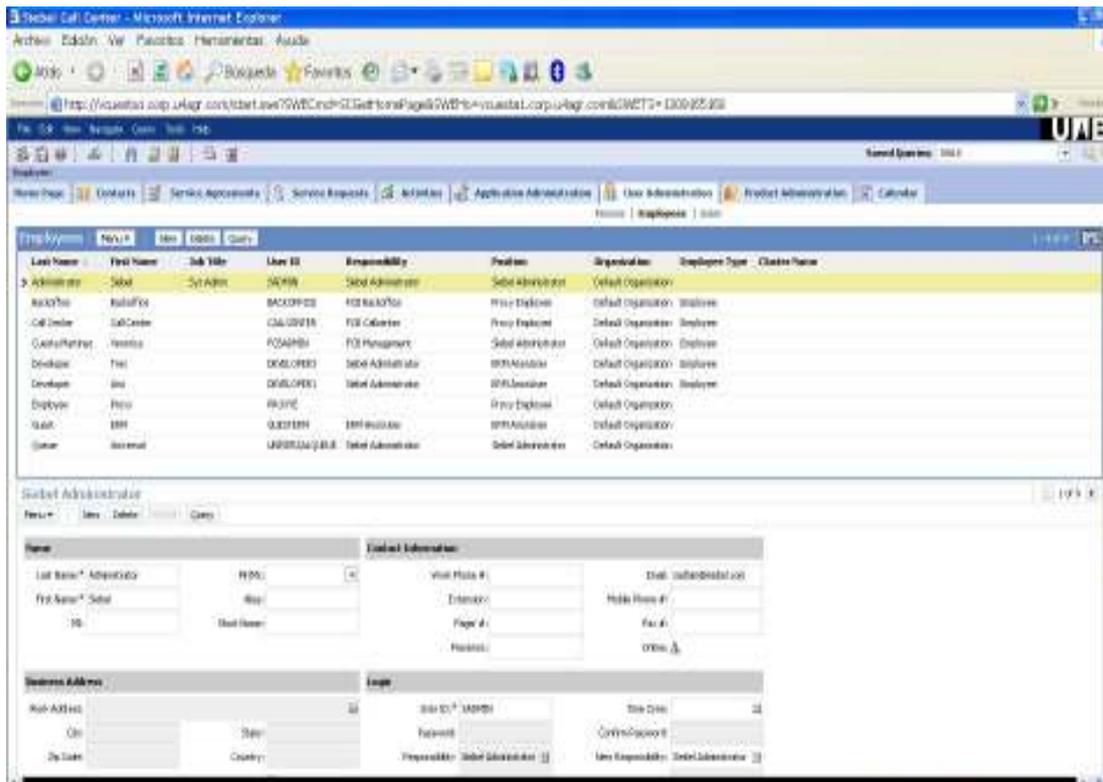


Figura 33 Vista de administración de usuarios

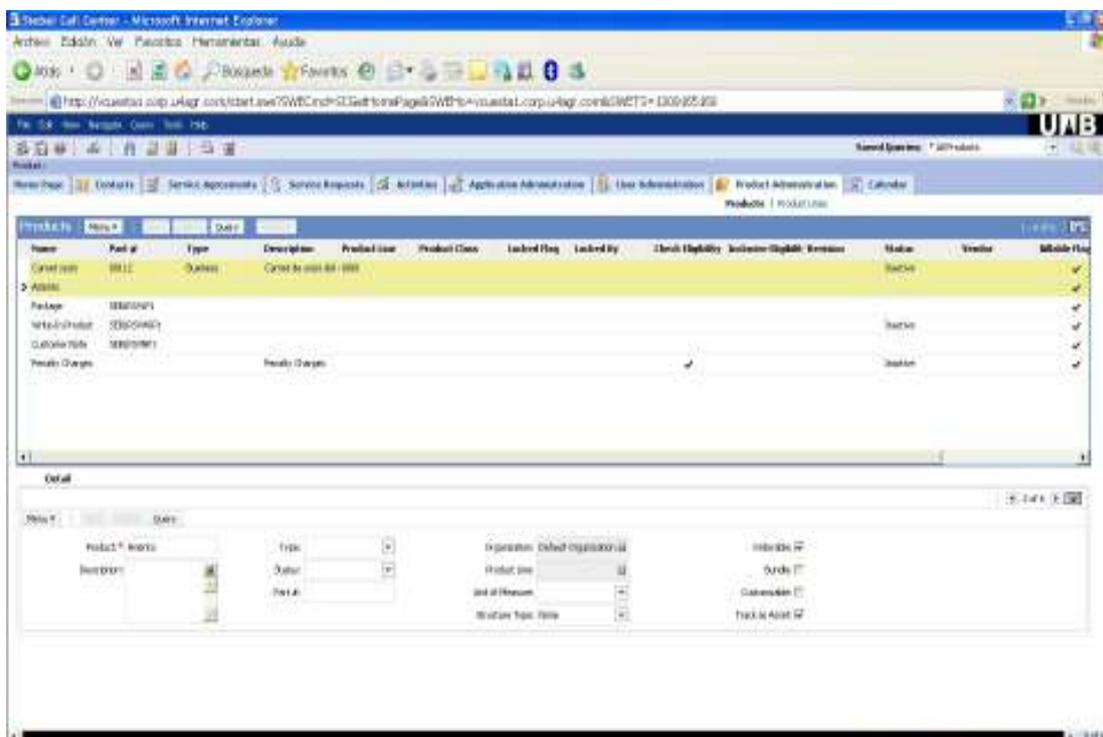


Figura 34 Vista de administración de productos

#### 4.5.4 Menú de usuario no administrador

Esta es la vista para un usuario sin privilegios de administrador: *call center* y *back office*, dónde encontramos diferentes opciones. Puesto que estos usuarios tendrán la primera toma de contacto con el cliente o socio y realizarán todo tipo de consultas y modificaciones, deben tener acceso a la vista de Contactos, Contratos, Solicitudes de Servicio y Actividades. Las opciones del menú varían en comparación con las de un usuario administrador.

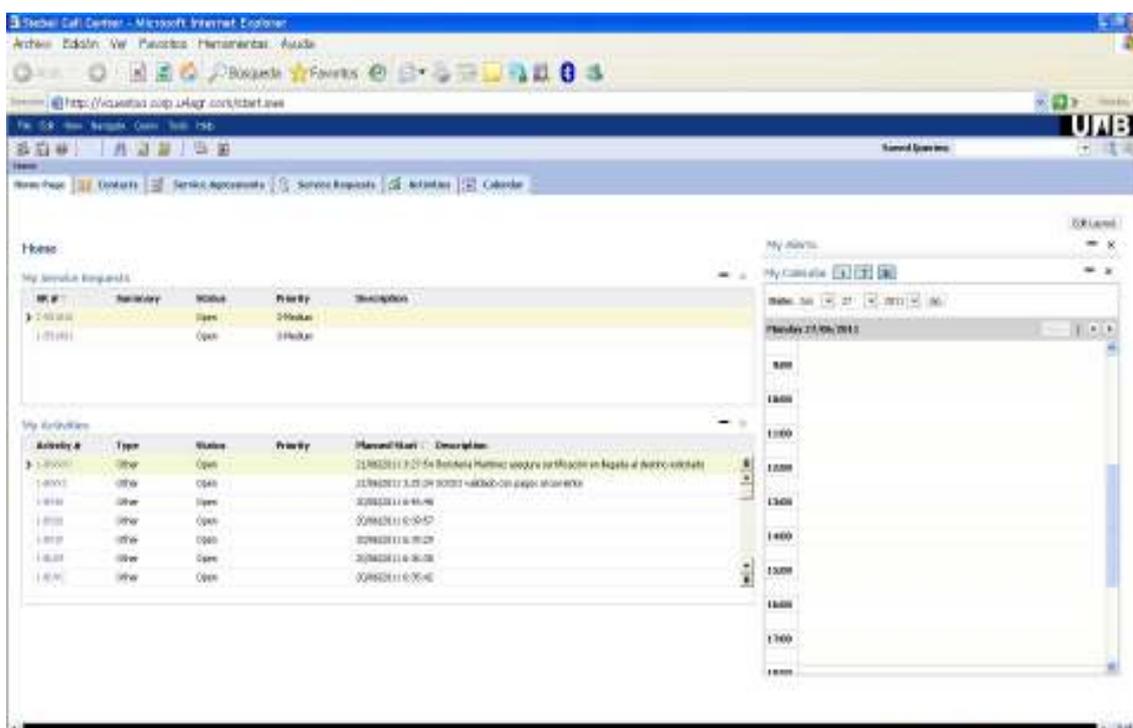


Figura 35 Página de inicio de callcenter

#### 4.5.5 Menú de página inicial

Al entrar en la aplicación se nos cargará automáticamente la página de inicio que mostrará las actividades y solicitudes de servicio pendientes en función del perfil de usuario con el que nos logemos.

En "My Service Request" se muestra un list applet con todas las solicitudes a tramitar con su estado correspondiente, por defecto aparecerá las que están abiertas y el día en que se abrieron. Para entrar en más detalle de cada una, si nos situamos encima de ellas podremos navegar a la solicitud en concreto.

En "My Activities" nos muestra el listado de actividades pendientes. Una actividad es una tarea que falta por hacer antes de cerrar una solicitud, de manera que muestra un histórico de cada una de las modificaciones que se han realizado sobre una incidencia específica.

### 4.5.6 Menú de contactos

Al hacer clic sobre la vista de contactos, aparecerá un listado de los socios activos, o pendientes de alta. Si nos situamos sobre un contacto concreto podremos visualizar en la parte inferior el detalle con la información completa, una vista con los datos bancarios, la relación de contratos activos, las solicitudes tramitadas de ese contacto y las actividades asociadas. Por último una vista de adjuntos dónde se podrá almacenar cualquier dato electrónico referente al contacto concreto.

### 4.5.7 Interface para dar de alta un nuevo contacto

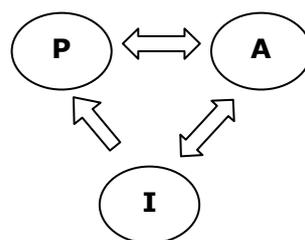
El empleado realiza un alta nueva de un cliente, para ello se han de solicitar los datos personales y bancarios pertinentes con la finalidad de poder realizar los pagos de los productos contratados del club.

Una vez realizada el alta del nuevo socio, se generará un estado del nuevo cliente que estará en modo pendiente hasta que se realice la validación de datos para comprobar la integridad de éstos y evitar que haya duplicados de los registros en la base de datos.

El proceso de validación de los datos, quedará realizado por parte del *back office* y pasará a un estado activo.

Una vez esté el cliente registrado, será considerado como socio y se tendrá que tener en cuenta los procesos de los trámites posibles que se puedan llevar a cabo en función de la circunstancia de cada uno.

La aplicación mostrará una vista principal de Contactos en la que se recoge y muestra en un listado (form\_list applet) la información requerida: DNI, apellidos, nombre, status. Este último campo determinará el estado de cada cliente, es decir, en cuanto esté creado, se le asociará un estado pendiente hasta que se valide, y el flujo que seguirá hasta ser validado es el siguiente: *pending/active/inactive*



Por defecto habrá un status para cada uno de los clientes que se vayan creando en el que aparecerá como pendiente hasta que el *back office* lo valide y seguirá el flujo correspondiente.

La vista de recogida de datos de clientes aparecerá listada en la parte superior, y es dónde se realizará la validación. La parte inferior de la vista de contactos dispondrá de seis apartados dónde se mostrará un detalle con todos los campos de cada cliente incluyendo sus datos bancarios, los contratos asociados a cada uno de ellos, los trámites gestionados, las actividades generadas para ello, y una última columna dónde se podrá adjuntar cualquier tipo de documento que se necesite para realizar

las solicitudes requeridas (contrato físico, carnet de socio, solicitud, reclamación, dni,..)

Cada trámite será considerado como una nueva incidencia, y gestionado por el empleado guardando relación directa con el socio en cuestión. De igual forma que cada vez que se trabaje sobre esas incidencias, el empleado encargado creará una actividad que reflejará el registro de todo el trabajo realizado en ese socio.

#### 4.5.8 Interface de los contactos

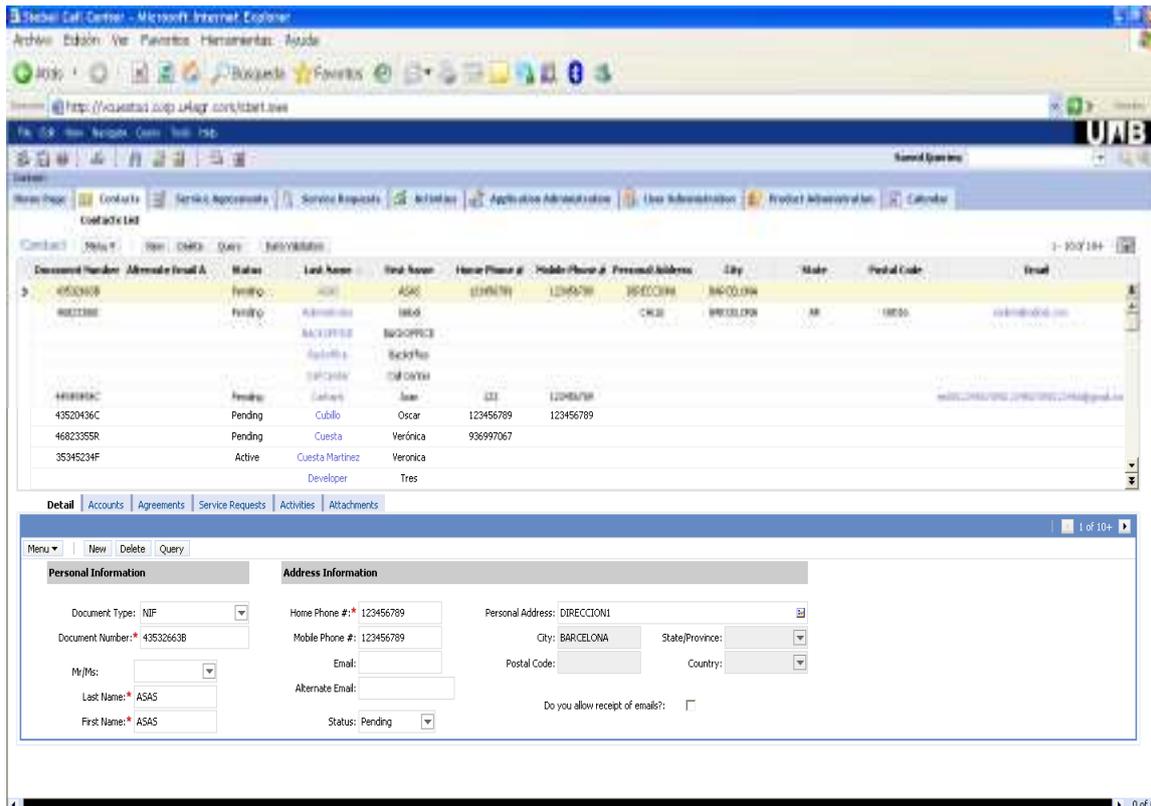
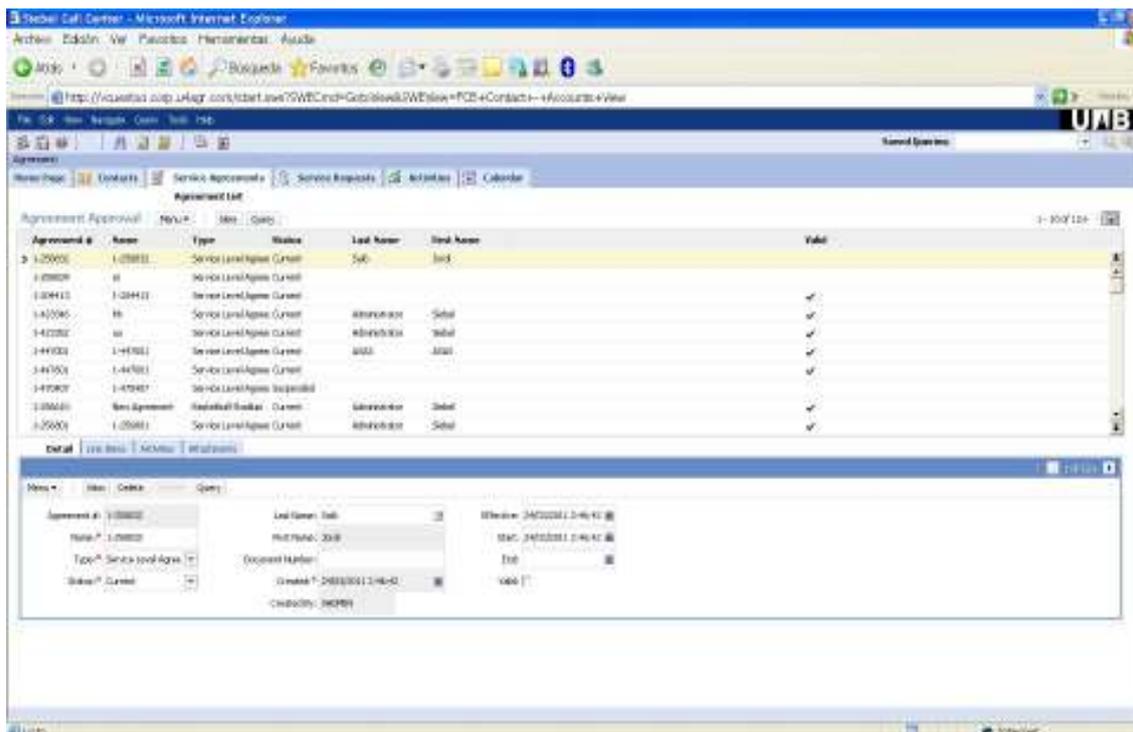


Figura 35. Interface de contactos

#### 4.5.9 Interface de los contratos

Al hacer clic sobre la vista de contratos, aparecerá un listado de los contratos con el detalle de cada cliente, los productos y servicios contratados y las actividades asociadas. En la parte inferior se muestra toda la información detallada en forma de formulario.



