



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

**Implementación de un sistema de  
gestión de asociaciones, basado en  
SuiteCRM.**

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Diego Miguel Tamarit

**Tutor:** Joan Fons i Cors

**Cotutor Empresa:** Raúl García González

2015-2016



# Resumen

---

Ticnux es una empresa dedicada a la implantación, configuración y la realización de desarrollos para SuiteCRM. Mediante el uso de tecnologías OpenSource, en este proyecto se ha diseñado y desarrollado un paquete para SuiteCRM que añade los módulos necesarios para realizar las tareas de gestión básicas y un sistema de facturación, adaptable a distintos modelos de asociación.

**Palabras clave:** SuiteCRM, OpenSource, Asociación, Gestión



# Tabla de contenidos

---

1	Introducción.....	7
1.1	Contexto.....	7
1.2	Problemática.....	7
1.3	Objetivos.....	7
1.4	Metodología.....	8
2	Contexto tecnológico.....	10
2.1	Aproximaciones similares.....	10
2.2	Tecnologías relevantes.....	11
2.2.1	PHP.....	12
2.2.2	Smarty.....	13
2.2.3	JQuery.....	13
2.2.4	MySQL.....	13
2.2.5	Apache.....	14
3	Análisis y Diseño.....	15
3.1	Funciones del producto.....	15
3.1.1	Gestión de asociados.....	16
3.1.2	Gestión de cargos.....	17
3.1.3	Facturación.....	17
3.2	Características del usuarios.....	18
3.2.1	Usuario Normal.....	18
3.2.2	Usuario Administrador del Sistema.....	18
3.3	LOPD.....	18
3.4	Seguridad.....	18
3.5	Especificación por módulos.....	19
3.5.1	Asociados.....	20
3.5.2	Familias.....	23

3.5.3	Altas y bajas.....	24
3.5.4	Sanciones.....	25
3.5.5	Histórico de cambios.....	25
3.5.6	Órganos.....	26
3.5.7	Cargos.....	26
3.5.8	Organización.....	26
3.5.9	Pagos.....	27
3.5.10	Facturas.....	27
3.5.11	Cuotas.....	28
3.5.12	Remesas.....	31
3.5.13	Administración.....	32
3.5.14	Tareas programadas.....	33
4	Implementación.....	35
4.1	Estrategias de implementación.....	35
4.2	SuiteCRM framework.....	36
4.2.1	Manifest.php.....	36
4.2.2	Logichooks.....	36
4.2.3	SugarController.....	38
4.2.4	SugarView.....	39
4.3	Herramientas de SuiteCRM.....	41
4.3.1	Constructor de módulos.....	41
4.3.2	Estudio.....	42
4.4	Detalles del desarrollo.....	42
4.4.1	Facturación.....	42
4.4.2	Cambios realizados.....	46
4.4.3	Fichero de log.....	46
5	Casos de Estudio.....	47
5.1	Club deportivo.....	47
5.1.1	Necesidades de la asociación.....	47



5.1.2 Soluciones.....	47
5.2 Comisiones falleras.....	51
5.2.1 Necesidades de la asociación.....	51
5.2.2 Soluciones.....	52
6 Conclusiones.....	59
6.1 Estado del proyecto.....	59
6.2 Futuras mejoras.....	59
6.3 Valoración personal.....	60
7 Bibliografía.....	61
7.1 Recursos consultados.....	61

# **1 Introducción**

## **1.1 Contexto**

SuiteCRM es un sistema de gestión de relaciones con los clientes (CRM) OpenSource que evoluciona de otro proyecto también OpenSource llamado SugarCRM. Como todo CRM, proporciona funcionalidades diseñadas para permitir a las empresas obtener un mayor control sobre el estado de las relaciones comerciales con sus clientes.

Utilizando los servicios proporcionados por Amazon Web Services, Ticnux ha desarrollado TicnuxCloud, un servicio que provee a las aplicaciones web de una infraestructura de alta disponibilidad accesible desde cualquier punto con acceso a internet. A través de una asignación de recursos dinámica basada en el número de peticiones realizadas, TicnuxCloud consigue maximizar la eficiencia de los recursos activos.

Por otro lado, en Ticnux cuentan con una gran experiencia en la implantación, configuración y mantenimiento de SuiteCRM así como en el desarrollo de nuevos módulos que satisfagan las necesidades particulares de cada cliente.

A través del desarrollo de un Módulo de gestión de asociaciones para SuiteCRM, Ticnux espera atraer las distintas asociaciones ofreciéndoles un producto que se adapte perfectamente a su modelo de negocio mientras les permite sacar el máximo partido de un potente CRM, y pueda ser instalado en un entorno seguro de alta disponibilidad que permita al cliente acceder a la aplicación desde cualquier dispositivo con acceso a Internet.

## **1.2 Problemática**

Las asociaciones son, en muchos casos, organizaciones sin ánimo de lucro que cuentan con escasos recursos. Por este motivo, muchas de ellas todavía utilizan métodos de gestión rudimentarios o aplicaciones anticuadas que, aunque satisfacen sus necesidades básicas, no les ofrecen la totalidad de las funcionalidades y los beneficios que se puede conseguir utilizando tecnologías más punteras. A través del uso de tecnologías OpenSource se puede diseñar una solución económica que permita a estas asociaciones actualizar sus herramientas de trabajo a un coste inferior que otras alternativas privativas.

## **1.3 Objetivos**

En este proyecto se desarrollarán un conjunto de módulos para SuiteCRM con la finalidad de facilitar la gestión de todo tipo de



asociaciones, así como a otras entidades que tengan un perfil similar, como pueden ser asociaciones musicales, clubes deportivos, colegios profesionales, fallas, y en general cualquier organización que esté basada en las figuras de Asociados y Cuotas.

Se añadirán las funcionalidades necesarias para permitir a las asociaciones llevar un control del estado de sus asociados (Alta, baja, suspendido) y permita programar nuevos cambios.

Se diseñarán mecanismos para automatizar las tareas de facturación más repetitivas (cobro de cuotas, generación de facturas y generación de remesas).

El sistema debe ofrecer un alto grado de flexibilidad, ya que debe permitir que asociaciones con necesidades distintas puedan configurar la aplicación para adaptarla a estas necesidades.

Debido a que la aplicación final será una combinación de las funcionalidades de SuiteCRM más las añadidas durante este proyecto, se deberá conseguir que el desarrollo realizado se integre con las funcionalidades básicas y no interfiera con el funcionamiento habitual del CRM.

## 1.4 Metodología

Para elaborar este proyecto se ha dividido el desarrollo en tres etapas (análisis, implementación y redacción de la memoria) dando comienzo en marzo y finalizando en junio. Pese a identificar la redacción de la memoria como una fase independiente, durante todo el proceso se documentarán los avances más relevantes, así como toda la información que pueda resultar de utilidad posteriormente.

Durante la fase de análisis se determinarán las funcionalidades que deberá ofrecer la aplicación, las tecnologías utilizadas para su desarrollo y las técnicas que se usarán durante la implementación de la aplicación. Para ello se estudiarán distintos modelos de asociaciones y se analizarán aplicaciones ya existentes que ofrezcan soluciones similares. El objetivo principal es desarrollar una especificación clara y concisa del producto que marque las pautas a seguir durante la programación.

Para el desarrollo de esta fase se ha estimado una duración de un mes, repartiendo las tareas de la siguiente forma:

- Primera semana: análisis de los distintos modelos de asociación e identificación de las necesidades fundamentales.
- Segunda semana: estudio de soluciones ya existentes.
- Tercera y cuarta semana: redacción del documento de especificación.



La fase de implementación tiene una estimación aproximada de un mes y medio, durante el cual se seguirá la especificación elaborada para desarrollar los distintos apartados de la aplicación. Los desarrollos se irán agrupando por módulos y por funcionalidades, realizando la validación en una fase posterior a cada iteración. Seguiremos en las fases de implementación y validación hasta que la aplicación cumpla con los requisitos especificados.

Atendiendo a las funcionalidades identificadas y a la especificación detallada en el capítulo tres, la etapa de implementación queda dividida de la siguiente forma:

- Desarrollo y validación de las funcionalidades gestión asociados.
- Desarrollo y validación de las funcionalidades de gestión de cargos.
- Desarrollo y validación de las funcionalidades de facturación.
- Validación general.

Para el desarrollo de las funcionalidades relativas a la gestión de asociados y de cargos se ha estimado una semana y media para cada una. Tres semanas para las funcionalidades de facturación y una última semana para realizar las últimas validaciones.

Durante las siguientes cuatro semanas se realizará la redacción de la memoria, la cual recogerá los detalles más significativos de la experiencia obtenida durante todo el proceso.



## **2 Contexto tecnológico**

SuiteCRM es un proyecto CRM (*customer relationship management*) OpenSource basado en la última versión OpenSource de SugarCRM (6.5). Un sistema CRM es una aplicación usada por las organizaciones para mantener y controlar el estado de las relaciones comerciales con sus clientes, clientes potenciales, y otros posibles contactos. La mayoría de sistemas CRM tienen aplicaciones comunes como herramientas para realizar y monitorizar campañas de marketing a través del envío de e-mails, la generación de informes personalizados, gestión de ventas o la programación de los distintos tipos de actividades (reuniones, llamadas, visitas, etc.) realizadas con los clientes.

Aunque en la actualidad existen diversas aplicaciones CRM OpenSource similares a SuiteCRM como Oro CRM, EspoCRM, CiviCRM o Zurmo, SuiteCRM está considerado como uno de los mejores. El uso de SuiteCRM en este proyecto ha sido marcado por el contexto laboral, debido a que Ticnux cuenta con años de experiencia en el desarrollo de módulos para SugarCRM y SuiteCRM.

Pese a que el uso final de la plataforma dependerá de cada asociación, puede llegar a ser muy interesante para estas el uso de las funcionalidades de realización campañas de marketing y envío de e-mails, como herramienta para la captación de nuevos socios y programas de fidelización para los actuales.

### **2.1 Aproximaciones similares**

Al buscar información sobre aplicaciones con unas características similares encontramos dos modelos distintos que resultan de interés para el desarrollo de este proyecto. En el primer grupo podemos ver aplicaciones con funcionalidades exclusivamente para la gestión administrativa de la asociación, pensadas para la instalación en un único ordenador o para uso interno en una red local. El segundo grupo describe aplicaciones web pensadas para acceder desde cualquier dispositivo con acceso a internet que añaden funcionalidades para controlar y mejorar las interacciones entre los socios y la asociación.

Por un lado, si nos centramos únicamente en las labores administrativas de las asociaciones encontramos aplicaciones como Socius de Baarasoft o Asociaciones XL de IdeSoft. Ambos proporcionan funcionalidades para la gestión de socios en el sistema y la generación de informes configurables sobre la vida de los mismos, el envío de e-mails y la posibilidad de automatizar la generación de remesas de recibos.

Socius integra en una misma aplicación modelos para representar la información de distintos tipos de asociación, así realiza una distinción entre grupos como Asociaciones musicales, Asociaciones de padres y madres, ONG, Asociaciones deportivas, Asociaciones religiosas, etc. Añadiendo para cada uno de los modelos distintos campos y funcionalidades. Por ejemplo en el módulo de Agrupaciones musicales podemos encontrar que a la ficha de socio le podemos asignar el instrumento que toca con la posibilidad de indicar si dicho instrumento ha sido prestado por la asociación. Por otra parte, su representación de una asociación de padres y madres incluye un módulo de actividades que representa las distintas actividades a las que se inscribe el niño.

Mientras que Asociaciones XL ofrece funcionalidades estándar sin detallar el tipo de asociación, IdeSoft también ofrece el programa Club deportivo XL que incluye módulos para representar los grupos familiares y la gestión de monitores, entrenadores, personal y trabajadores.

Por otro lado, si añadimos las funcionalidades brindadas por SuiteCRM (campañas de marketing, generación de informes, control de ventas, gestión de reuniones, llamadas, etc. ), y buscamos plataformas que sigan el modelo de negocio *Software as a Service* (SaaS), encontramos varias aplicaciones, englobadas bajo la categoría de *Association Management Software* o *Membership Software*, diseñadas tanto para cumplir las labores de gestión administrativa de la asociación, como para proporcionarles herramientas para facilitar la captación y la fidelización de sus socios. Adicionalmente este tipo de aplicaciones están diseñadas para funcionar en la nube, proporcionando a los usuarios la posibilidad de trabajar con la herramienta desde cualquier dispositivo con acceso a Internet.

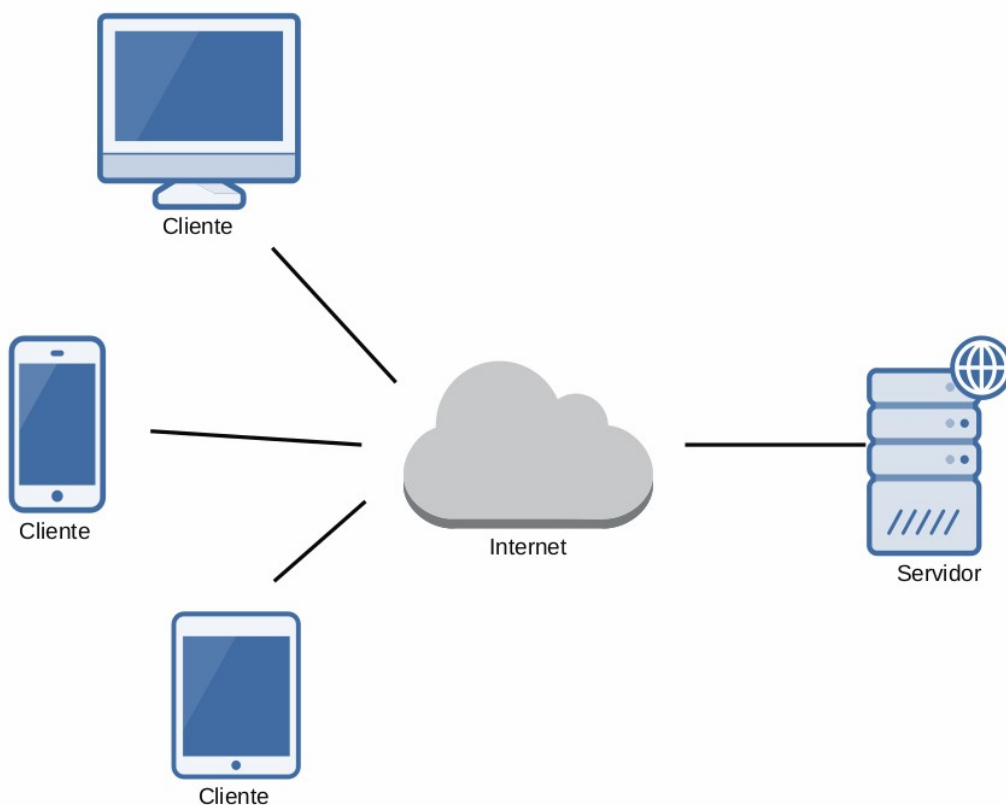
Dentro de este grupo de aplicaciones podemos destacar algunas como WebLink Connect, Association Catalyst o Raklet. Este tipo de plataformas incluyen funcionalidades de gestión de asociados, como pueden ser la impresión de informes, gestión de facturación (incluyendo en algunas como Raklet el cobro directo al asociado y el acceso de este a un panel desde el que controlar todos sus pagos realizados a la asociación), el control del estado del socio en el sistema o el acceso a la vida del mismo dentro de la asociación, y otras funcionalidades adicionales como la venta de entradas para eventos o cursos de formación, el control de asistencia a los mismos o la generación de campañas de e-mail y su monitorización a través del uso de la información obtenida al abrir el e-mail o visitar un enlace del texto.

## 2.2 Tecnologías relevantes

Debido a que SuiteCRM es una aplicación web que sigue el modelo cliente servidor, en este proyecto utilizaremos lenguajes distintos según tengamos que modificar secciones correspondientes a la representación de



las vistas en el lado del cliente o modificaciones sobre la lógica de negocio realizadas en el lado del servidor.



*Imagen 1: Arquitectura Cliente - Servidor*

En la arquitectura cliente-servidor distintos clientes pueden estar simultáneamente conectados a la aplicación realizando peticiones que el servidor se encarga de procesar y responder enviando la información correspondiente a cada uno.

Los lenguajes de programación y tecnologías que utilizaremos en este proyecto nos vienen determinados en su mayor parte por las características del framework SuiteCRM.

### **2.2.1 PHP**

PHP es un lenguaje de scripting de código abierto que puede ser ejecutado en la mayoría de los servidores web y en casi todos los sistemas operativos. Se ha vuelto muy popular debido a su uso en el desarrollo de aplicaciones web, utilizado para la programación en el lado del servidor, PHP permite una generación dinámica de los documentos HTML enviados al cliente.

Entre las ventajas de utilizar PHP cabe destacar que su uso es totalmente libre, es un lenguaje de programación con un tiempo de

aprendizaje corto, permite el estilo de programación orientado a objetos y cuenta con una comunidad de programadores muy extensa.

Dado que PHP es un lenguaje interpretado, puede presentar tiempos de respuesta considerablemente más elevados que su equivalente si usásemos un lenguaje de programación de bajo nivel. Este punto puede ser atenuado mediante el uso de memorias cache que almacenen las respuestas a las peticiones realizadas con mas frecuencia.

Otro inconveniente es que el único método para ocultar el código generado es mediante la ofuscación, una técnica que aunque dificulta la comprensión del código, no imposibilita su lectura.

### **2.2.2 Smarty**

Smarty es un motor de plantillas para PHP utilizado en la programación web. Facilita herramientas para separar la lógica de la aplicación de la capa de presentación a través de un sistema cómodo y eficiente. Esta separación permite que el código sea mucho mas depurable y permite que futuras modificaciones en cualquiera de las dos capas (lógica o presentación) no altere el funcionamiento de la otra.

### **2.2.3 JQuery**

JQuery es un framework de javascript que proporciona al desarrollador una abstracción muy potente. Javascript es un lenguaje popularizado por su uso en el lado del cliente, ya que puede ser interpretado por la mayoría de los navegadores.

Su uso en las aplicaciones web es muy variado, en este proyecto lo utilizaremos para realizar modificaciones dinámicas sobre las distintas vistas y la realización de validaciones en los formularios antes de enviar los datos al servidor. También haremos uso de la función Ajax de JQuery, la cual nos permite realizar peticiones al servidor de una forma asíncrona y sin necesidad de recargar la página visualizada por el usuario.

SuiteCRM incluye en todas sus vistas las librerías correspondientes a la última versión de JQuery, por lo que no será necesario cargar ninguna librería adicional cuando vayamos a hacer uso de alguna de sus funciones.