**Figura 36 Interface de los contratos "Service Agreements"**

#### 4.5.10 Menú de solicitudes de servicio

Este es el menú de la vista de solicitudes de servicio dónde quedan registrados todas los posibles trámites que pueda solicitar un socio del club. Se considera el centro neurálgico del aplicativo ya que cualquier petición ha de ser registrada desde esta vista. Cada trámite será considerado como una incidencia para el usuario empleado.

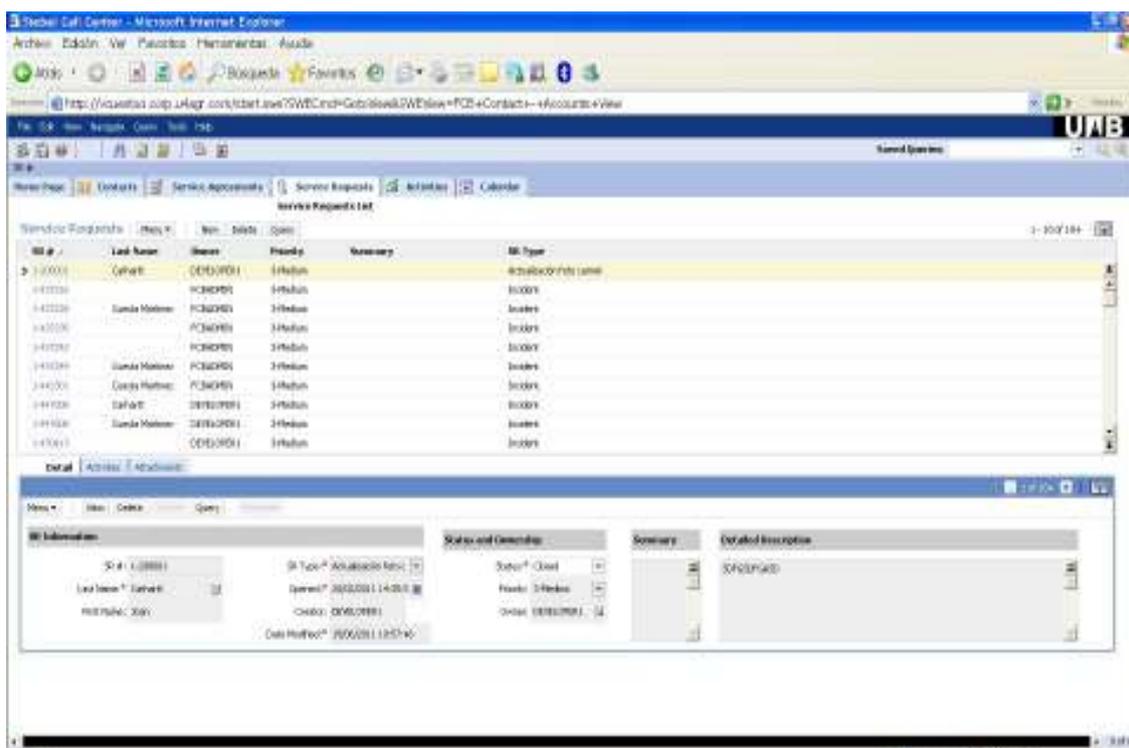
El menú de esta vista nos proporciona poder crear incidencias nuevas, consultarlas, y ver su detalle.

#### 4.5.11 Interface de solicitudes de servicio

La vista asociada a las solicitudes de servicio vendrá caracterizada por un primer listado en la parte superior de la pantalla dónde el empleado podrá visualizar y crear todas las posibles solicitudes que puedan tramitarse: los trámites son los siguientes: a) actualización de la foto de carnet; b) duplicados de abono(s), c) duplicado de carnet de socio, d) cambio de asiento, e) carnet y diploma de socio difunto, f) renovaciones abonos, g) renovaciones carnets, h) bajas de abonos, i) nuevos abonos y j) petición del envío de revistas. Cada trámite tendrá un link asociado para poder visualizar su detalle en la parte inferior de la vista. Puesto que el empleado tramitará una actividad según su área funcional tendrá una responsabilidad para poder realizar un trámite u otro, y cada vez que trabaje en una incidencia para el socio creará una actividad.

Otra vista que se debe ofrecer es la de poder adjuntar un documento auxiliar para la realización de la gestión, ya sea a nivel de certificado de defunción (caso trámite e) o bien adjuntar un documento identificativo para petitionar un servicio nuevo (caso trámite j).

La vista inferior de la interface de **trámites** quedará estructurada en un primer nivel dónde se reflejará cada **detalle** de éstos, un segundo nivel dónde se verán las **actividades** generadas por los empleados para cada uno de los trámites (según responsabilidad), y un tercero dónde se **adjuntará** todos los documentos asociados a las gestiones que lo requieran.



**Figura 37. Vista solicitudes de servicio "Service Requests"**

#### **4.5.12 Menú de actividades**

Al hacer "clic" sobre la vista "Activities", aparecerá un listado de todas las actividades cread con su detalle. En función de la responsabilidad del perfil que se logue en la aplicación podrá ver sus actividades, o bien, todas las actividades.

Se podrán consultar y crear nuevas actividades, aunque lo normal sea que vengan asociadas por una solicitud de servicio previa.

#### **4.5.13 Interface de actividades**

En la vista superior se muestra un listado de actividades, y en la vista inferior el detalle de cada una de ellas junto a los datos del socio al que afecta dicha solicitud.

En función de la responsabilidad con la cual accedamos, tendremos acceso a todas las actividades, o bien a "mis actividades".

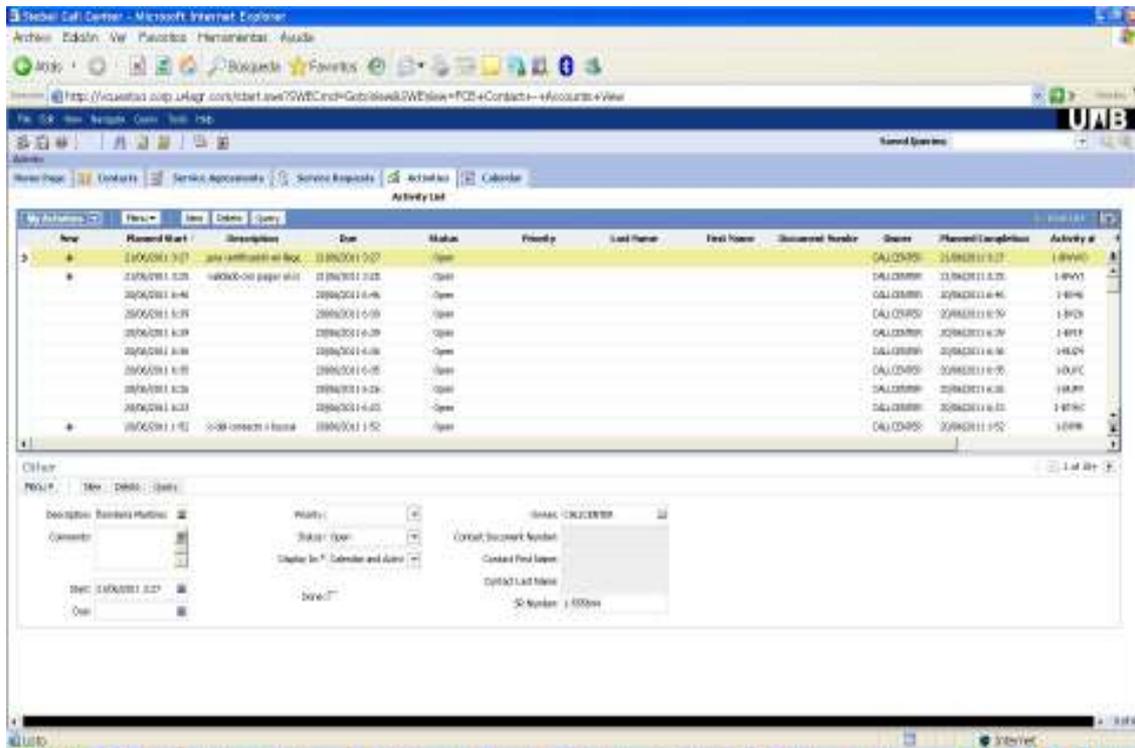


Figura 38. Vista solicitudes de actividades "Activities"

#### 4.5.14 Interface del calendario

Por último nos aparece una interface que muestra un calendario con las solicitudes y actividades pendientes del usuario que acceda a la aplicación:

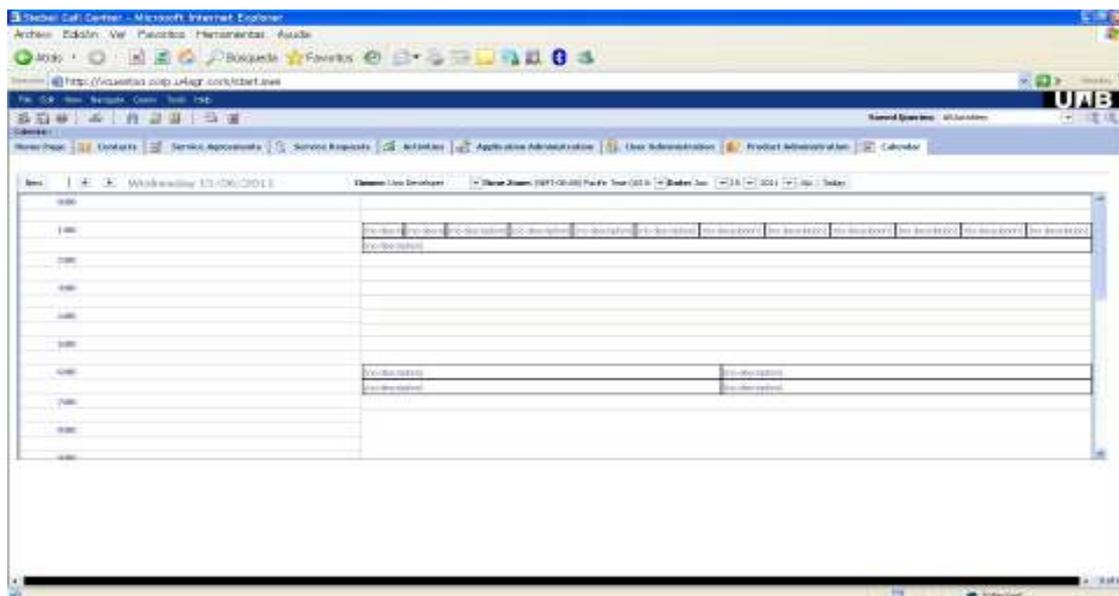


Figura 39. Vista calendario "Calendar"

## 4.6 ORGANIZACIÓN DE LOS ARCHIVOS EN EL SERVIDOR

A continuación se muestra la estructura física del servidor dónde encontramos los archivos de repositorio de siebel:

### C:\sba80\client\OBJECTS\ENU

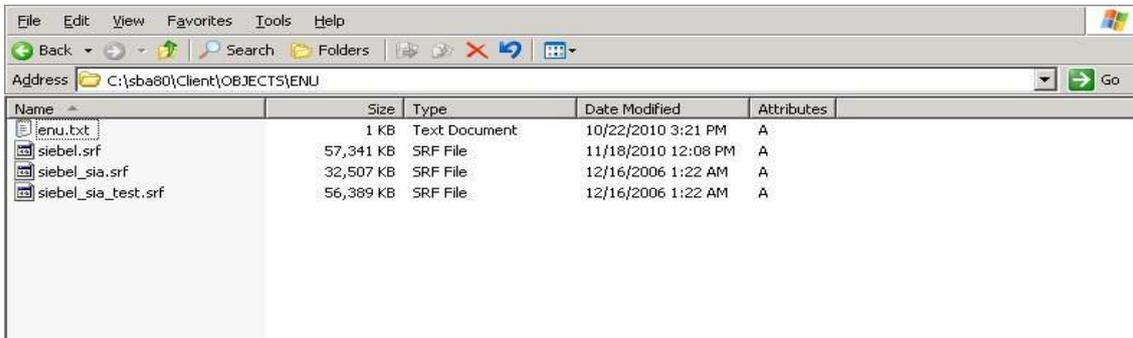


Figura 40. Organización archivos Windows Server

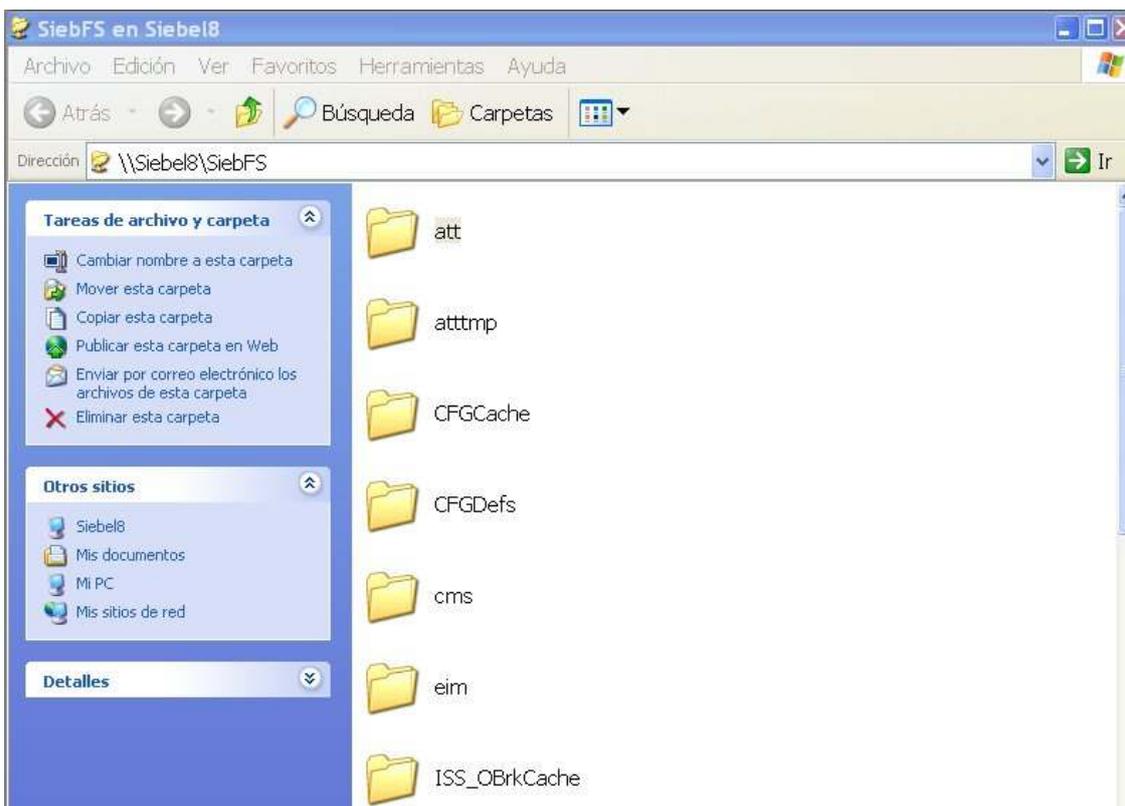


Figura 41 Carpeta del servidor dónde se almacenan los archivos adjuntos

### 4.6.1 Permisos y control de acceso a datos

El software de gestión de la base de datos que se ha instalado es **Oracle Database 10 g**. Accediendo al Database Control de la base de datos con el usuario SYSTEM creado para su mantenibilidad, podemos gestionarla, incluyendo la parte de la administración de usuarios a la cual se ha accedido para ello:

<b>USER</b>	<b>PASSWORD</b>
DEVELOPER1	DEVELOPER1
CALLCENTER	CALLCENTER
BACKOFFICE	BACKOFFICE
SYSTEM	SYSTEM8XXX

**Figura 42 Usuarios de la base de datos**



Figura 43 Vista de administración de usuarios

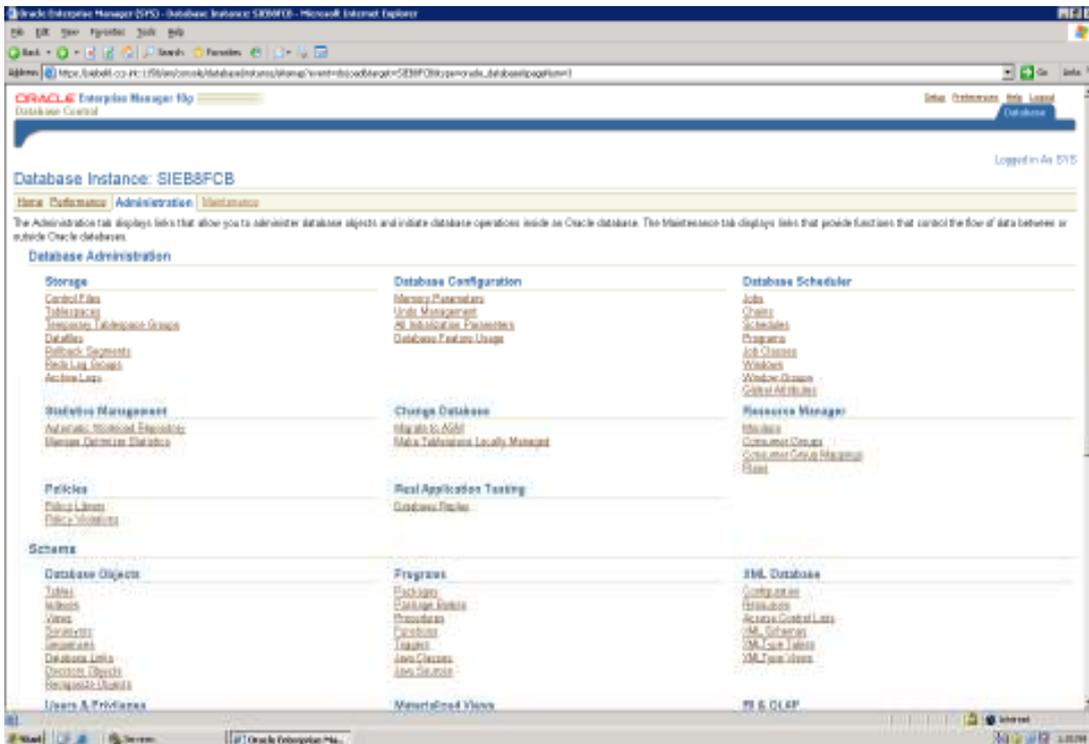


Figura 44 Vista de administración de la base de datos

En la aplicación SIEBEL gestionamos la administración a través de un usuario que tiene privilegios de "Siebel Administrator" llamado SADMIN o bien DEVELOPER1.

Asignamos al usuario CALLCENTER las responsabilidades pertinentes para que tenga acceso exclusivo a las vistas correspondientes a ese perfil. Para ello navegamos hacia la vista de administración de usuarios y pinchamos en el desplegable para asignarlas (*responsability*): en el caso el CALLCENTER, por defecto tendrá una responsabilidad de Call Center Administration por defecto, y todas las asociadas con el área de Call Center. De la misma forma se asignan las responsabilidades al usuario BACKOFFICE.

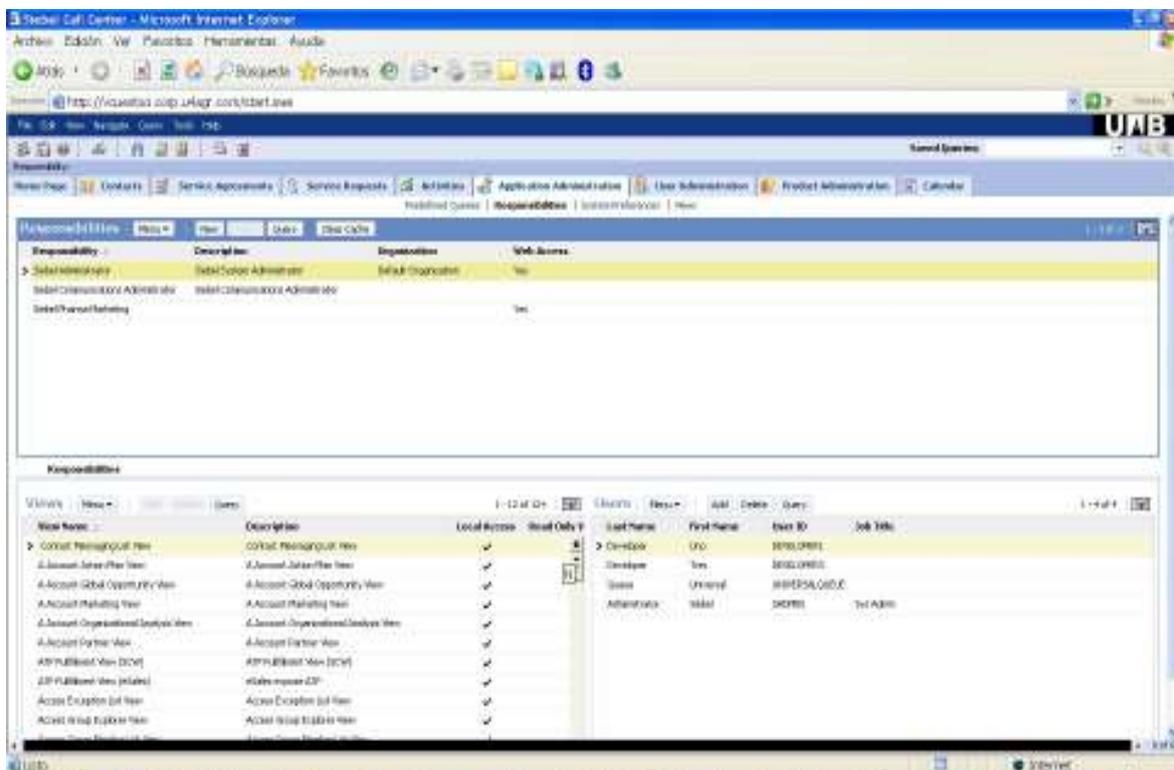


Figura 45. Vista web administración Siebel

## 4.6.2 VISTAS/RESPONSABILIDADES

VIEWS (Name)	Funcionalitat	USERS
<b>GENERAL</b>		
0. FCB Sales Home Page View	Home Page: página de inicio donde se ven las solicitudes y actividades pendientes del usuario que se logea en el sistema	CALLCENTER BACKOFFICE ADMINISTRATOR
<b>CONTACTS</b>	<b>FCB Contracts Screen</b>	
1. FCB Visible Contact List View	Vista de creación de usuarios y de consulta de datos de usuarios con su detalle	CALLCENTER BACKOFFICE ADMINISTRATOR
2. FCB Contact - Accounts View	Vista para la recogida de datos bancarios de los clientes. (Iben)	CALLCENTER ADMINISTRATOR
3. FCB Contact-Contract List View	Vista relación de contratos y clientes, permite consultar contratos y crear nuevos para un cliente en concreto. (CALLCENTER CONSULTA Y MK CREA)	CALLCENTER BACKOFFICE ADMINISTRATOR
4. FCB Contact-Service Request List View	Vista de creación de solicitudes de servicio y consulta de SFs	CALLCENTER BACKOFFICE ADMINISTRATOR
5. FCB Contact-Activities Detail View	Vista para visualizar un listado de actividades asociado a cada contacto	CALLCENTER ADMINISTRATOR
6. FCB Contact Attachment View	Vista de adjuntos para archivar documentos requeridos (La documentación que firma el cliente al darse de alta, la archiva MKT)	CALLCENTER BACKOFFICE ADMINISTRATOR
<b>SERVICE AGREEMENTS</b>	<b>Agreement Screen</b>	
1. FCB Agreement Detail View	Vista con el detalle de cada contrato. Mostramos los contactos y las fechas de los contratos. (Detalle).	CALLCENTER BACKOFFICE ADMINISTRATOR
2. FCB Agreement Service Request List View	Vista que permite crear solicitudes de servicio para el contrato seleccionado.	BACKOFFICE ADMINISTRATOR
3. FCB Agreement Line Items View	Vista que permite asociar la línea de productos (carnes/asiento) al contrato seleccionado.	CALLCENTER BACKOFFICE ADMINISTRATOR
4. FCB Agreement Activities List View	Vista que permite crear actividades que contengan el histórico de las SFs con el status correspondiente.	CALLCENTER BACKOFFICE ADMINISTRATOR
5. FCB Agreements Attachment View	Vista que permite adjuntar la documentación necesaria para realizar las actividades pertinentes y modificarlas al contrato que toque	CALLCENTER BACKOFFICE ADMINISTRATOR
<b>SERVICE REQUESTS</b>	<b>FCB Service Request Screen</b>	
1. FCB Service Request List View	Vista para ver el detalle de las solicitudes de servicio que demanda un cliente con el detalle de cada una	CALLCENTER BACKOFFICE ADMINISTRATOR
2. FCB Service Request Activity View	Vista para visualizar las actividades relacionadas con cada solicitud de servicio	CALLCENTER BACKOFFICE ADMINISTRATOR
3. FCB Service Request Attachment View	Vista de adjuntos para archivar documentos requeridos	CALLCENTER BACKOFFICE ADMINISTRATOR
<b>ACTIVITIES</b>	<b>FCB Activites Screen</b>	
1. FCB Activity List View	Vista para visualizar las actividades propias de cada usuario (mis actividades)	CALLCENTER BACKOFFICE ADMINISTRATOR
2. FCB All Activity List View	Vista para visualizar las actividades relacionadas con cada solicitud de servicio	CALLCENTER BACKOFFICE ADMINISTRATOR
<b>ADMINISTRATION</b>		
<b>APPLICATION ADMINISTRATION</b>	<b>System Administration Screen</b>	
1. Predefined Query Administration View	Vista para administrar las queries que se podrán ejecutar de forma predefinida a la hora de navegar por la aplicación y muestre unos valores concretos predefinidos	ADMINISTRATOR FCADMIN
2. Responsibility Administration View	Vista de administración de las responsabilidades	ADMINISTRATOR FCADMIN
3. View Administration View	Vista de administración de las vistas en las que les podremos asignar las responsabilidades para tener acceso o no a cada una.	ADMINISTRATOR FCADMIN
<b>USER ADMINISTRATION</b>	<b>User Administration Screen</b>	
1. All Person Administration View	Vista de administración de las personas que están dadas de alta, a nivel de contactos y empleados (podemos asignar una responsabilidad)	ADMINISTRATOR FCADMIN
2. Employee List View	Vista de administración de empleados que tienen acceso a la aplicación.	ADMINISTRATOR FCADMIN
3. All User Administration View	Vista de administración de usuarios que tienen acceso a la aplicación.	ADMINISTRATOR FCADMIN
<b>PRODUCT ADMINISTRATION</b>	<b>ISS Unified Administration Screen</b>	
1. ISS Product Administration View	Vista de administración de productos, donde podemos dar de alta nuevos, o bien, administrar los ya existentes.	ADMINISTRATOR FCADMIN
2. Product Line Administration View	Vista de administración de la línea de productos, donde podemos dar de alta nuevos, o bien, administrar los ya existentes.	ADMINISTRATOR FCADMIN

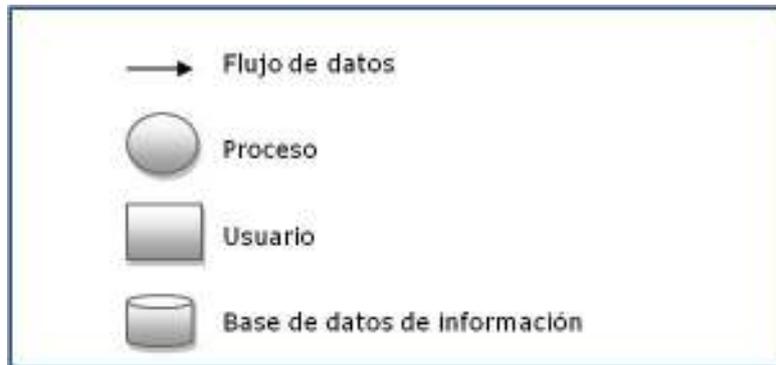
Figura 46. Vistas y responsabilidades

### 4.6.3 Lógica de negocio

A continuación se detallan los 2 escenarios más representativos extraídos del documento de requerimientos del cliente:

Ciclo de vida del socio y ciclo de los abonos.

La leyenda representativa para los diagramas es la siguiente:



**Figura 47. Leyenda de diagramas**

El cambio de estado de las solicitudes de servicio y el propietario de éstas son los factores más determinantes del flujo del negocio. Por norma general cada trámite será considerado como una *Service Request* o solicitud de servicio (petición del cliente) y la secuencia de flujo estándar con las acciones asociadas es la siguiente:

<sup>3</sup>



**Figura 48. Flujo de estados de solicitudes de servicio**

---

<sup>3</sup> CC: call center BO: back office

## CICLO DE VIDA DE LOS SOCIOS

### ALTAS USUARIOS (RF1, RF2, RF3)



El cliente solicita una nueva alta de socio, de manera que el call center crea una actividad al back office para que valide los datos del cliente proporcionados en la base de datos correspondiente. Una vez verificada la consulta le cede la responsabilidad de nuevo al call center, creando una actividad para que efectúe el alta y se lo comunique al nuevo cliente asignándole un número de socio. El cliente quedará en estado activo.