### **2.2.4 MySQL**

Es un sistema de gestión de bases de datos relacional. Se utiliza comúnmente en aplicaciones web junto a PHP y Apache por la buena integración de estas tecnologías OpenSource y su bajo coste de adquisición.



## 2.2.5 Apache

Apache es un servidor web de código abierto compatible con la mayoría de sistemas operativos actuales. En este proyecto lo utilizaremos para instalar las instancias de pruebas y poder testear la aplicación.

La siguiente comparativa muestra la cuota de mercado de los distintos servidores activos a lo largo de los últimos años. En ella podemos apreciar como apache ha sido históricamente y continua siendo el servidor web más utilizado.

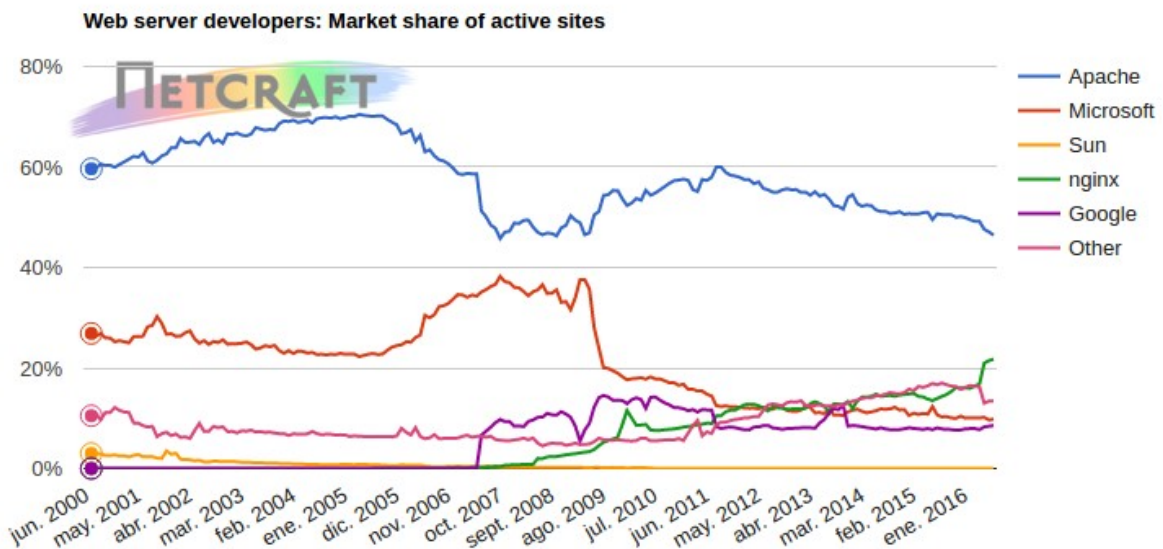


Imagen 2: Comparativa servidores web - <https://news.netcraft.com/>

## **3 Análisis y Diseño**

### **3.1 Funciones del producto**

Una asociación se suele ver como una agrupación de personas que se unen para desarrollar alguna actividad común o lograr algún propósito colectivo.

Todas las asociaciones necesitan mantener, ya sea en papel o en formato digital, información sobre la ficha de cada asociado, las comunicaciones realizadas con ellos y otra documentación propia de la entidad. Al realizar el almacenamiento de estos datos en un formato digital y utilizar un sistema de búsqueda eficiente, las asociaciones pueden reducir los tiempo en desarrollar las labores administrativas.

Un buen uso de la información como “desde cuando está en la asociación”, “a qué eventos ha asistido”, “tiene familia dentro de la asociación” o “qué rol desempeña dentro de la asociación”, les permitirá proporcionar un trato personalizado a cada uno de los socios y anticipar posibles problemas a través del estudio de eventos pasados.

En este proyecto nos centramos en la figura del asociado como centro de la asociación, el socio se da de alta en el sistema al inscribirse en la asociación por primera vez. Una vez dentro deberá pagar una cuota de permanencia, estará sujeto a una normativa interna y tendrá acceso al uso de los servicios que ofrezca.

De la representación del asociado se podrá extraer la información necesaria en cada momento a través de la realización de búsquedas sobre diversos módulos, asegurándonos de mantener dicha información accesible de una forma cómoda para los usuarios.

Además de mantener la información y proporcionar acceso a ella, la aplicación ofrecerá otras funcionalidades para automatizar las tareas realizadas relativas al asociado. Para ello se desarrollarán herramientas que permitan planificar los cambios en el estado (Alta o Baja) y suspensiones (sanciones impuestas por incumplimiento de los estatutos).

Aunque en la mayoría de situaciones se entiende asociación como una organización sin ánimo de lucro, todas las entidades necesitan disponer de un capital para lograr desarrollar su actividad. La manera más habitual de lograrlo para las asociaciones es a través de donaciones y mediante el cobro de cuotas. En este proyecto nos centraremos en la necesidad de las distintas organizaciones de cobrar una tasa a sus socios, asumiendo que cada una de ellas los realiza utilizando una serie de parámetros distintos (tiempo de pago, descuentos por asociado, recurrencia de los pagos, etc.)



distintos e intentando prever cada una de estas necesidades para diseñar su solución.

Las funcionalidades relativas al proceso de facturación y cobro de cuotas incluirán la planificación y automatización en la generación de los pagos y la generación del documento XML SEPA DIRECT DEBIT para el cobro a través de domiciliación bancaria de las cuotas recurrentes dentro de la zona SEPA (formado por los miembros del Espacio Económico Europeo y SanMarino, Suiza y Mónaco).

Por último, todas las entidades necesitan realizar ciertas tareas de administración (toma de decisiones, tesorería, organización actividades, etc.), lo que implica que personas internas o externas a la asociación deben desarrollar estos cargos. Usualmente, en las organizaciones sin ánimo de lucro estos cargos suelen dividirse en dos órganos, el órgano de gobierno (generalmente denominado “asamblea de socios”) y órgano de representación (normalmente “junta directiva”).

Esta aplicación permitirá almacenar la información sobre los cargos que desarrolla cada asociado, permitiendo realizar la configuración en base a los estatutos de cada asociación, y realizar búsquedas sobre el estado de estos cargos en cualquier momento pasado de la vida de la asociación.

Para poder comprender mejor las funcionalidades que ofrecerá esta aplicación se han dividido en tres grupos:

### **3.1.1 Gestión de asociados**

**Altas y bajas:** se creará un sistema que permita mantener en todo momento el estado (alta o baja) del asociado. Se deberá mantener la información de los cambios realizados sobre los que se podrá realizar búsquedas y generar informes. El usuario podrá planificar un alta (o baja) y el estado del asociado cambiará automáticamente al llegar el día indicado.

**Suspensiones:** se desarrollará un módulo que permita a la dirección de la asociación sancionar a los asociados durante un periodo de tiempo, por los motivos que indiquen en sus estatutos. Las suspensiones se deberán poder planificar y el asociado deberá volver a su estado anterior automáticamente cuando finalice el periodo de suspensión.

Todos los cambios relativos a las altas (o bajas) y a las suspensiones se verán reflejados en un histórico de cambios sobre el que se podrán realizar consultas y podrá ser utilizado para preparar informes.

Para dar soporte a estas funcionalidades se crearán los módulos altas y bajas, suspensiones e histórico de cambios. El módulo “Contacts” perteneciente al núcleo de SuiteCRM será modificado para corresponder con la figura de asociado.



### **3.1.2 Gestión de cargos**

Se desarrollarán los módulos necesarios para permitir la representación en el sistema de las distintas funciones que desarrollan los asociados dentro de la organización así como la posibilidad de agrupar los distintos cargos en diferentes categorías (secretaría, tesorería, dirección... ).

La información se deberá mantener indefinidamente en el sistema con la finalidad de poder realizar futuras búsquedas y la posibilidad de generar informes.

La gestión de cargos se representará mediante los siguientes módulos:

- Órganos
- Cargos
- Organización

### **3.1.3 Facturación**

La aplicación permitirá la representación virtual de las facturas de la asociación, las cuales se podrán agrupar bajo un documento remesa que podrá ser tramitado en cualquier banco de la zona SEPA.

Se deberá poder programar la generación automática de facturas acorde a las necesidades de cada asociación. Esto se conseguirá a través del desarrollo de un módulo de cuotas con los campos de configuración necesarios.

Para dar soporte a las funciones de facturación se desarrollarán los siguientes módulos:

- Cuotas
- Pagos
- Facturas
- Remesas



## 3.2 Características del usuarios

Basándose en los privilegios que tienen los usuarios, SuiteCRM los divide en dos grupos:

### 3.2.1 Usuario Normal

Representa al usuario final de la aplicación. En nuestro contexto este será el rol que deberán tener los encargados de las tareas administrativas de la asociación.

### 3.2.2 Usuario Administrador del Sistema

Representa al usuario con máximos privilegios sobre la aplicación. Aunque es recomendable que al menos un usuario miembro de la asociación desarrolle el rol de administrador, es necesario tener unos conocimientos previos sobre SuiteCRM antes de poder configurar correctamente una instancia, por lo que siempre se recomendará contar con la asistencia de un experto para realizar la configuración inicial y, a poder ser, las tareas de mantenimiento.

## 3.3 LOPD

El uso de esta aplicación implicará que se están almacenando información de carácter personal correspondiente a los distintos miembros de la asociación. Por ello es necesario que la organización tome las medidas necesarias para cumplir con la normativa descrita en la LOPD.

Deberán, entre otras, cosas solicitar un consentimiento explícito por parte de los socios a la hora de obtener la información y garantizarles los derechos ARCO (acceso, rectificación, cancelación y oposición).

Aunque las funciones de facturación de esta aplicación implique la necesidad de almacenar información sobre operaciones financieras de los asociados, esta información es utilizada únicamente para realizar el cobro de las cuotas. Mientras no se almacene en el sistema información adicional, no prevista en este proyecto, todos los datos almacenados se englobarán en la categoría de “nivel de seguridad básico” y deberán atender a la normativa descrita en el Real Decreto 1720/2007, de 21 de diciembre.

## 3.4 Seguridad

SuiteCRM es una aplicación web y como tal está sujeta a las diversas amenazas que se pueden encontrar en internet. Por este motivo, es necesario destacar algunos puntos relativos a la seguridad de la aplicación.

El equipo de desarrolladores de SuiteCRM trabaja constantemente en solucionar las vulnerabilidades detectadas por ellos o por los miembros de la comunidad, a pesar de esto, es importante tomar medidas adicionales

para asegurar al máximo posible que nuestro sistema está seguro. Mantener actualizado el sistema, la versión de PHP y el servidor web puede evitar que la información manejada por la aplicación caiga en manos no deseadas.

Por otro lado, para interactuar con la aplicación es necesario tener una cuenta de usuario dentro del sistema e identificarse dentro de la misma. Una vez dentro, los usuarios podrán trabajar con la información almacenada en los distintos módulos, dependiendo de sus privilegios dentro del sistema. Aunque este requisito añade aparentemente una capa extra de seguridad, esta es a su vez el punto crítico de la aplicación. El uso de ingeniería social o un simple descuido por parte del usuario, puede ocasionar que un atacante logre acceder al sistema sin que quede constancia de ello. Por ello es muy importante que los responsables de la organización conciencien a todos los usuarios y les insten a seguir buenas prácticas a la hora de interactuar con la aplicación.

Las medidas que se pueden tomar pueden llegar a ser muy complejas, pero seguir una serie de acciones básicas como utilizar contraseñas complejas, no apuntar las contraseñas en ningún medio o obligar a actualizar las contraseñas periódicamente, pueden frustrar la mayoría de ataques.

### **3.5 Especificación por módulos**

En SuiteCRM existe una herramienta llamada Estudio que permite a los usuarios administradores de la aplicación la personalización de todos los módulos. Permite entre otras cosas la creación de nuevos campos y la modificación de las diferentes vistas (Edición, Detalle, Lista... ). Esta es una característica a la cual se le quiere sacar partido en este proyecto ya que posibilitará que cada asociación pueda personalizar los distintos módulos para que la adaptación al uso de esta aplicación sea lo más exacta posible.

Todos los módulos tienen por defecto los campos nombre (que será el campo mostrado en los módulos relacionados), fecha de creación, fecha de última modificación y usuario asignado.

Fecha de creación y fecha de última modificación se calculan automáticamente gracias al framework.

Usuario asignado se genera automáticamente con la identificación del usuario que está creando el registro.



A continuación veremos el esquema de relaciones entre los distintos módulos:

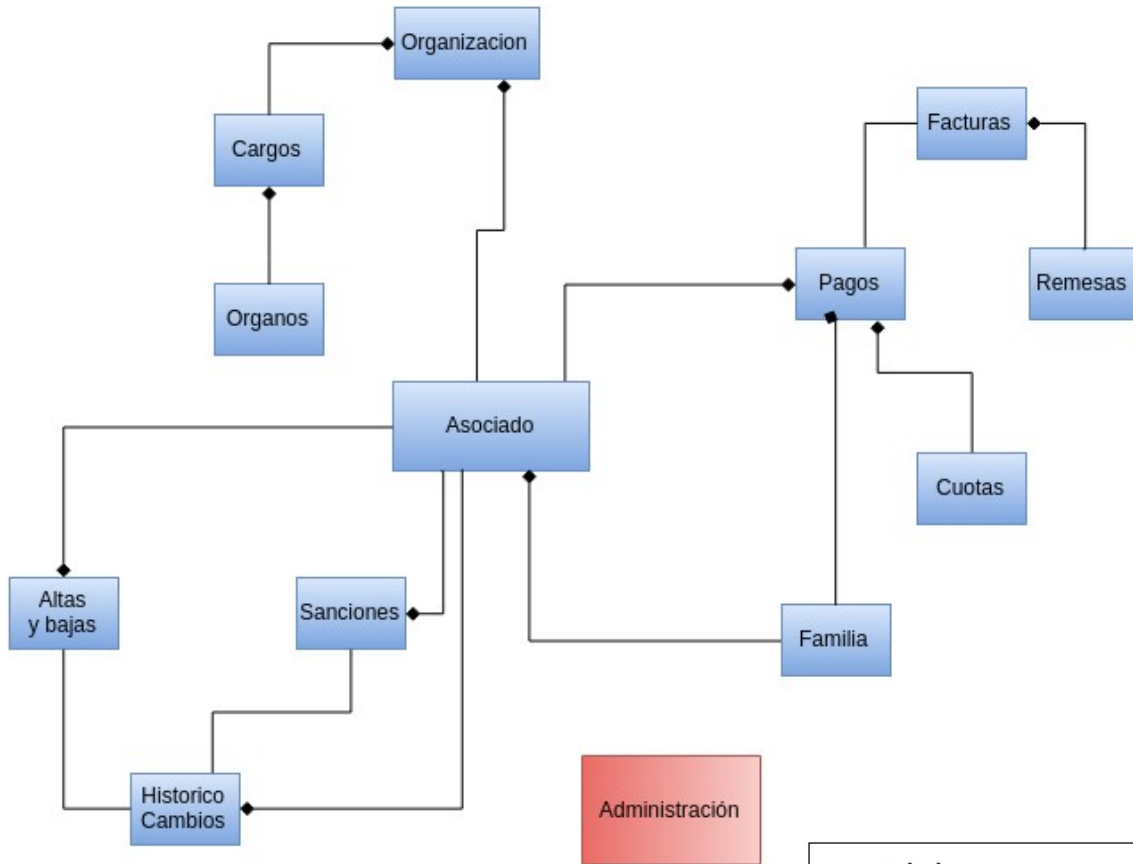
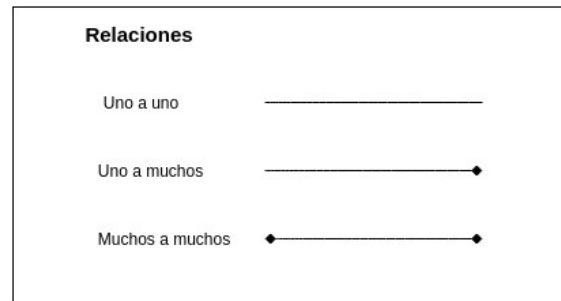


Imagen 3: Esquema de relaciones entre módulos



### 3.5.1 Asociados

#### Descripción

El módulo de asociados representa a la figura principal de la asociación, los socios. Debido a la importancia de esta figura para la aplicación, desde la vista de detalle de un asociado se pueden realizar la mayor parte de las acciones relacionadas con la gestión de asociados.

#### Campos

Nombre	Tipo	Observaciones
bic	varchar	Autogenerado basado en el campo

<b>Nombre</b>	<b>Tipo</b>	<b>Observaciones</b>
		codigo_entidad
codigo_cuenta	varchar	Si alguno de los campos tiene valor en el momento de guardar se comprobará la validez de los mismos. No se podrá guardar el asociado a menos que todos los campos sean correctos o estén vacíos.
codigo_dc	varchar	
codigo_entidad	varchar	
codigo_iban	varchar	
codigo_sucursal	varchar	
edad	int	Autogenerado
estado	enum	Autogenerado (Alta o Baja)
familia	bool	Autogenerado (Indica la pertenencia del asociado a una familia)
forma_pago	enum	Indica en cuantos pagos desea el asociado pagar las cuotas anuales
generar_facturas	bool	Si está activo el sistema convertirá los pagos generados automáticamente en facturas
generar_pagos	bool	Si está activo el sistema generará automáticamente pagos para este asociado
iban	varchar	Autogenerado basado en los campos codigo_cuenta, dc, entidad, iban y sucursal
nif	varchar	Si el campo no es vacío se comprobará su validez en el momento



Nombre	Tipo	Observaciones
		de guardar el asociado
perfil	enum	La lista de perfiles se podrá editar desde el panel de administración
remesar	bool	Si está activo se podrán incluir las facturas de este asociado en las remesas.
sancion	bool	Autogenerado.
sexo	enum	

### **Procesos**

#### Al guardar

- Calcular edad
- Generar IBAN y BIC
- Crear alta: Si el registro es nuevo crear un nuevo registro de alta

### **Acciones**

- Dar de alta / Dar de baja: Sólo se mostrará una de las dos acciones dependiendo del estado del asociado. Cambia el estado del asociado (y crea un nuevo registro Alta o Baja) inmediatamente.
- Planificar alta / Planificar baja: Sólo se mostrará una de las dos acciones dependiendo del estado del asociado. Redirige a la vista de edición de Altas y Bajas para que el usuario indique la fecha en la que se producirá el cambio. Una tarea programada se encarga de llevar a cabo el cambio cuando llegue la fecha indicada.
- Sancionar: Redirige a la vista de edición de Sanciones para que el usuario pueda indicar el periodo durante el que el asociado estará sancionado. Una tarea programada se encarga de cambiar el estado del asociado cuando comienza y cuando termina la sanción.
- Generar pago: Redirige a la vista de edición de Pagos para que el usuario pueda completar los datos restantes.