Figura 49. Altas usuarios

**SOLICITUD DE SERVICIO Actualizar foto (RF4.1)**



El cliente solicita una actualización de la fotografía de carnet acreditando su identidad al empleado call center, y éste envía la responsabilidad a back office creándole una actividad.

El empleado back office valida que los datos del cliente en la base de datos correspondiente sean correctos y se dispone a eliminar la fotografía antigua. Una vez eliminada, realiza la fotografía al socio y la actualiza en la base de datos.

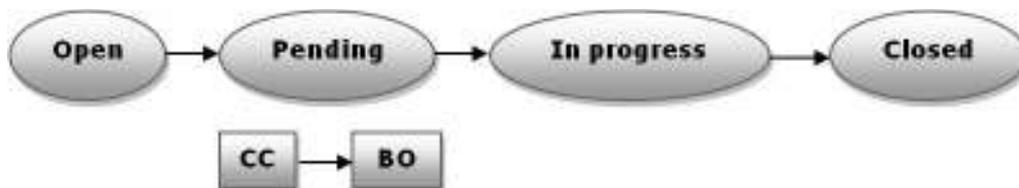


Figura 50. Actualizar fotografía

### SOLICITUD DE SERVICIO Duplicado de abono ó carnet de socio (RF4.2)

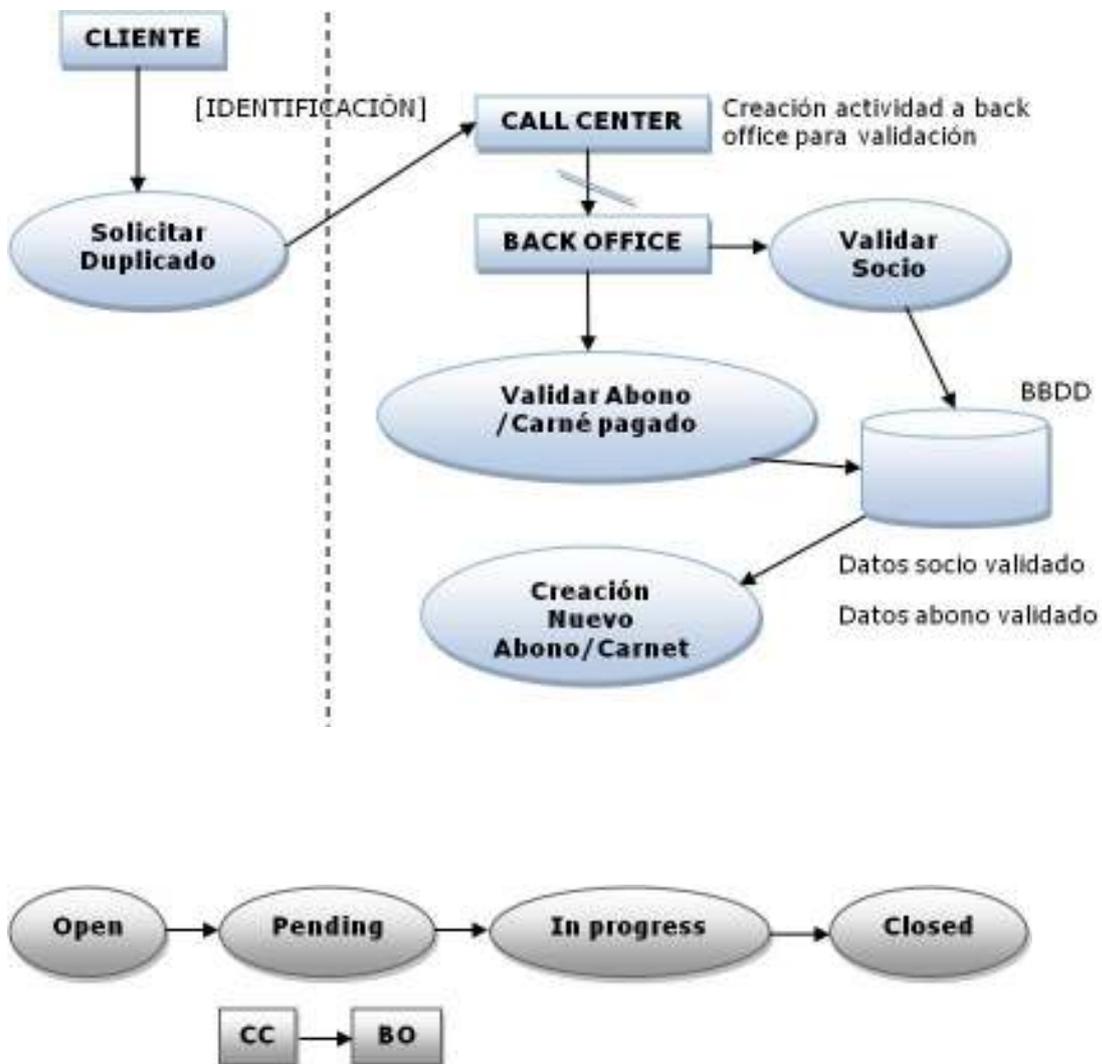
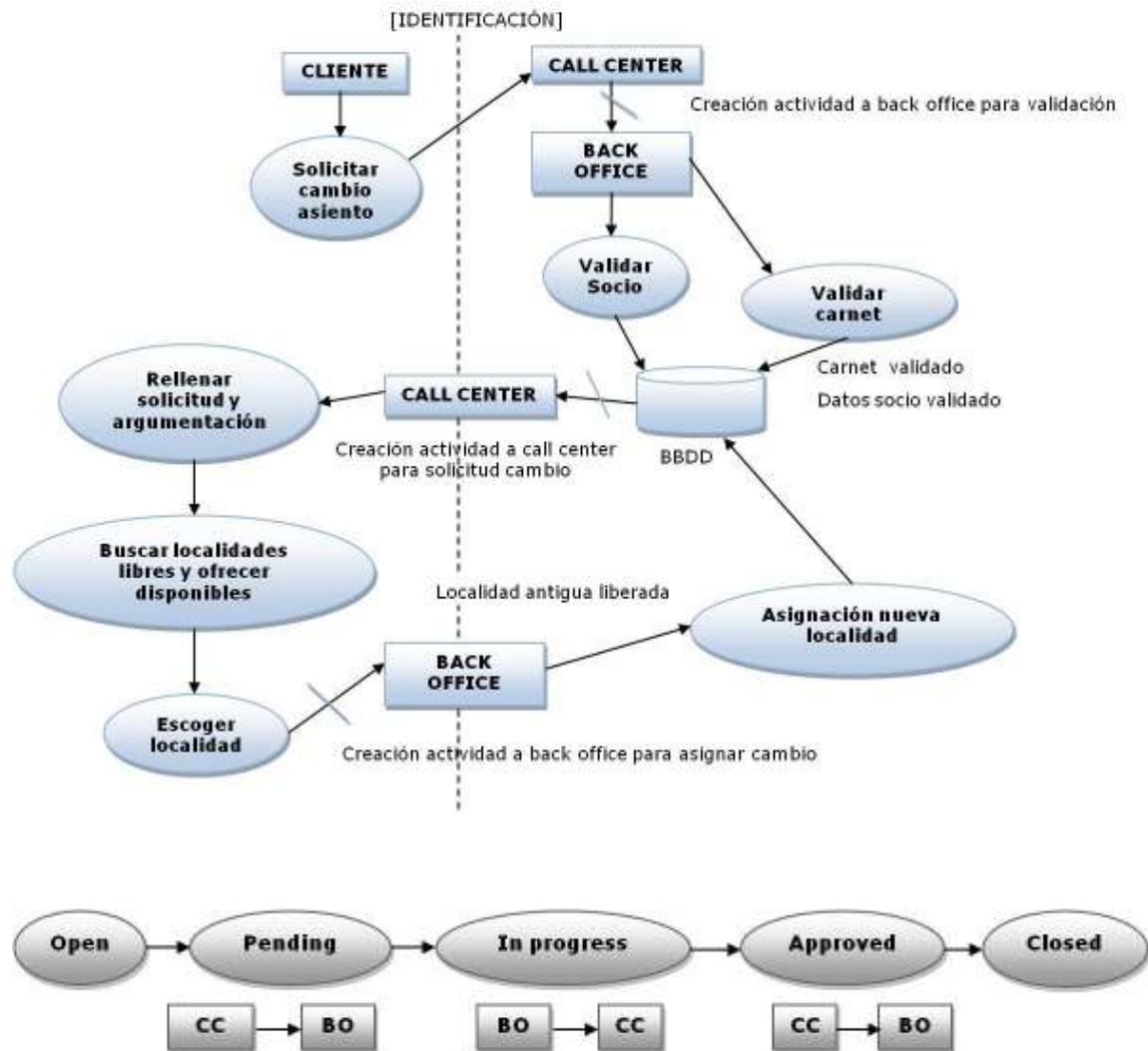


Figura 51. Duplicados

El cliente solicita un duplicado (carnet o abono) con un documento acreditativo. El call center recibe la solicitud, la crea y cede la responsabilidad al back office creándole una actividad que validará los datos del cliente y sus abonos en la base de datos. Una vez esté todo correcto, el back office crea un nuevo abono/carnet y actualiza los datos.

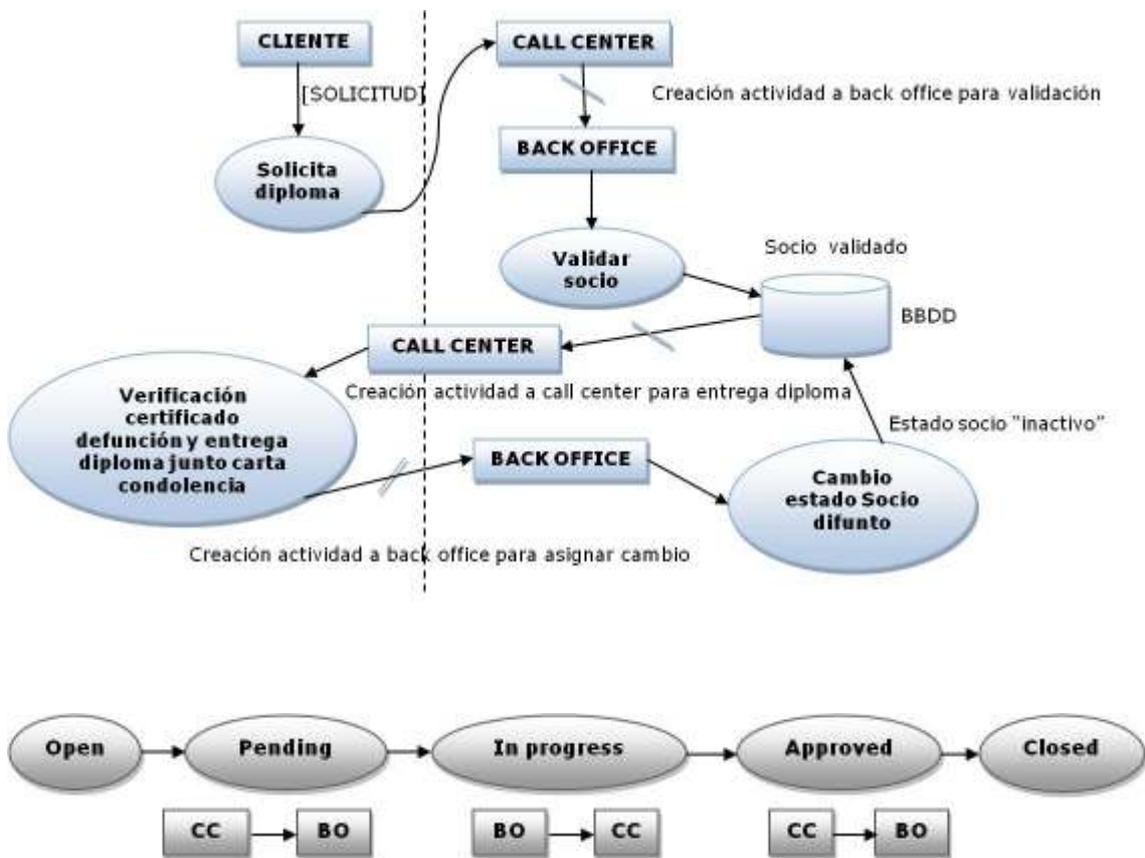
**SOLICITUD DE SERVICIO Cambio de localidad/asiente (RF4.3)**



**Figura 52. Cambio de asiento**

El cliente solicita un cambio de asiento al call center y éste crea una actividad para que el back office valide los datos. Una vez validados se cede la actividad al call center para que el cliente formalice la solicitud, se dispone a buscar localidades nuevas, y una vez escogidas el back office se encargará de realizar las modificaciones de los datos en el contrato del cliente.

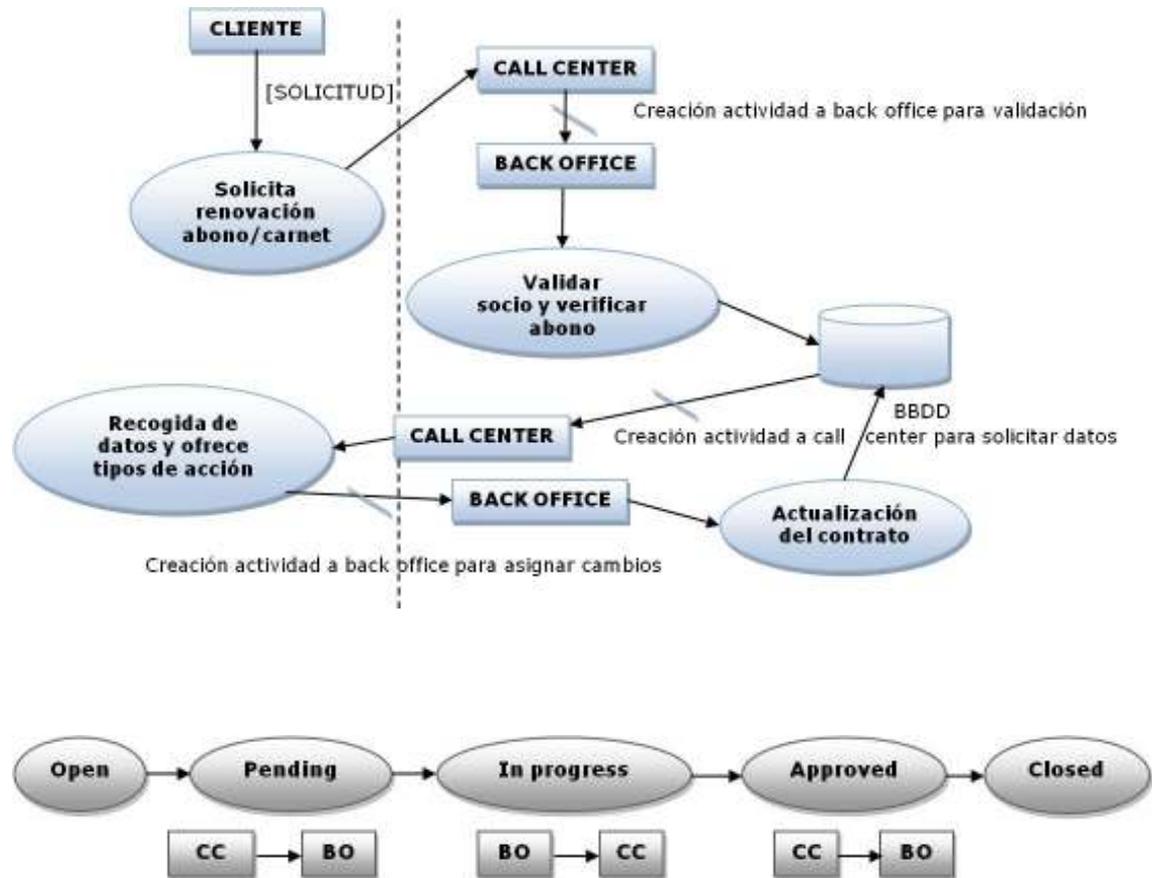
**SOLICITUD DE SERVICIO Carnet y diploma de socio difunto (RF4.4)**



**Figura 53.Socio difunto**

El familiar de un socio difunto solicita el diploma y éste crea una actividad para que el back office valide los datos. Una vez validados se cede la actividad al call center para que verifique la certificación de defunción para entregarle el diploma y la carta de condolencia a los familiares. Call center crea una actividad para que back office asigne el cambio en el contrato y cambie de estado a "inactivo" el cliente en la base de datos.