### **Comentarios**

El módulo Asociados será una personalización del módulo Contacts del core de SuiteCRM, campos como Nombre, Apellidos, Foto... ya están incluidos por defecto así como relaciones con el resto de módulos del core.

Los campos `codigo_sucursal`, `codigo_iban`, `codigo_entidad`, `codigo_dc`, `codigo_cuenta` y `nif` son necesarios para poder generar los documentos XML correspondientes a las remesas.

### 3.5.2 Familias

#### **Descripción**

Los asociados pueden pertenecer a una familia con la finalidad de agrupar los pagos de todos los miembros en uno sólo.

#### **Campos**

Nombre	Tipo	Observaciones
<code>generar_pagos</code>	bool	
<code>generar_facturas</code>	bool	
<code>remesar</code>	bool	
<code>iban</code>	varchar	
<code>nif</code>	varchar	
<code>bic</code>	varchar	
<code>codigo_cuenta</code>	varchar	Los datos bancarios se generarán basándose en los datos del Responsable de la familia
<code>codigo_dc</code>	varchar	
<code>codigo_entidad</code>	varchar	
<code>codigo_iban</code>	varchar	
<code>codigo_sucursal</code>	varchar	
<code>forma_pago</code>	enum	
<code>responsable</code>	related(asociados)	Miembro de la familia cuyos datos se utilizarán para el cobro de las facturas.

#### **Procesos**

##### Al guardar

- Rellenar campos facturación
- Generar BIC e IBAN



#### Al añadir relación con asociado

- Cambiar estado asociado: Activar checkbox familia

#### Al eliminar una relación con asociado

- Cambiar estado asociado: Desactivar checkbox familia

#### **Comentarios**

Si una familia tiene activo el checkbox de generar pagos, se generarán pagos para la familia independientemente de las opciones individuales de cada asociado.

### **3.5.3 Altas y bajas**

#### **Descripción**

Cada vez que se desee cambiar el estado del asociado (Alta o Baja) se deberá hacer creando una nueva entrada en este módulo. Aunque las acciones “Dar de alta” y “Dar de baja” se pueden realizar cómodamente desde el panel de asociado la única manera de poder planificar un alta es a través de este módulo.

#### **Campos**

<b>Nombre</b>	<b>Tipo</b>	<b>Observaciones</b>
tipo	enum	Alta o Baja
fecha_efectiva	date	Fecha en la que se produce el cambio

#### **Procesos**

##### Al añadir relación con asociado

- Generar nombre (basado en nombre de asociado).
- Cambiar estado de asociado (alta o baja).
- Crear entrada en el histórico de cambios.
- Validar alta/baja: No se permite la creación de altas o bajas si el asociado se encuentra ya en ese estado.

##### Antes de borrar

- Bloquear borrado: Sólo se permite el borrado del último registro de alta/baja de cada asociado. Una vez borrado se revierte el estado del asociado y se crea una entrada en el histórico de cambios.

#### **Comentarios**

Una tarea programada se encargará de cambiar el estado de los asociados cuando llegue la fecha efectiva de un Alta o Baja



### 3.5.4 Sanciones

Representa las posibles penalizaciones que se les pueda aplicar a los asociados por motivos de incumplimiento del reglamento interno.

#### **Campos**

<b>Nombre</b>	<b>Tipo</b>	<b>Observaciones</b>
concepto	varchar	
fecha_inicio	date	
fecha_fin	date	

#### **Procesos**

##### Al añadir relación con asociado

- Comprobar que la sanción es válida: Una sanción es válida si el asociado relacionado no está sancionado ni tiene ninguna sanción planificada.
- Si la sanción es válida cambiar el campo “sanción” del asociado (si estamos dentro de su período de vigencia) y generar una entrada en el histórico de cambios.

##### Al eliminar una relación con asociado

- Cambiar el campo “sanción” del asociado se estamos dentro del periodo de vigencia de la sanción y generar una entrada en el histórico de cambios.

### 3.5.5 Histórico de cambios

#### **Descripción**

Se trata de un módulo meramente informativo desde el cual se podrá controlar la historia de todos los asociados con un único golpe de vista.

#### **Campos**

<b>Nombre</b>	<b>Tipo</b>	<b>Observaciones</b>
tipo	enum	Alta, Baja, Sanción, Cambio Perfil, Nuevo Cargo
antiguo	varchar	
nuevo	varchar	



Nombre	Tipo	Observaciones
fecha_cambio	date	

### **Comentarios**

El módulo no tiene ninguna lógica propia.

Se generarán entradas cada vez que se cree una nueva sanción o una alta/baja.

Se generará una entrada cada vez que se produzca un cambio en el perfil de un asociado

## **3.5.6 Órganos**

### **Descripción**

Representa un subgrupo dentro de la asociación encargados de desarrollar unas tareas determinadas

### **Comentarios**

Este módulo no tiene campos adicionales, únicamente una relación con Cargos.

## **3.5.7 Cargos**

### **Descripción**

Representa un cargo específico dentro de un órgano.

### **Comentarios**

Este módulo no tiene campos adicionales, únicamente la relación con Órganos y Organización

Sólo podrá haber un asociado desempeñando un cargo al mismo tiempo.

Un asociado podrá desempeñar varios cargos distintos al mismo tiempo.

## **3.5.8 Organización**

### **Descripción**

Módulo que relaciona a un asociado con un cargo durante un período de tiempo

### **Campos**

Nombre	Tipo	Observaciones
fecha_inicio	date	
fecha_fin	date	

### **Procesos**

Al añadir relación con asociado

- Comprueba que el cargo seleccionado no está siendo desempeñado por otro asociado durante el mismo período de tiempo.

### 3.5.9 Pagos

#### **Descripción**

Representa una obligación de pago por parte del asociado o de la asociación.

Se añade como paso previo a las facturas con la finalidad de que el usuario pueda validar los pagos generados automáticamente si así lo desea.

#### **Campos**

<b>Nombre</b>	<b>Tipo</b>	<b>Observaciones</b>
estado	enum	Emitido, Pagado, Anulado, Impagado o Convertido a factura
tipo	enum	Pago (Asociado paga a asociación)  Cobro (Asociado cobra de Asociación)
importe	decimal	
fecha_ejecucion	datetime	
concepto	varchar	

#### **Procesos**

##### Al añadir relación

- Comprobar que el pago está relacionado con una Familia o con un Asociado pero nunca con los dos a la vez.
- Si se relaciona una factura con el pago se cambiará su estado a "Convertido a factura"

##### Al guardar

- Si el pago ha sido convertido a factura no se podrá volver a modificar ninguno de sus campos.

#### **Acciones**

- Convertir a factura.



### 3.5.10 Facturas

#### Campos

Nombre	Tipo	Observaciones
remesar	bool	Si está activo la factura se podrá incluir en una remesa
fecha_ejecucion	datetime	
estado	enum	Emitida, Pagada, Anulada, Impagada
serie	enum	Dos series definidas "P" (facturas tipo pago) y "C" (facturas tipo cobro)
importe	decimal	
tipo	enum	Pago (Asociado paga a asociación) Cobro (Asociado cobra de Asociación)
concepto	varchar	
codigo	varchar	Autogenerado. Concatenación de Serie + (número de facturas + 1) + / + año

#### Procesos

##### Al guardar

- Validar y generar código: No pueden haber facturas generadas con fecha de ejecución posterior a la que se está creando ahora.
- Si la factura es correcta generar el código.

### 3.5.11 Cuotas

#### Descripción

Módulo principal para la gestión de facturación. Se deberán crear las cuotas necesarias dependiendo de los requisitos de la asociación para que el sistema genere correctamente los pagos.

### Campos

Nombre	Tipo	Observaciones
activa	bool	Sólo las cuotas activas generarán pagos automáticamente.
todos_asociados	bool	Si está activo la cuota generará pagos para todos los asociados independientemente de su perfil
todos_edades	bool	Si está activo la cuota generará pagos para todos los asociados independientemente de su edad
genera_pagos	bool	Sólo las cuotas con generar_pagos activo generarán pagos automáticamente.
tipo	enum	Anual, Única y Periódica
periodicidad	enum	Sólo para cuotas periódicas, indica la frecuencia de generación de pagos
tipo_pago	enum	Pago (Asociado paga a asociación)  Cobro (Asociado cobra de Asociación)
fecha_inicio	date	
fecha_fin	date	Una tarea programada cambiará las cuotas a inactivas cuando llegue su fecha de fin



<b>Nombre</b>	<b>Tipo</b>	<b>Observaciones</b>
nuevos_asociados	bool	Si NO está activo el sistema sólo generará pagos para los asociados que existían antes de la fecha de inicio indicada
perfil	enum	Permite generar pagos para un subconjunto de asociados basándose en su perfil
descuentos_perfil	bool	Aplica los descuentos por perfil indicados en el área de Administración
descuentos_edad	bool	Aplica los descuentos por edad indicados en el área de Administración
importe	decimal	
edad_from	int	Permite generar pagos para un subconjunto de asociados basándose en su edad
edad_to	int	
cobro_total	bool	Sólo para cuotas anuales. En el último periodo de la cuota genera un pago compensatorio hasta llegar al total del importe de la cuota.