**SOLICITUD DE SERVICIO Renovaciones abonos y/ó carnets (RF4.5)**



**Figura 54. Renovaciones**

El cliente solicita una renovación. El call center crea una actividad para que el back office valide los datos. Una vez validados se cede la actividad al call center para que el cliente entregue los datos y documentación necesaria que será adjuntada y registrada. Se crea una actividad para que el back office actualice el contrato del socio indicando la nueva prórroga.

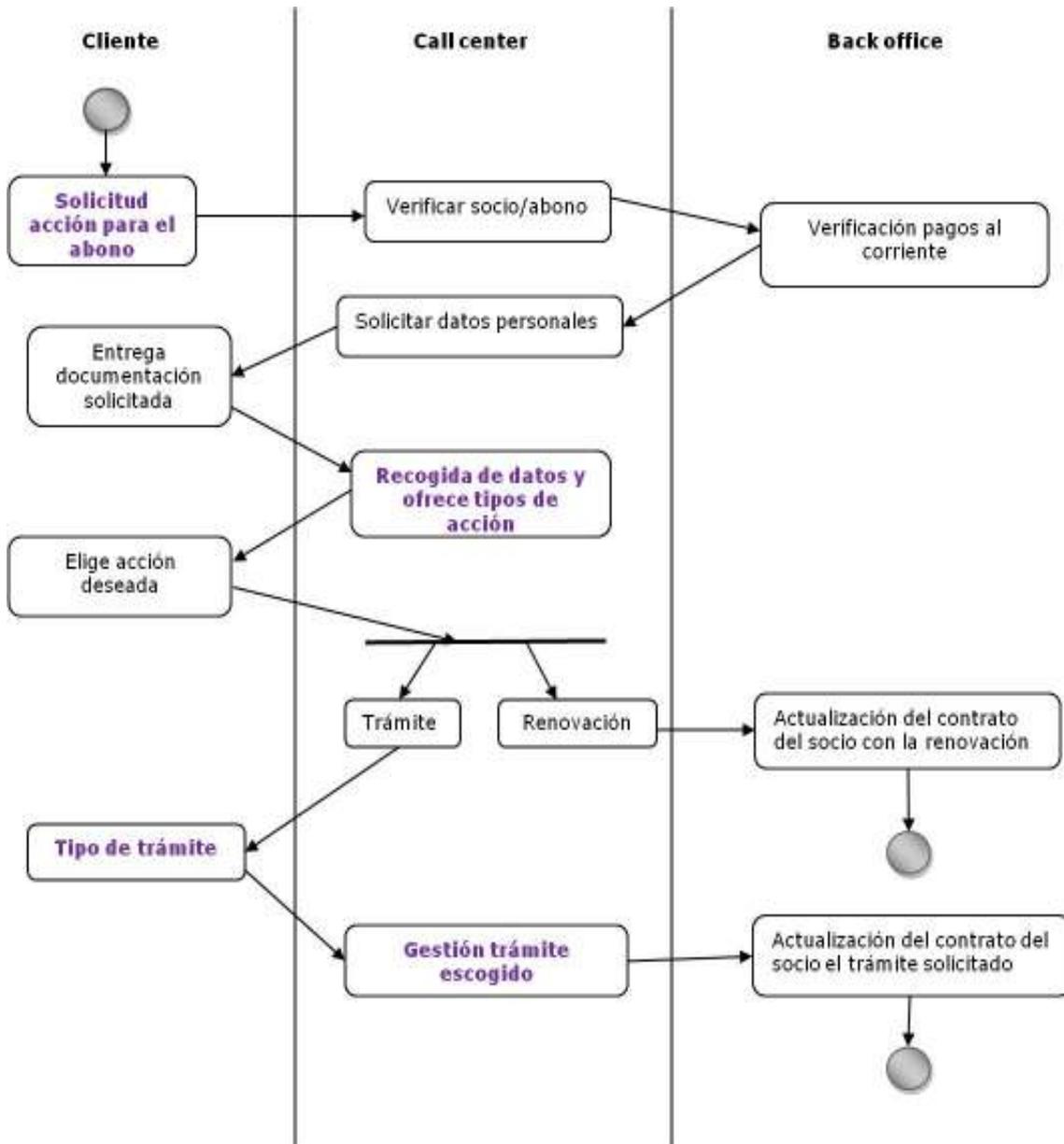
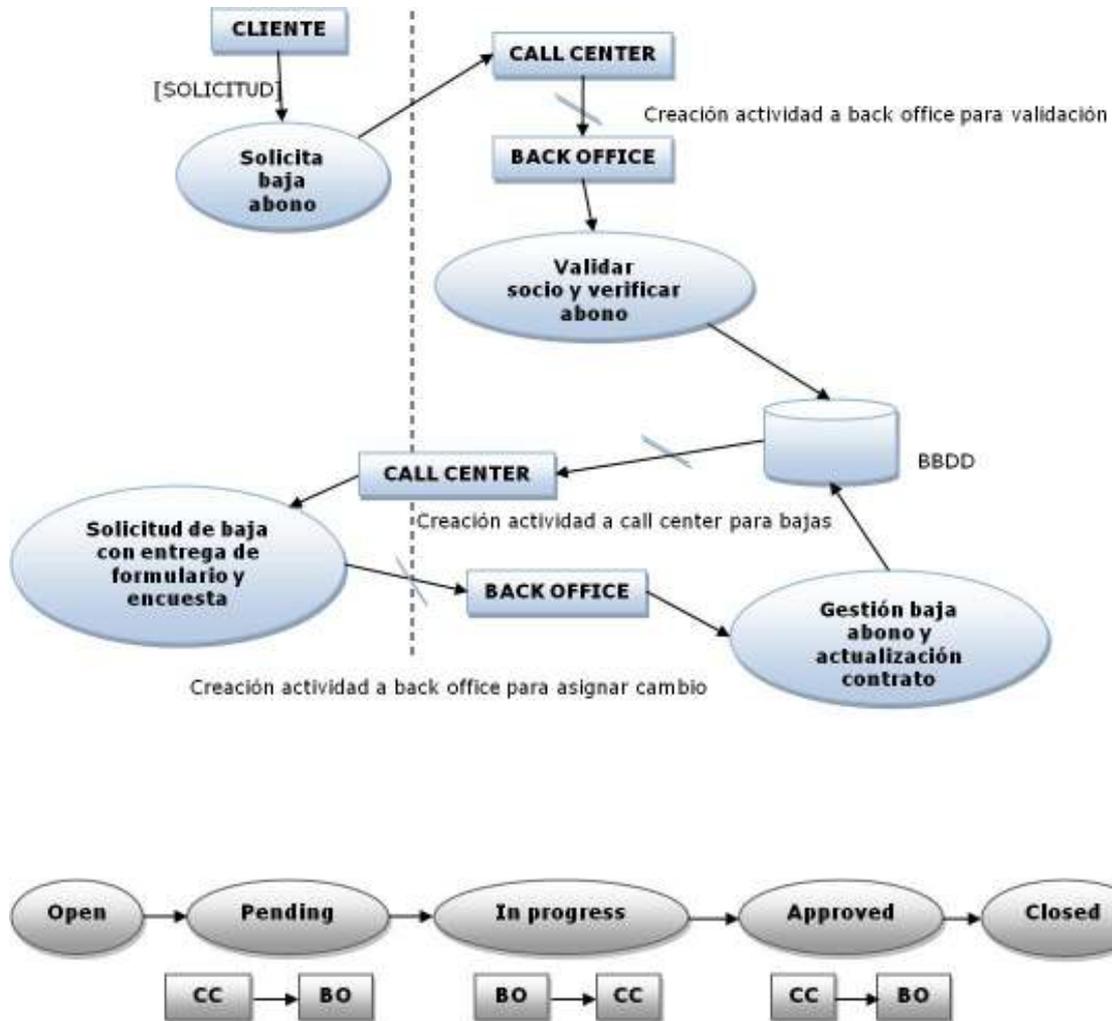


Figura 55. Renovaciones 2

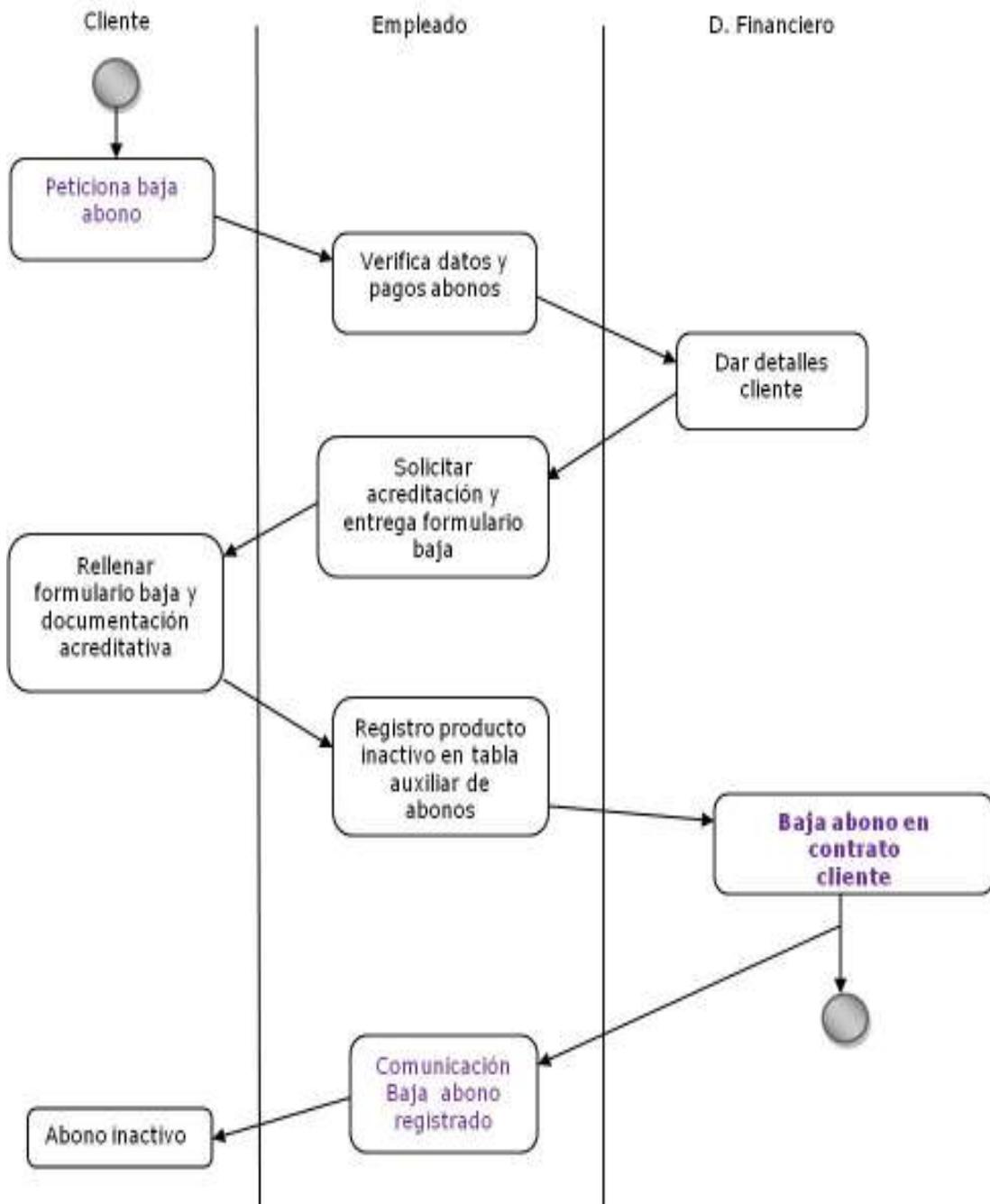
**SOLICITUD DE SERVICIO Registro de bajas de abonos (RF4.6)**



**Figura 56. Bajas abonos**

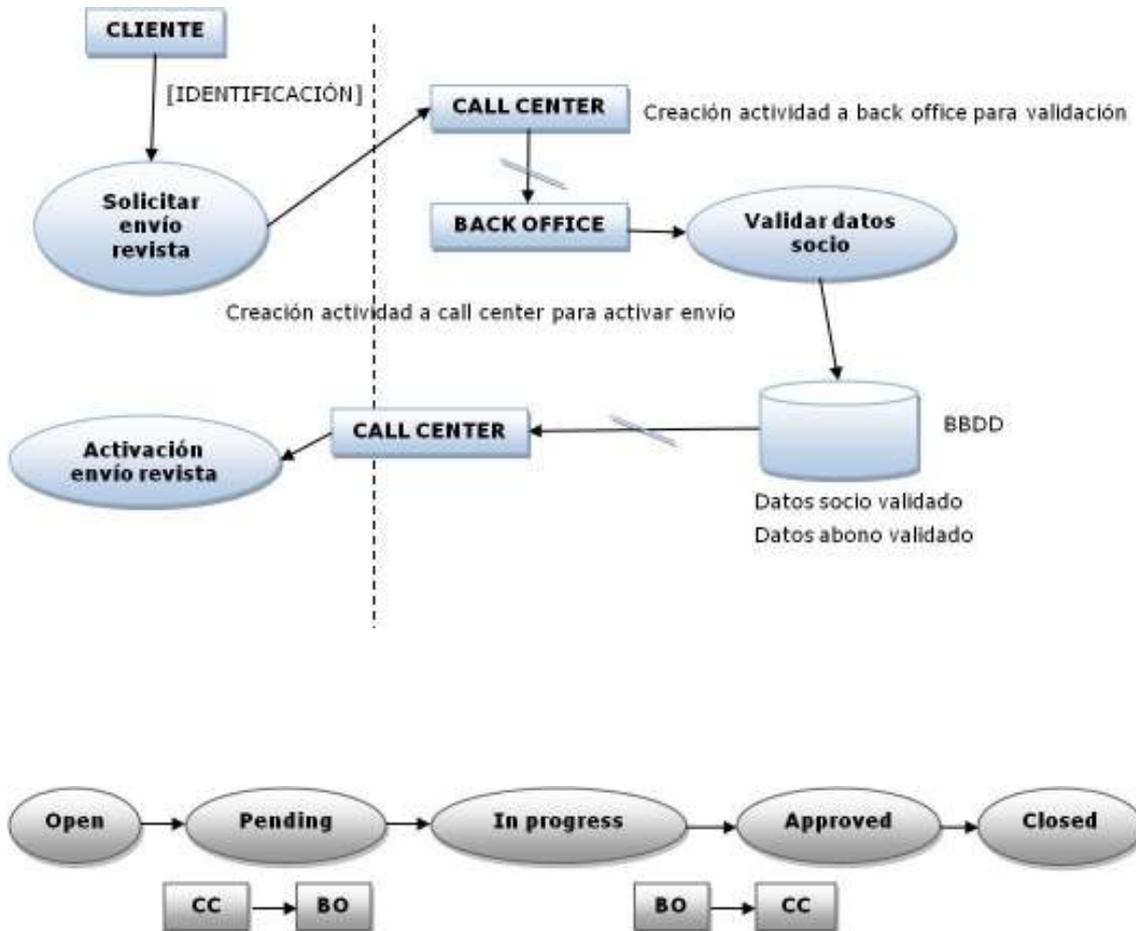
El cliente solicita una baja de un abono contratado y el call center que ha recogido la llamada crea una actividad para que el back office valide los datos del socio y de los abonos activos. Una vez validados se cede la actividad al call center para que el cliente formalice la baja y le solicita una encuesta de satisfacción para que quede registrado el motivo de ello.

Una vez recogida la información cede la responsabilidad de nuevo al back office para que inactive el abono y actualice el contrato.



**Figura 57. Bajas abonos 2**

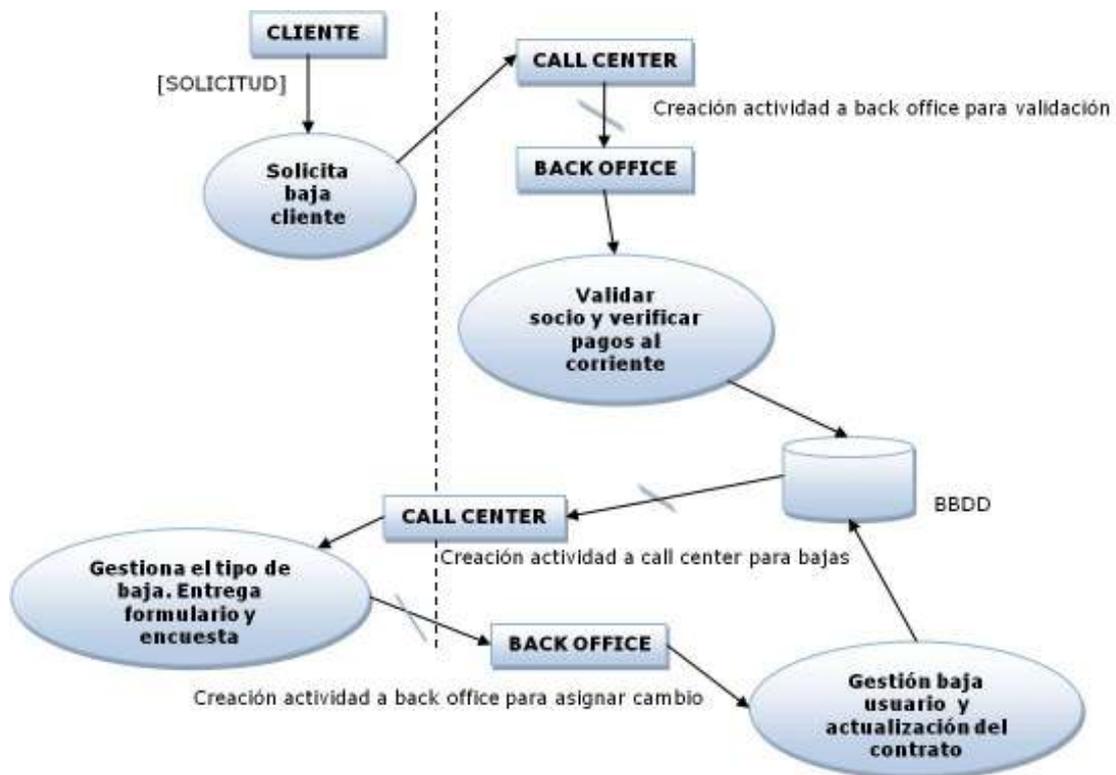
**SOLICITUD DE SERVICIO** Petición de envío de revistas (RF4.7)



**Figura 58. Petición envío revista**

El cliente solicita el envío de revistas en su domicilio atendido por el call center y éste crea una actividad para que el back office valide los datos. Una vez validados se cede la actividad al call center para que el cliente formalice la solicitud y recoga la información necesaria para activar dicho servicio.

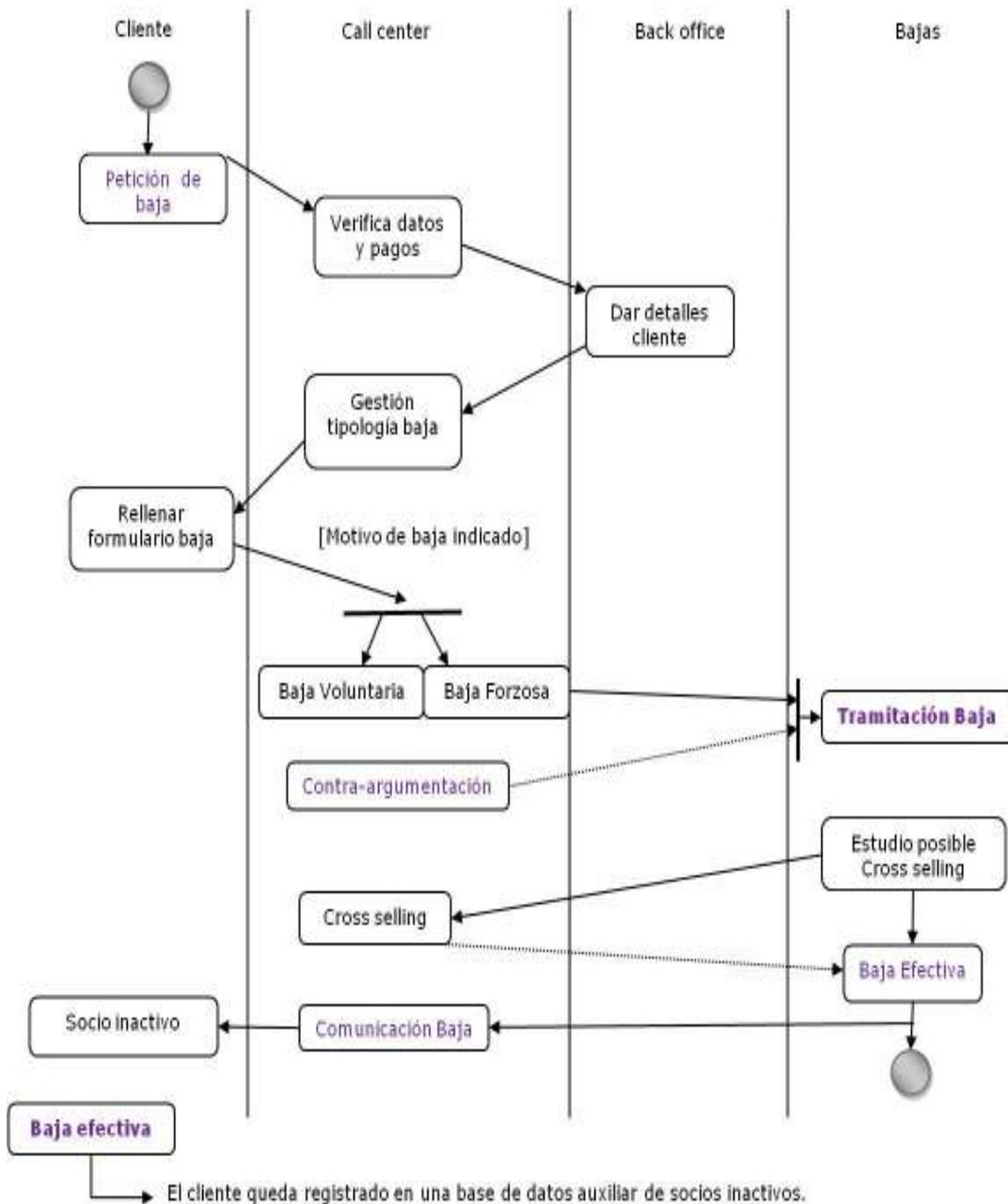
**SOLICITUD DE SERVICIO Registro de bajas de usuarios (RF5)**



**Figura 59. Bajas usuarios**

El cliente solicita una baja del club, el call center que ha recogido la llamada crea una actividad para que el back office valide los datos del socio y que los pagos estén al corriente. Una vez validados se cede la actividad al call center para que gestione la baja y le solicita una encuesta de satisfacción para que quede registrado el motivo.

Se creará una actividad para el back office con el fin de que gestione el trámite de baja actualizando el contrato e inactivando el estado del socio para que se actualice la información en la base de datos.



**Figura 60 Registro de bajas de usuarios**

## CICLO DE VIDA DE LOS ABONOS

### SOLICITUD DE SERVICIO Solicitud de abono (RF6)

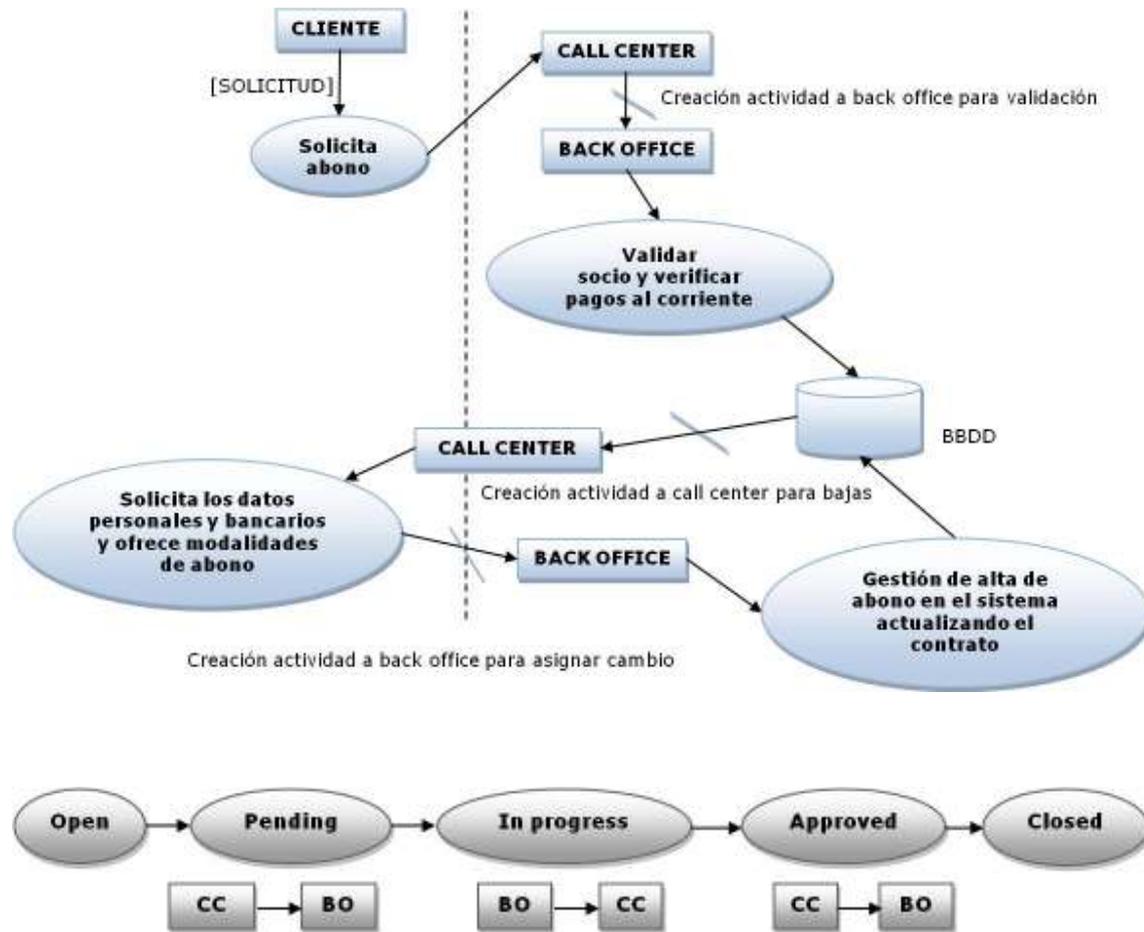


Figura 61 Solicitud de abono

El cliente solicita un nuevo abono, de manera que el call center crea una actividad al back office para que valide los datos del cliente en la base de datos correspondiente. Una vez verificada la consulta le cede la responsabilidad de nuevo al call center, creando una actividad para que efectúe el alta y se lo comunique al nuevo cliente asignándole un número de socio. El cliente quedará en estado activo.



# 5

---

# Pruebas

## 5 PRUEBAS

### 5.1 PRUEBAS DE INTEGRACIÓN

A nivel de integración de la aplicación el navegador que muestra mayor compatibilidad con el modo de interactividad alta, que es el que refleja una navegación más eficiente, es el Microsoft Internet Explorer 7.0.

Puesto que el cliente del que dispuse me lo dieron con la instalación de Siebel, no pude controlar exhaustivamente todas las posibles casuísticas, pero sí que es cierto que a raíz de realizar las primeras pruebas con la aplicación estándar empecé a encontrarme problemas de interacción con el navegador Microsoft Internet Explorer 7.0 ya que no funcionaban los menú contextuales de la aplicación, de manera que tuve que realizar un downgrade a Explorer 6.0 en modo de interactividad alta para que pudiera navegar sin ningún tipo de problema.

	Microsoft Windows 7	Microsoft Windows XP SP2	Microsoft Vista	Microsoft Windows 2000
Microsoft Internet Explorer 6.0	NO	SI	NO	SI
Microsoft Internet Explorer 7.0	NO	SI	SI	NO
Microsoft Internet Explorer 8.0	NO	SI	SI	NO
Safari 2.0.1	NO	NO	NO	NO
Mozilla Firefox	NO	SI	NO	SI

**Figura 62** Compatibilidad de navegadores y sistema operativo

## 5.2 PRUEBAS DE INSTALACIÓN

Se realizó una instalación tanto del cliente como del servidor. Para ello se debieron configurar los archivos correspondientes de la solución vertical del call center escogida para el club deportivo.

La instalación comienza con la base de datos Oracle en el servidor, seguidamente se instalan los componentes Siebel, el gateway y el servidor de la base de datos.

Por último se realiza la instalación de la herramienta de desarrollo Siebel Tools.

## 5.3 PRUEBAS DE RECUPERACIÓN

Se han realizado pruebas para garantizar una buena recuperación de la base de datos y de los objetos de negocio en caso de error catastrófico.

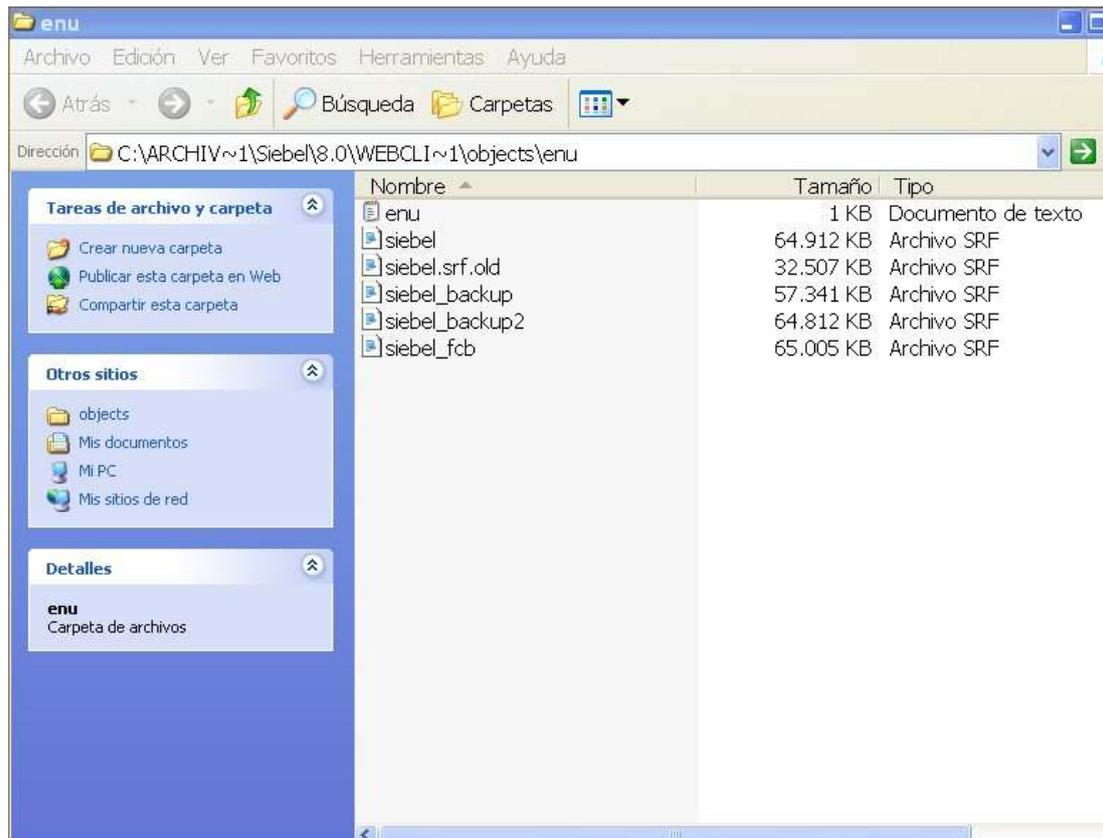
Se realizó un archivo **.bat** de copia para hacer backups del archivo de repositorio de siebel para prever posibles pérdidas en cuanto se empezaron a realizar el esqueleto de las primeras interfaces:

Se realizó un archivo.bat que sobrescribía el archivo siebel.srf por uno que he renombrado con siebel\_backup.srf

La idea es que se haga una copia del archivo repositorio.