### **Procesos**

#### Al guardar

- Comprobar Pagos: si la cuota ya ha generado pagos sólo se permitirá cambiar los campos "Activa" , "Fecha fin" y "Generar pagos".



- Si la cuota está activa y genera pagos, generará pagos para los asociados correspondientes.

### **Acciones**

- Generar pago manual: Redirige a la vista de edición de pagos con la información de la cuota ya cargada.
- Generar pagos: Generará los pagos pertinentes para los asociados.

### **Tipos de Cuota**

#### Anual

El importe indicado será el importe total que se dividirá en X pagos dependiendo del campo “forma\_pago” de cada uno de los asociados.

#### Periódica

El importe indicado será el utilizado para cada uno de los pagos. Se ejecutará cada X tiempo, dependiendo del campo “periodicidad” de esta cuota.

#### Única

El importe indicado será el utilizado para cada uno de los pagos. Cada asociado sólo podrá tener un único pago relacionado con esta cuota.

## **3.5.12 Remesas**

### **Descripción**

Conjunto de facturas agrupadas bajo un mismo concepto con la finalidad de poder generar los documentos XML “SEPA DIRECT DEBIT” o “SEPA CREDIT TRANSFER”

### **Campos**

<b>Nombre</b>	<b>Tipo</b>	<b>Observaciones</b>
tipo	enum	Pago (Asociado paga a asociación)  Cobro (Asociado cobra de Asociación)
importe_previsto	decimal	Autogenerado. Suma del importe de todas las facturas relacionadas.
estado	enum	Pendiente, Lista, Pasada
fecha_ejecucion	datetime	



## **Procesos**

### Al añadir o borrar relación

- Calcular importe previsto: Recalcular el importe previsto de la remesa

## **Acciones**

- Buscar facturas por cuota: Sólo visible si la remesa está relacionada con una cuota. Busca y relaciona las facturas relacionadas con la cuota, estado emitido, que estén marcadas para remesar y que no estén relacionadas con otra remesa.
- Buscar facturas por tipo: Sólo visible si la remesa no está relacionada con una cuota. Busca y relaciona todas las facturas del mismo tipo que la remesa, estado emitido, que estén marcadas para remesar y que no estén relacionadas con otra remesa.
- Generar documento XML:

Si la remesa es de “tipo pagos” generará un documento SEPA DIRECT DEBIT acorde a la especificación *pain.008.001.02.xsd*.

Si la remesa es de “tipo cobros” generará un documento SEPA CREDIT TRANSFER acorde a la especificación *pain.001.001.03.xsd*.

En el documento generado se incluirán todas las facturas relacionadas con la cuota y que estén en estado Emitidas. Si alguno de los asociados o de las facturas produce un error, el documento generado tendrá como prefijo la cadena de texto “ERROR\_”, el cual deberá quedar registrado.

La aplicación deberá permitir la realización de pruebas antes de generar el documento definitivo.

- Validar remesa: Únicamente estará visible si la remesa ya ha generado al menos un documento xml. Esta acción cambiará el estado de todas las facturas relacionadas a “Pagada”.

## **Comentarios**

Para poder generar cualquiera de los documentos es necesario que los asociados relacionados tengan rellenos los campos “NIF” e “IBAN”.

El campo “fecha de mandato” (MndtId) del xml representa la fecha en la que el asociado autorizó la transacción, debe de ser al menos 3 días antes que la fecha de ejecución de la remesa y se obtiene de la fecha del último alta del asociado.



Para que la asociación pueda utilizar estos documentos es necesario que establezcan comunicación previamente con su banco para iniciar el proceso y que les proporcionen su “Identificación de presentador” que deberán introducir en el área de administración.

### 3.5.13 Administración

Sólo podrán acceder a esta área los usuarios que tengan permisos de administración.

#### **Campos**

<b>Nombre</b>	<b>Observaciones</b>
Lista de perfiles	Permite crear la lista de perfiles de la asociación junto al descuento aplicado para cada perfil
Lista de edades	Permite crear una lista de rangos de edades y seleccionar el descuento que se aplicará (si lo hubiese) a cada una
Utilizar log personalizado	Sólo para desarrolladores. Permite la creación de un log por el cual se loguearán únicamente las acciones realizadas por los métodos propios. Esto se realizará de forma adicional al log habitual (suitecrm.log) .
Nombre del fichero de log	
Identificación del presentador	Información necesaria para la generación de documentos “SEPA DIRECT DEBIT” y “SEPA CREDIT TRANSFER”.  La identificación del presentador se deberá solicitar por parte de la asociación en su banco habitual.
Nombre del presentador	
IBAN del presentador	
BIC del presentador	
DNI / NIF del Acreedor	



### 3.5.14 Tareas programadas

#### **Descripción**

Definimos tareas programadas como las funciones que deberán ejecutarse en segundo plano para realizar diversas tareas.

Todas las tareas programadas deberán ser configuradas para ejecutarse una vez al día.

#### **Tareas**

- Actualizar Edad: Actualiza el campo edad de los asociados.
- Buscar Altas: Busca altas y bajas cuya fecha de ejecución sea hoy para cambiar el estado del asociado.
- Buscar Sanciones: Busca sanciones que empiecen o terminen hoy para cambiar el estado del asociado.
- Generar Pagos: Genera pagos de las cuotas que deban generar pagos hoy.
- Buscar Cuotas: Busca cuotas que comiencen o terminen hoy para cambiar su estado.

# 4 Implementación

## 4.1 Estrategias de implementación

Para conseguir desarrollar todo lo descrito en la especificación anterior se ha seguido un proceso evolutivo comenzando por los casos mas simples e incrementando la complejidad poco a poco.

El primer paso ha sido la creación de la estructura de los distintos módulos, creación de campos, vistas básicas y relaciones entre módulos. Cada módulo tendrá una representación, en forma de tabla, en la base de datos utilizada. La base de datos es dependiente del entorno sobre el que se despliegue la aplicación, en nuestro caso hemos utilizado una base de datos MySQL para el entorno de pruebas. Cuando se cree un nuevo registro el framework asignará automáticamente un identificador único (clave primaria) para el mismo.

Las relaciones entre los módulos quedarán representadas mediante una tabla intermedia que incluirá una clave ajena apuntando al identificador de los dos módulos relacionados. Como medida de seguridad, SuiteCRM incluye en todas las tablas el campo "deleted", que indicará si el registro (o la relación) está borrado o no. De esta forma aunque un usuario borre un registro de la aplicación, el sistema mantendrá la información en la base de datos durante un tiempo determinado (configurable por los administradores).

Una vez que tenemos toda la base de la aplicación creada, se ha añadido la lógica a los distintos módulos, para ello hemos utilizado algunos de los componentes del framework que serán descritos mas adelante. Para poder validar que las funcionalidades desarrolladas son correctas, cada nueva adición al proyecto ha sido testeada instalando las nuevas modificaciones en la instancia de pruebas. Se ha utilizado un sistema de versiones para facilitar que, en caso de error irrecuperable, se pueda retroceder a la última versión estable recuperando el último paquete correcto.

Por último se ha incluido lógica en las vistas utilizando JQuery, esto nos ha permitido modificar dinámicamente el contenido de las mismas basándonos en el valor de determinados campos. También hemos utilizado este recurso para realizar validaciones sobre los campos y mostrar mensajes de error para guiar al usuario cuando cometa alguna equivocación que pudiese dejar el sistema en un estado inconsistente.



## 4.2 SuiteCRM framework

Como ya hemos comentado, SuiteCRM está basado en SugarCRM, aunque lleva ya varios años funcionando como proyecto independiente y añadiendo nuevas funcionalidades con mucha frecuencia, todo el núcleo de SuiteCRM sigue siendo la versión más reciente de SugarCRM Community Edition, motivo por el cual los nombres de las clases principales mantienen el prefijo “Sugar”.

### 4.2.1 Manifest.php

El archivo manifest.php es el fichero que permite al instalador de SuiteCRM distribuir los distintos tipos de archivos a sus distintas ubicaciones y realizar las tareas necesarias para que se integren con el resto de la aplicación.

Para generar un paquete que se pueda instalar en cualquier instancia de SuiteCRM se deberá generar un archivo .zip que incluya todos los archivos del proyecto y el fichero manifest.php en su raíz.

Encontramos distintos apartados dentro del manifest para los distintos tipos de archivos (idioma, nuevos campos, nuevos módulos, relaciones ... ) y una categoría “copy” que nos permite mover el archivo a cualquier ubicación.

No vamos a entrar en detalle de la distribución de los archivos de SuiteCRM, pero sí que es importante conocer que todos los ficheros pertenecientes a los módulos base se encuentran dentro del directorio “modules” en la raíz de la instancia y que, cualquier modificación que se deba realizar sobre dichos ficheros debe ir al directorio “custom/modules/”. De esta forma conseguimos que las actualizaciones de SuiteCRM no afecten al núcleo de la aplicación, y por lo tanto en futuras actualizaciones del sistema seguirá siendo coherente y los desarrollos realizados dentro del directorio “/custom/” deberían seguir funcionando, consiguiendo que los desarrollos sean *upgrade safe*.

### 4.2.2 Logichooks

SuiteCRM permite introducir lógica tras la ejecución de determinados eventos a través de la definición de logichooks.

Para mantener una coherencia con la estructura del proyecto, se ha creado la clase [Module]Manager.php en la raíz de cada módulo, incluyendo en ella todas las funciones del módulo que serán asignadas a un logichook.