**C:\Archivos de programa\Siebel\8.0\web client\OBJECTS\ENU\siebel.srf**

**C:\Archivos de programa\Siebel\8.0\web client\OBJECTS\ENU\siebel\_backup2.srf**



**Figura 63. Backup archivo repositorio**

El archivo .bat de backup contiene el siguiente comando:

```
copy "C:\ARCHIV~1\Siebel\8.0\web client\OBJECTS\ENU\siebel.srf"  
"C:\ARCHIV~1\Siebel\8.0\web client\OBJECTS\ENU\siebel_backup.srf"
```

## **5.4 PRUEBAS DE CONFIDENCIALIDAD**

La seguridad de la aplicación depende en su mayoría del propio sistema Oracle-Siebel CRM pero un diseño erróneo podría permitir acceder a la información a usuarios que no debieran, por lo tanto se ha comprobado que se garantiza la confidencialidad y que solamente los usuarios autorizados tienen acceso a la información que se corresponde con su perfil. Evitamos usos malintencionados.

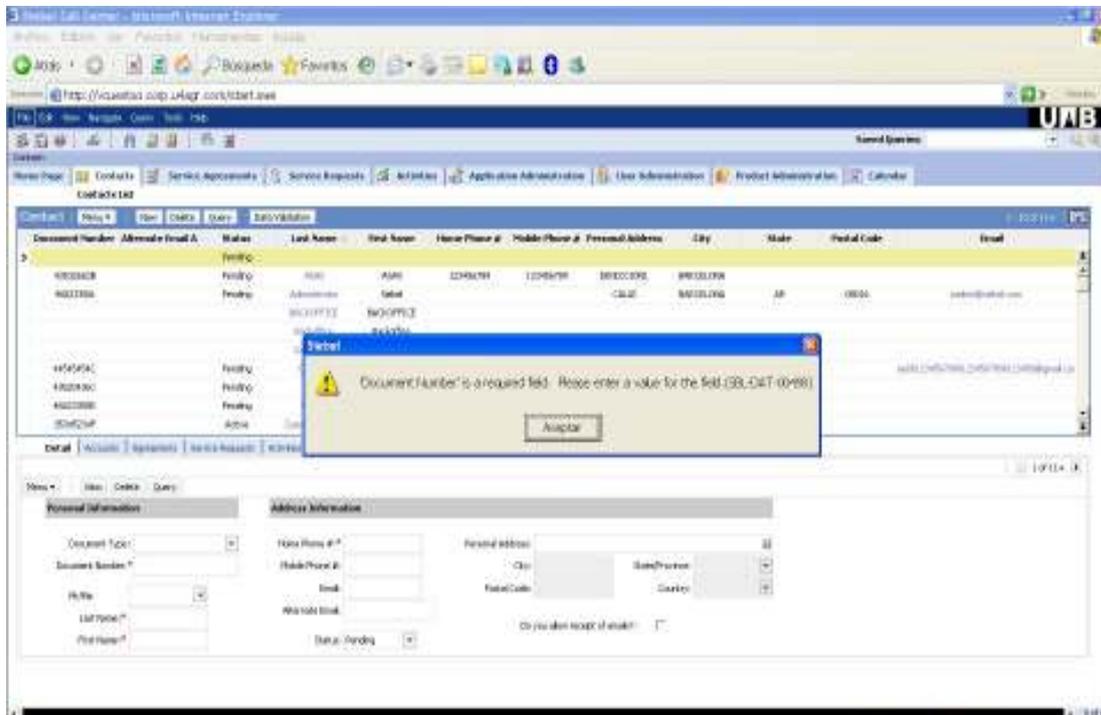
Así pues, puesto que la aplicación utiliza diferentes perfiles de usuario, se han realizado pruebas de autenticación para asegurar que cada uno puede acceder tan solo a la parte correspondiente. Para ello el usuario administrador será el responsable de asignar el acceso necesario para cada uno.

También se ha protegido la base de datos con un nombre y contraseña.

Para verificar el funcionamiento seguro del acceso, se han realizado pruebas con una autenticación falsa y con diferentes perfiles y hemos comprobado que no se

puede acceder ni con una contraseña falsa, ni con un usuario inexistente, ni que para un perfil determinado se acceda a información accesible únicamente por un perfil diferente.

## 5.5 VALIDACIONES

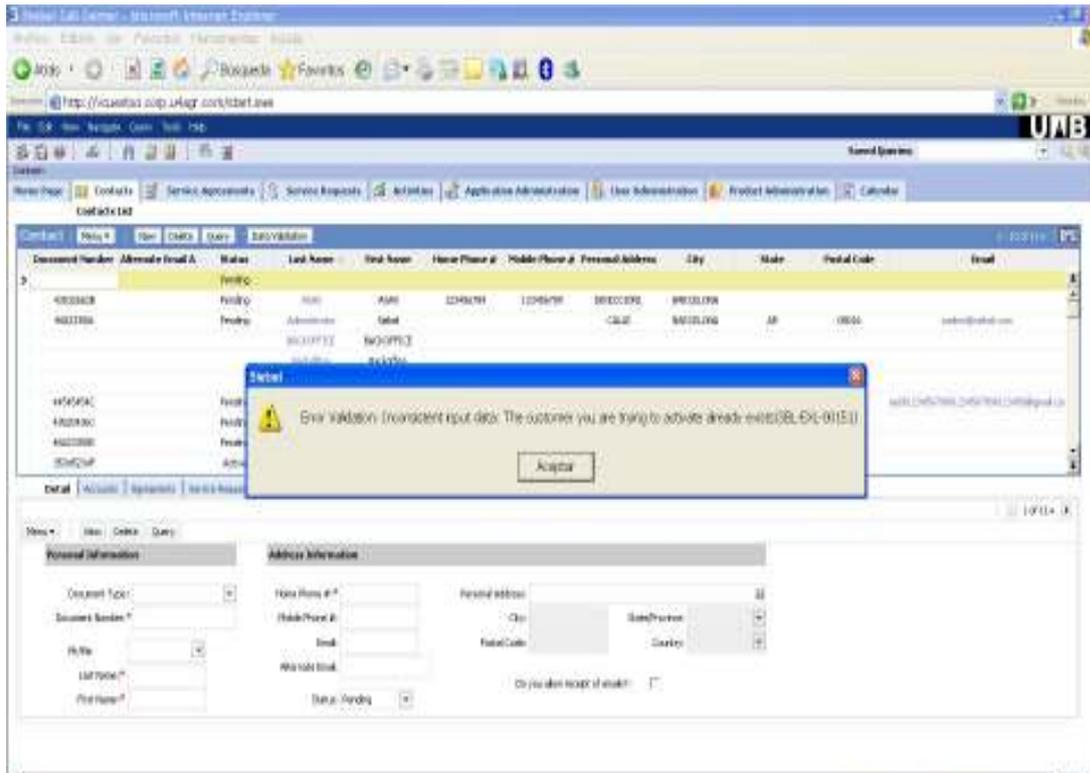


**Figura 64. Control de error DNI no introducido**

A la hora de crear contactos nuevos existe un control que verifica que el DNI es un campo requerido, con lo cual el sistema nos alertará si lo dejamos vacío.

Para la validación del DNI del cliente que estamos creando, se ha creado un business service en eScript (lenguaje propietario de Siebel)

El *business service* que se ha creado es de Validaciones Contactos, y el script correspondiente que se ha desarrollado es con un método llamado valida\_dni



**Figura 65. Control de error socio duplicado**

Puesto que se ha de realizar una validación para verificar que el usuario introduzca bien el DNI, en el applet de introducción de nuevos clientes en la base de datos, en el caso de la pantalla de contactos, para ir guardando cada registro se realiza un commit a la base de datos, es decir que se queda cada registro automáticamente posicionado, para que por defecto se almacene sin tener que “clickar” un botón, en Siebel tools la forma de realizar esto es mediante la invocación de un writerecord.

Existe un control interno para que se introduzca un DNI existente (es decir que no sea uno incorrecto/aleatorio), y que, como idea más importante a destacar, una vez hayamos introducido uno, se realizará una comprobación para verificar que no exista otro cliente con el mismo DNI, y por tanto, no existan duplicados de socio. Dicha comprobación se realizará de forma automática por defecto al cambiar de registro para crear un nuevo contacto y no será apreciable hasta que no se introduzcan dos socios con el mismo DNI, momento en que se mostrará un mensaje de error, y nos impedirá grabar el nuevo contacto.

Por otro lado, hay una verificación para que no existan duplicados de socio, y en el caso que se introduzcan dos DNIs iguales habrá un mensaje de inconsistencia de datos.

# 6

---

# Conclusiones

## 6 CONCLUSIONES

El objetivo personal de este hito ha sido alcanzado, pues he logrado un acercamiento a la tecnología Siebel y he realizado la implantación de un proyecto en su totalidad. De manera que ha sido una satisfactoria experiencia profesional.

He podido vivir cada una de las etapas en las que ha ido creciendo, desde un diseño, hasta la gestión en servidor remoto, configuración de VPN, a desarrollar y gestionar una base de datos de Oracle.

Al concluir el proyecto se puede decir que se han intentado alcanzar todos los requisitos solicitados, pero que por inexperiencia en la gestión de proyectos, y por un leve retraso, se decidió dejar en un segundo nivel los procesos y funcionalidades que afectan al ciclo de los abonos de los socios del club deportivo.

De manera que se puede poder a plantear un segundo prototipo de este aplicativo en líneas futuras. No obstante el grado de satisfacción es alto ya que se han alcanzado otras funcionalidades importantes.

### 6.1 ALCANCE INICIAL DEL PROYECTO

A continuación detallo la propuesta del alcance del proyecto a implementar:

Se han seleccionado las siguientes funcionalidades extraídas del documento "Requerimientos - CRM Club Deportivo" como alcance de este proyecto. Cada funcionalidad ha sido enfocada bajo un prisma Siebel para que la implementación del proyecto pueda resultar en un paquete completo, dentro de sus limitaciones.

#### CICLO DE VIDA DEL SOCIO

##### 1. *Solicitud de alta con los datos del cliente y datos bancarios*

Uso de la entidad CONTACT, *vanilla*<sup>4</sup> en Siebel, permitir que el Empleado pueda crear nuevos contactos. Entre los datos del contacto (Nombre, Apellidos, DNI, Dirección, Número de Socio, Motivo de Baja (únicamente obligatorio si el estado de socio está relacionado con la baja), etc.) se incluirían los datos bancarios utilizados para domiciliar el pago de todos los productos contratados con el Club.

Al crearse el nuevo registro de socio (cliente) el estado del registro quedará como "Pendiente", a la espera de que el alta se valide.

##### 2. *Validación de datos (no existan duplicados de socio).*

Esta funcionalidad está integrada dentro de Siebel y viene por la User Key a nivel de columna (PERSON UID de la S CONTACT). En esa columna habría que guardar el

---

<sup>4</sup> *Vanilla*: de origen

valor que se usará como USER KEY para controlar la integridad de la USER KEY.  
Valor propuesto: DNI

### *3. Aceptación por parte del club.*

Una vez se considere que el registro de socio es válido, el Empleado cambiará el estado del socio a "Activo". El estado del cliente estará controlado por un modelo de estados. Al "Activarse" el cliente, se le asignará automáticamente un número de socio.

### *4. Realización de trámites:*

- a. Actualización foto del carnet de socio*
- b. Duplicado de abono(s)*
- c. Duplicado del carnet de socio*
- d. Cambio de localidad*
- e. Carnet y diploma de socio difunto*
- f. Renovaciones abonos*
- g. Renovaciones carnes*
- h. Bajas abonos*
- i. Nuevos abonos*
- j. Petición envío revista*

Todos estos trámites serán tratados como incidencias por parte de los Empleados y relacionados con el socio en concreto. El Empleado, una vez trabaje en esas incidencias, creará una Actividad, conectada con la incidencia que toque, en la que recogerá todo el trabajo realizado en el cliente que toque.

### *5. Registro de Bajas*

- a. Actualmente la baja de socio es únicamente una baja administrativa. Con el nuevo sistema se quiere gestionar diferentes tipologías de baja que pueden tener diferentes procesos (ej. Bajas voluntarias o forzosas). En ciertos procesos se podrán definir diferentes estados de una baja e intentar fidelizar durante el proceso de baja, es decir, intentar evitarla, y en caso de no poder realizar cross-selling para mantener la relación cliente – Club*
- b. Registrar los diferentes motivos de causa de la baja. Una posible relación de pasos del proceso de baja puede ser: Solicitud -Contra argumentación para evitar la baja –Tramitación -Cross selling-Motivo de la baja –Baja efectiva -Comunicación.*

El Empleado modificará consecutivamente la baja del Socio con diferentes estados (Petición de Baja, Tramitación de Baja, Crossselling, Baja Efectiva). En todos ellos el campo de MOTIVO DE BAJA será obligatorio de rellenar.

## **CICLO DE VIDA DE LOS ABONOS**

1. *Solicitud del abono*
2. *Recogida de los datos personales, datos de contacto, tipo de abono solicitado o preferencias*  
*Existen abonos de temporada para ( a detallar en el análisis funcional):*
  - *Estadio Principal de fútbol*
  - *Estadio Secundario de fútbol*
  - *Estadio de Baloncesto*
  - *Parking*
3. *Diferentes acciones del ciclo de vida de los abonos:*
  - a. *Renovaciones*
  - b. *Trámites*
4. *Registro de la baja*

Tanto los abonos como el carné de socio se considerarán como productos de Siebel, conectados con el cliente mediante un Contrato. Cada Socio tendrá un único contrato con 1...N productos.

Cada funcionalidad ha sido enfocada bajo un prisma Siebel para que la implementación del proyecto pueda resultar en un paquete completo, dentro de sus limitaciones.

## **6.2 RELACIÓN DE PROBLEMAS APARECIDOS Y NO IDENTIFICADOS PREVIAMENTE**

Por insuficientes características de rendimiento en los componentes hardware que conforman el servidor que he utilizado para el desarrollo de mi aplicación me vi con diversos problemas, ya que la base de datos y los servicios de Oracle debían de estar levantados para que mi cliente dedicado funcionara, estos consumían el rendimiento total de ésta.

Mencionar, que se tuvieron que deshabilitar servicios de Windows que eran necesarios para que Siebel funcionara correctamente. Y un servidor que soporte correctamente estos servicios junto a los de la base de datos requiere de más capacidad de RAM y procesador.

## **6.3 CUMPLIMIENTO DE OBJETIVOS**

Tanto el objetivo personal como el profesional han sido bastante satisfactorios, pues me ha servido como experiencia para familiarizarme con el entorno de proyectos en consultoría y, a nivel profesional, he intentado ajustarme

estrechamente a los requerimientos funcionales que se me solicitaron junto a las diferentes revisiones y cambios que me solicitaba mi tutor de proyecto.