Todas las funciones asignadas a un logichook recibirán como variables de entrada “\$bean”, “\$event” y “\$arguments”. \$bean es una instancia de un objeto SugarBean, y es una representación del modelo del módulo que ha disparado el evento. De ella podemos extraer el valor de los distintos

campos y obtener instancias de los registros de otros módulos con los que está relacionado. Además también contiene información sobre el estado del registro antes de que se produjese el evento que disparó la función.

Los distintos logichooks utilizados en este proyecto han sido:

- `before_save`: Se ejecuta antes de que se guarde un registro en un módulo. En este proyecto lo hemos utilizado para rellenar algunos campos autogenerados o realizar validaciones sobre los datos previamente a ser guardados.

El código mostrado a continuación es una función definida en la clase `ContactManager` y asignada a un logichook `before_save`. Su función es rellenar los campos “bic” e “iban” basándose en los campos relativos a la cuenta bancaria ya validados previamente.

```
function feedAccountFields(&$bean, $event, $arguments) {  
    if (!empty($bean->codigo_entidad) && !empty($bean->codigo_cuenta) && !empty($bean->codigo_dc) && !  
empty($bean->codigo_iban) && !empty($bean->codigo_sucursal)){  
        global $app_list_strings;  
        $bean->bic = $app_list_strings['cod_entity_cod_bic'][$bean->codigo_entidad];  
        $bean->iban = $bean->codigo_iban . $bean->codigo_entidad . $bean->codigo_sucursal . $bean->  
codigo_dc . $bean->codigo_cuenta;  
    }  
}
```

- `after_save`: Se ejecuta inmediatamente después de que el registro se haya guardado. Su uso implica que el registro se ha guardado correctamente y cualquier error que se produzca durante un logichook `after_save` podrá dejar el sistema en un estado inconsistente.

Cuando trabajamos con los logichooks `after_save` y `before_save` hay que tener especial cuidado a la hora de actualizar los registros. Realizar un guardado sobre el elemento que ha disparado el evento puede provocar la aparición de bucles infinitos si no se controlan con cuidado.

- `before_delete`: Se disparará antes de que se produzca el borrado de un registro. Es de especial utilidad para comprobar que el borrado de un registro es válido. En este proyecto lo hemos utilizado para actuar ante el borrado de un alta o baja, en caso de que se trate de borrar un registro que no es el último relacionado con el asociado bloqueamos el borrado para evitar un estado inconsistente del sistema. En caso de que el registro borrado sea el último, el borrado es correcto y deberemos revertir el estado del asociado y generar una nueva entrada en el histórico de cambios.
- `before_relationship_add`: Es ejecutada antes de que se añada una relación con otro módulo. En los eventos disparados por añadir o

borrar relaciones podemos encontrar la información sobre el módulo relacionado en la variable. A continuación muestro un fragmento de la función procesarSancion() extraído de la clase SancionesManager que, tras añadir una relación con un asociado, determina si esta es válida o no en base al estado (sancionado) del asociado.

```
$invalid = FALSE;

if ($arguments['related_module'] == 'Contacts'){

    $this->asocLog->printLog("## (procesarSancion) Related module contacts", 'debug');

    $contact = BeanFactory::getBean('Contacts', $arguments['related_id']);

    if ($contact->sancion == '1'){

        $invalid = TRUE;

        $msg = "Error al crear sanción, el asociado ya está sancionado";

    } else {

        $contact->load_relationship('asoc_sanciones_contacts');

        foreach ($contact->asoc_sanciones_contacts->getBeans() as $ele) {

            if (dateToMysql($ele->fecha_inicio) > date("Y-m-d") && $ele->id != $bean->id) {

                $invalid = TRUE;

                $msg = "Error al crear sanción, el asociado ya tiene una sanción planificada";

            }

        }

    }

}
```

- after\_relationship\_add: Se ejecutará después de que se añada una relación con otro módulo. En el proyecto lo hemos utilizado para completar campos del registro con información del módulo relacionado.

### 4.2.3 SugarController

SugarCRM implementa el patrón de diseño modelo vista controlador (MVC), realizando una división entre la lógica de negocio y la capa de presentación.

El modelo es la representación de la información del objeto que deberá ser enviada a la capa de presentación. En Sugar se representa por la clase SugarBean

La vista es la capa responsable de presentar la información del modelo al usuario.

El controlador es la capa responsable de la lógica de de la aplicación. Se encarga de realizar las operaciones como guardar o decidir que vista renderizar en cada momento.

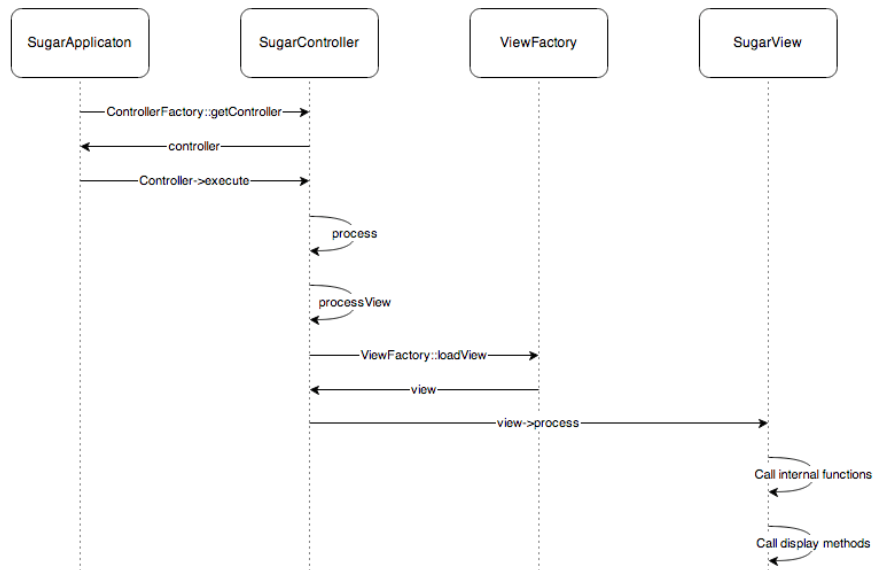


Imagen 4: MVC SugarCRM

SugarCRM permite al desarrollador añadir nuevas funciones a los distintos módulos creando un archivo “controller.php” que extienda la clase SugarController.

En este caso hemos utilizado esta funcionalidad para crear las acciones que deberán realizar los distintos módulos a disposición del usuario. Para ello hemos creado estas funciones con el prefijo “action\_” de forma que el framework los añada como puntos de acceso y puedan ser llamados desde la interfaz de usuario.

#### 4.2.4 SugarView

Como ya hemos comentado, las vistas son las responsables de procesar la información enviada por el controlador y presentársela al usuario. En SugarCRM encontramos 3 vistas distintas (lista, detalle y edición), aunque el framework permite la creación de nuevas vistas personalizadas extendiendo la clase SugarView.

SugarCRM también permite la personalización de las vistas existentes por defecto extendiendo la clase correspondiente (ViewDetail, ViewEdit o ViewList) o, en el caso de modificar una vista de un módulo ya existente, extendiendo de esta. En nuestro caso hemos utilizado esta funcionalidad únicamente para cargar las distintas funciones JQuery que hemos creado para modificar los datos. A continuación podemos ver la clase CustomContactsViewDetail en la que modificamos la vista de detalle del módulo de contactos (Asociados) para incluir las funciones correspondientes.

```

<?php
if(!defined('sugarEntry') || !sugarEntry) die('Not A Valid Entry Point');
  
```

```
require_once('modules/Contacts/views/view.detail.php');

class CustomContactsViewDetail extends ContactsViewDetail
{
    public function display(){
        parent::display();

        echo "<script src='custom/include/asoc_Functions/asoc_functions.js'></script>";

        echo "<script src='custom/modules/Contacts/views/detailViewManager.js'></script>";
    }
}
```

Al acceder a un módulo nos encontraremos con la vista de lista. En ella, en la parte superior encontraremos un área en el que podemos realizar búsquedas basándonos en los campos del módulo que hayan sido incluidos (el administrador de la aplicación puede escoger los campos que quieran utilizar para las búsquedas) y en la parte inferior visualizaremos los resultados. Por defecto los resultados se mostrarán ordenados en base a su fecha de creación.