Dados los objetivos iniciales, hay algunas funcionalidades a nivel de procesos de negocio que han quedado justamente desarrolladas por falta de experiencia a nivel de planificación y que no han podido resolverse por falta de tiempo, pero que sin embargo hemos podido llegar a percibir y poder llegar a estimar una valoración al tiempo a emplear.

En los requerimientos funcionales nos piden poder procesar bajo el punto de vista de un club deportivo los ciclos de la vida de un socio y la de un abono, y puesto que hemos priorizado los conceptos a nivel de socio, mostramos una funcionalidad con mucho menos complejidad que de la esperada.

## 6.4 MODIFICACIONES EN EL ALCANCE Y ALCANCE DEFINITIVO DEL PROYECTO

Inicialmente se me proporcionó un modelo de datos que a medida que se ha ido desarrollando el proyecto me he visto en la necesidad de cambiar ligeramente para que la configuración fuera ajustada y adecuada con los objetos que me ofrecía la herramienta de desarrollo Siebel de Oracle: Siebel Tools.

Una de las modificaciones a nivel de objeto de negocio repercute en los productos, ya que se ha modificado para que existan diferentes líneas de producto (asientos y carnets) replanteando el uso de las tablas de la base de datos, puesto que inicialmente existían sólo productos que dependían de los contratos directamente. La nueva tabla incorporada en el modelo de datos es la S\_PROD\_LN.

## 6.5 RETRASOS EN LA PLANIFICACIÓN

### 6.5.1 Cronograma real y desviación

Tarea	Descripción	Horas	Desviación
1	Formación	40	150
2	Investigación	60	100
3	Análisis	60	75
4	Diseño	110	25
5	Codificación	210	105
6	Pruebas unitarias	40	30
7	Documentación	40	85
8	Defensa del proyecto	10	10
<b>TOTAL</b>		<b>560</b>	<b>580</b>

Figura 37 Cronograma real

Por falta de experiencia en la planificación de proyectos, se ha decidido tratar el ciclo de abonos en menos profundidad, y se puede considerar una futura línea de ampliación para próximas versiones de este proyecto.

## 6.6 LÍNEAS DE AMPLIACIÓN

No se prevén actualizaciones en este prototipo a no ser que haya cambios en los requerimientos funcionales iniciales, o bien se profundice en los procesos del ciclo de los abonos o en algún punto diferente del modelo de gestión contemplado.

También se ha de tener en cuenta que la escalabilidad de la aplicación es totalmente alcanzable ya que a diferencia de otros entornos de desarrollo, se puede bloquear un proyecto o bien, se desarrolla a nivel de objetos, permitiendo su modificación independientemente del proyecto. Es más, es un desarrollo que favorece la concurrencia de usuarios, a pesar de que en mi caso se trataba de un desarrollo individual y no he tenido que contemplar esa casuística.

Como comento en puntos anteriores se podría considerar una ampliación importante la de diseñar el ciclo de abonos y su correcto funcionamiento, de manera que una segunda versión en el caso que la hubiera podría estar centrada en este ítem de forma más extensa.

## 6.7 VALORACIÓN FINAL

Para concluir la presente memoria realizaré una valoración personal para dar mi impresión sobre el desarrollo de lo que ha significado para mí este proyecto.

Puedo garantizar que la idea planteada desde un principio sobre la línea que iba a tener mi proyecto no se corresponde con la aplicación resultante, pues ha sido para mí un reto personal ir superando cada dificultad que se iba añadiendo a diario y que no siempre he tenido los conocimientos iniciales necesarios. De manera que queda implícita la tarea de investigación añadida de aprender una tecnología totalmente nueva para mí y de solventar los problemas que han ido apareciendo.

Por otro lado dada la naturaleza de mi perfil técnico al principio me costó adaptarme a ver las bases de datos y los objetos referenciados bajo un prisma de negocio y a traducir cada una de las funcionalidades como un proceso y/o gestión, y por lo tanto, valoro de forma positiva el resultado final que se presenta como proyecto final de carrera ya que supera mis expectativas.

Gracias a la implantación de este proyecto he mejorado mis conocimientos de programación en script y de consultas de bases de datos y me ha permitido lograr un acercamiento a la tecnología Siebel. Puedo decir que he ampliado mis conocimientos adquiridos durante mi etapa como estudiante referente a gestión de bases de datos y servidores remotos.

Por toda esta valoración puedo concluir que el resultado final de este nuevo hito ha sido totalmente satisfactorio y me ha preparado un poco más a encarar mi nueva etapa profesional en el mundo laboral que nos espera.

# 7

---

# Bibliografía

## 7 BIBLIOGRAFÍA

### 7.1 FUENTES ELECTRÓNICAS

#### Website de Oracle

- ❑ <https://support.oracle.com/CSP/ui/flash.html> (Siebel support )
- ❑ <http://www.oracle.com/es/index.html> (Website de Oracle)

#### Foros de consulta de errores de Siebel:

- ❑ <http://www.siebelbook.com> (comunidad profesional Siebel)
- ❑ <http://siebel.ittoolbox.com/> (comunidad profesional IT)

#### Fuente de información general:

- ❑ [www.google.es](http://www.google.es)

### 7.2 FUENTES BIBLIOGRÁFICAS

Para la realización de este proyecto me he apoyado en las siguientes fuentes de información:

- Consultores de U4 especializados en Siebel CRM con conocimientos tecnológicos avanzados.
- Documentación proporcionada:
  - Propuesta Alcance Proyecto Siebel.
  - Propuesta de Modelo de Datos.
  - Estructura Documentación Apoyo Proyecto Siebel v.02
- Oracle METALINK: herramienta web Oracle de soporte que permite reportar y seguir incidencias en una base de datos de conocimiento para resolver dudas y gestionar las actualizaciones y parches.  
Se trata de un foro de expertos en el cual poder contrastar opiniones referentes a problemas de Oracle.

-----  
Firmado: **Verónica Cuesta Martínez**

## **8 ANEXOS**

Se adjunta un CD-Rom con la presente memoria y los siguientes contenidos:

1. Archivo srf (siebel repository file) con el desarrollo del proyecto del club deportivo: siebel.srf
2. Propuesta Alcance Proyecto Siebel.doc
3. Propuesta de Modelo de Datos.doc
4. Requerimientos- CRM Club Deportivo.doc
5. Project Plan.mpp
6. Vistas y responsabilidades.xls
7. Memoria del proyecto "Implantación CRM.PDF"

## **Anexo A: CODIFICACIÓN**

### **BUSINESS SERVICE: XXX Creaciones Actividades SR -Script**

```
function buscar_login(id)
{
    var bo=TheApplication().GetBusObject("Users");
    var bc=bo.GetBusComp("User");
    var login="";

    bc.SetViewMode(AllView);
    bc.ActivateField("Login Name");
    bc.ClearToQuery;
    bc.SetSearchExpr("[Id] =" + id + "");
    bc.ExecuteQuery(ForwardOnly);
    if (bc.FirstRecord())
    {
        login=bc.GetFieldValue("Login Name");
    }
    return(login);
}
```

```
function cambio_SR (Inputs, Outputs)
{
    var in_id = Inputs.GetProperty("idSR");
    var tipoSR = Inputs.GetProperty("SR Type");

    var estado_actual = "";
    var id_propietario = "";
    var propietario = "";
    var propietario_id = "";
    var resultado = "OK";
    var err_desc = "";
    var cambio_prop="N";
    var nuevo_propietario="";
    var bo=TheApplication().GetBusObject("XXX Service Request");
    var bc=bo.GetBusComp("XXX Service Request");

    bc.SetViewMode(AllView);
    bc.ActivateField("Status");
    bc.ActivateField("Description");
    bc.ActivateField("Owner");
    bc.ActivateField("Owned By Id");

    bc.ClearToQuery;
    bc.SetSearchSpec("Id", in_id);
    bc.ExecuteQuery(ForwardOnly);
    if (bc.FirstRecord)
    {
```

```
estado_actual = bc.GetFieldValue("Status");
propietario = bc.GetFieldValue("Owner");
propietario_id = bc.GetFieldValue("Owned By Id");
// descripcion = bc.GetFieldValue ("Description");

if (estado_actual == "Open")
{
    bc.SetFieldValue("Status", "Pending");
}

if (estado_actual == "Pending")
{
    bc.SetFieldValue("Status", "In Progress");
    if (propietario_id == "1-AQSE")// CC
    {
        bc.SetFieldValue ("Owned By Id", "1-AQX3");
        nuevo_propietario="1-AQX3";
        cambio_prop="Y";
    }
}

if (estado_actual == "In Progress")
{
    if((tipoSR!="Actualización foto
carnet")&&(tipoSR!="Duplicado de carnet de
socio")&&(tipoSR!="Duplicado de abonos"))
    {
        bc.SetFieldValue("Status", "Approved");
    }
    else
```

```
{  
  
    bc.SetFieldValue("Status", "Closed");  
  
}  
  
if (tipoSR == "Cambio de localidad" || tipoSR ==  
"Carnet socio difunto" || tipoSR == "Bajas abonos" ||  
tipoSR == "Registro de bajas usuarios" || tipoSR ==  
"Solicitud de abono" || tipoSR == "Registro de bajas  
abonos")  
  
{  
  
    if (propietario_id == "1-AQX3")// BO  
  
    {  
  
        bc.SetFieldValue ("Owned By Id", "1-AQSE");  
  
        nuevo_propietario="1-AQSE";  
  
        cambio_prop="Y";  
  
    }  
  
}  
  
}  
  
  
if (estado_actual == "Approved")  
  
{  
  
    if (tipoSR == "Alta usuario" || tipoSR == "Petición  
envío revista")  
  
    {  
  
        if (propietario_id == "1-AQX3")  
  
        {  
  
            bc.SetFieldValue ("Owned By Id", "1-AQSE");  
  
            nuevo_propietario="1-AQSE";  
  
            cambio_prop="Y";  
        }  
  
    }  
  
}
```

```
else
{
//if (tipoSR == "Cambio de localidad" || tipoSR ==
"Carnet socio difunto" || tipoSR == "Bajas abonos" ||
tipoSR == "Registro de bajas usuarios" || tipoSR ==
"Solicitud de abono" || tipoSR == "Registro de bajas
abonos" )

    if (propietario_id == "1-AQSE")// CC
    {

        bc.SetFieldValue ("Owned By Id", "1-AQX3");
        nuevo_propietario="1-AQX3
        cambio_prop="Y";

    }

}

bc.SetFieldValue("Status", "Closed");
}
bc.WriteRecord();
}
else
{
    resultado="KO";
}
Outputs.SetProperty("resultado", resultado);
Outputs.SetProperty("cambio_prop", cambio_prop);
Outputs.SetProperty("nuevo_propietario", nuevo_propietario);
bc=null;
bo=null;
}
```

```
function crea_actividades_SR (Inputs, Outputs)
{
    try
    {
        var id_SR_ACT=Inputs.GetProperty("idSR");
        var descrip_SR= Inputs.GetProperty("Description");
        var tipoSR = Inputs.GetProperty("SR Type");
        var nuevo_propietario=Inputs.GetProperty("nuevo_propietario");
        var login="";

        login=buscar_login(nuevo_propietario);

        //var OWNER=Inputs.GetProperty("Propietario");

        var resultado="OK";
        var bo=TheApplication().GetBusObject("XXX Service Request");
        var bc=bo.GetBusComp("XXX Activity");
        bc.ActivateField("Activity SR Id");
        bc.ActivateField("Description");
        bc.ActivateField("Primary Owned By");

        bc.NewRecord(NewAfter);
        bc.SetFieldValue("Activity SR Id", id_SR_ACT);
        bc.SetFieldValue("Description", descrip_SR);
        //bc.SetFieldValue("Owned By", nuevo_propietario);
        bc.SetFieldValue("Primary Owned By", login);
        //bc.SetFieldValue("Owner",OWNER); //Podria necesitar un "Pick"
```

```
        bc.WriteRecord();

        bc.GetFieldValue("Id");

        Outputs.SetProperty("resultado", resultado);

    }

    catch(e)

    {

//        TheApplication().RaiseErrorText(e.errText)

        resultado="KO";

        Outputs.SetProperty("resultado", resultado);

    }

    finally

    {

        bc=null;

        bo=null;

    }

}

function existe_actividad(SRId)

{

    var bo=TheApplication().GetBusObject("XXX Service Request");

    var bc=bo.GetBusComp("XXX Activity");

    var existe="FALSE";

    bc.SetViewMode(AllView);

    bc.ActivateField("Activity SR Id");

    bc.ClearToQuery;

    bc.SetSearchExpr("[Activity SR Id] =" + SRId + " AND ([Status] <> 'Close') AND ([Status] <> 'Declined')");

    bc.ExecuteQuery(ForwardOnly);

    if (bc.FirstRecord)
```

```
{
    existe="TRUE";
}
return(existe);
}
function Service_PreInvokeMethod (MethodName, Inputs, Outputs)
{
    if(MethodName=="Siguiente")
    {
        cambio_SR(Inputs, Outputs);
        return (CancelOperation);
    }
    if(MethodName=="crea_actividades_SR")
    {
        crea_actividades_SR(Inputs, Outputs);
        return (CancelOperation);
    }
    return (ContinueOperation);
}
```

### **XXX Service Request Detail Applet:**

```
function WebApplet_PreInvokeMethod (MethodName)
{
    if (MethodName == "Siguiente")
    {
        this.BusComp().WriteRecord();

        var bservice = TheApplication().GetService("XXX Creaciones
        Actividades SR");

        var Inputs = TheApplication().NewPropertySet();
        var Outputs = TheApplication().NewPropertySet();

        var idSR = this.BusComp().GetFieldValue("Id");
        var desc = this.BusComp().GetFieldValue("Description");
        var tipoSR = this.BusComp().GetFieldValue("SR Type");
        var estado = this.BusComp().GetFieldValue("Status");

        var resultado = "";
        var resultado_crear_activ = "";
        var cambio_prop="N";
        var nuevo_propietario="";

        Inputs.SetProperty("idSR", idSR);
        Inputs.SetProperty("SR Type", tipoSR);
        Inputs.SetProperty("Description", desc);

        bservice.InvokeMethod("Siguiente", Inputs, Outputs);
        resultado=Outputs.GetProperty("resultado");
```

```
cambio_prop=Outputs.GetProperty("cambio_prop");
if (resultado == "OK" && cambio_prop == "Y")
{
    nuevo_propietario=Outputs.GetProperty("nuevo_propietario");
    Inputs.SetProperty("nuevo_propietario", nuevo_propietario);
    bservice.InvokeMethod("crea_actividades_SR", Inputs,
    Outputs);
    resultado_crear_activ=Outputs.GetProperty ("resultado");
    if (resultado_crear_activ != "OK")
    {
        TheApplication().RaiseErrorText(" It is not possible to
        create the activity in the Service Request. Please
        contact the system administrator.");
    }
}
else
{
    if (resultado!="OK")
    {
        TheApplication().RaiseErrorText(" It is not possible to
        change the Service Request Status. Please contact the
        system administrator.");
    }
}
this.BusComp().InvokeMethod ("RefreshRecord");
this.BusObject().GetBusComp("XXX Activity").ClearToQuery;
return (CancelOperation);
}
return (ContinueOperation);
}
```

```
function WebApplet_PreCanInvokeMethod (MethodName, &CanInvoke)
{
    if (MethodName == "Siguiente")
    {
        if (this.BusComp().GetFieldValue("Status") != "Closed")
        {
            CanInvoke = "TRUE";
            return( CancelOperation );
        }
    }
    return (ContinueOperation);
}
```

## Anexo B: PROPUESTA DE MODELO DE DATOS