Nombre	Concepto	Estado	Importe	Remesar	Pago	Remesa
Cuota anual Familia Sanz 2016-07-06	Cuota anual	Emtida	312.00	✓	Cuota anual Familia Sanz 2016-07-06	Remesa Julio 2016
Cuota anual Tammie Barbieri Arizola 2016-07-06	Cuota anual	Emtida	195.00	✓	Cuota anual Tammie Barbieri Arizola 2016-07-06	Remesa Julio 2016
Cuota anual Sibyl Aveles Bonelli 2016-07-06	Cuota anual	Emtida	195.00	✓	Cuota anual Sibyl Aveles Bonelli 2016-07-06	Remesa Julio 2016
Cuota anual Victoria Batta Barrett 2016-07-06	Cuota anual	Emtida	195.00	✓	Cuota anual Victoria Batta Barrett 2016-07-06	Remesa Julio 2016
Cuota anual Bryce Barreiro Bloomgren 2016-07-06	Cuota anual	Emtida	195.00	✓	Cuota anual Bryce Barreiro Bloomgren 2016-07-06	Remesa Julio 2016
Cuota anual Ethan Assad Ayuso 2016-07-06	Cuota anual	Emtida	195.00	✓	Cuota anual Ethan Assad Ayuso 2016-07-06	Remesa Julio 2016
Cuota anual Valrie Berczy Ballena 2016-07-06	Cuota anual	Emtida	195.00	✓	Cuota anual Valrie Berczy Ballena 2016-07-06	Remesa Julio 2016
Cuota anual Wonda Baran Beachell 2016-07-06	Cuota anual	Emtida	195.00	✓	Cuota anual Wonda Baran Beachell 2016-07-06	Remesa Julio 2016
Cuota anual Steven Almeter Boga 2016-07-06	Cuota anual	Emtida	195.00	✓	Cuota anual Steven Almeter Boga 2016-07-06	Remesa Julio 2016
Cuota anual Shariene Baber Bartlett 2016-07-06	Cuota anual	Emtida	195.00	✓	Cuota anual Shariene Baber Bartlett 2016-07-06	Remesa Julio 2016
Cuota anual Shalandia Betry Ashaf 2016-07-06	Cuota anual	Emtida	195.00	✓	Cuota anual Shalandia Betry Ashaf 2016-07-06	Remesa Julio 2016
Cuota anual Pepe Garcia Garcia 2016-07-06	Cuota anual	Emtida	97.50	✓	Cuota anual Pepe Garcia Garcia 2016-07-06	Remesa Julio 2016
Cuota anual Juan Zas 2016-07-06	Cuota anual	Emtida	32.50	✓	Cuota anual Juan Zas 2016-07-06	Remesa Julio 2016

Imagen 5: Vista de lista módulo Cuotas

Desde la vista de lista podemos acceder a la vista de detalle o de edición de cualquiera de los registros mostrados.

La vista de detalle de un módulo es una representación en modo sólo lectura de la información del registro accedido. En ella encontramos además de todos los campos del registro en la parte superior, la representación en forma de subpaneles de la información del los registros relacionados de otros módulos. De esta forma, en nuestra aplicación, accediendo a la vista de lista de un asociado podremos visualizar sus pagos, facturas, altas y bajas, sanciones... y podremos acceder a cada una de ellas. Los subpaneles que queremos mostrar y los campos que visualizaremos en cada uno de ellos es también configurable por los usuarios administradores de la aplicación.





Historico de cambios					
Nombre	Tipo	Valor antiguo	Valor nuevo	Fecha del cambio	
Alta Steven Almeter Boga 2016-07-06	Alta		Alta	07/06/2016	editar

Altas y Bajas		
Nombre	Tipo	Fecha efectiva
Alta Steven Almeter Boga 2016-07-06	Alta	02/18/2016

Pagos						
Nombre	Estado	Concepto	Tipo	Importe	Fecha de ejecución	
Cuota Gimnasio Steven Almeter Boga 2016-07-06	Convertido a factura	Cuota Gimnasio	Pago: Asociado paga a Asociación	15.00	07/06/2016 12:00	editar
Cuota Libre Senior A Steven Almeter Boga 2016-08-16	Convertido a factura	Cuota Libre Senior A	Pago: Asociado paga a Asociación	100.00	08/16/2016 12:00	editar
Comida Senior B Julio 2016 Steven Almeter Boga 2016-07-06	Convertido a factura	Comida Senior B Julio 2016	Pago: Asociado paga a Asociación	30.00	07/06/2016 12:00	editar
Cuota anual Steven Almeter Boga 2016-07-06	Convertido a factura	Cuota anual	Pago: Asociado paga a Asociación	195.00	07/06/2016 12:00	editar
Cuota Libre Senior A Steven Almeter Boga 2016-08-16	Convertido a factura	Cuota Libre Senior A	Pago: Asociado paga a Asociación	100.00	08/16/2016 12:00	editar
Cuota Libre Senior A Steven Almeter Boga 2016-08-16	Convertido a factura	Cuota Libre Senior A	Pago: Asociado paga a Asociación	30.00	08/16/2016 12:00	editar
Cuota Libre Senior A Steven Almeter Boga 2016-08-16	Convertido a factura	Cuota Libre Senior A	Pago: Asociado paga a Asociación	100.00	08/16/2016 12:00	editar

*Imagen 6: Subpaneles vista detalla módulo Asociados*

Los subpaneles también pueden ser utilizados para crear nuevas relaciones del registro con el módulo en cuestión e incluso podremos crear nuevos registros en dicho módulo a través de un formulario reducido o acceder al formulario de edición completo para rellenar toda la información necesaria.

Desde la vista de detalle también podremos realizar acciones sobre el registro desplegando el menú contextual que encontramos en la parte superior a la derecha del botón “Editar”, por defecto encontramos las acciones “Duplicar”, “Eliminar”, “Buscar duplicados” y “Ver registro de cambios”. En este proyecto hemos utilizado ese menú para añadir las distintas acciones que hemos desarrollado para cada uno de los módulos, como por ejemplo “Dar de alta” y “Dar de baja” en el módulo de Asociados o “Buscar facturas por cuota” y “Generar documento XML” en el módulo de Remesas.

Accederemos a la vista de edición de un registro cuando queramos modificar alguno de los campos (o queramos crear un registro nuevo). Al presionar el botón “Guardar” que encontraremos tanto en la parte superior como en la inferior, los datos del registro serán validados y, en caso de que la validación sea correcta, enviados al servidor para ser almacenados en la base de datos, activando los logichook correspondientes descritos anteriormente.

## 4.3 Herramientas de SuiteCRM

### 4.3.1 Constructor de módulos

El constructor de módulos es una herramienta accesible para los usuarios con permisos de administrador que permite la creación y el despliegue de nuevos módulos a través de una interfaz web amigable. Aunque no permite añadir ningún tipo de lógica a los módulos creados, si



que resulta de utilidad para dar los primeros pasos en cualquier desarrollo para SuiteCRM.

En este proyecto se ha utilizado esta herramienta para elaborar el código inicial de los módulos, creando así la estructura básica del proyecto sobre la cual se han realizado las modificaciones necesarias para construir el resultado deseado.

### 4.3.2 Estudio

Esta herramienta proporciona a los usuarios administradores la posibilidad de modificar algunos aspectos de los módulos, tanto del core de la aplicación como los desarrollados por terceras partes.

Permite la creación de nuevos campos, editar las etiquetas visibles, establecer relaciones con otros módulos y modificar el orden y los campos que aparecen en las vistas de los diferentes módulos.

Aunque en este proyecto se han creado los campos necesarios para satisfacer las funciones básicas comunes a la mayoría de asociaciones, se espera que cada una de estas tenga necesidades particulares que no queden satisfechas con lo desarrollado en este proyecto. Por ello esta herramienta puede ser de gran utilidad, ya que permitirá que usuarios con poco o ningún conocimiento de programación puedan realizar las personalizaciones sobre los módulos desarrollados para acabar de adaptar el diseño de la aplicación a sus necesidades.

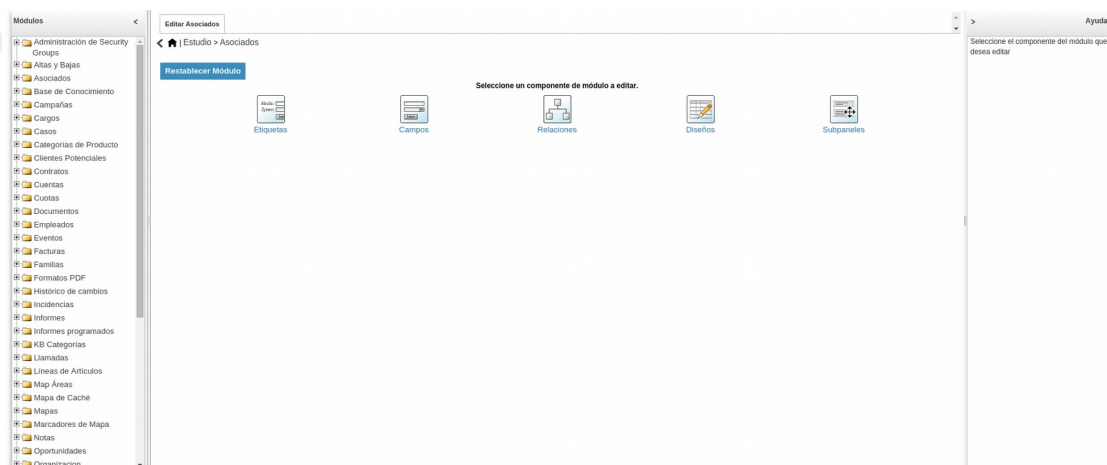


Imagen 7: Herramienta Estudio

## 4.4 Detalles del desarrollo

### 4.4.1 Facturación

El sistema de generación de pagos y facturas ha sido desarrollado de manera que permite diversas configuraciones y se puede utilizar de diversas maneras adaptándose a las necesidades particulares de cada asociación.

Las cuotas que tengan el checkbox “Activa” marcado podrán generarán pagos para un subconjunto de los asociados de alta en la asociación cada vez que un usuario active la acción “Generar pagos” a través del link proporcionado en el menú contextual de la vista de detalle de la cuota en cuestión.

Para permitir que se puedan representar en el sistema el mayor número de situaciones posibles en el contexto de una asociación, hemos utilizado los campos “Tipo”, “Fecha inicio” y “Periodicidad” de la cuota y “Forma de pago” del asociado, para determinar la frecuencia de generación de pagos permitida.

A continuación vamos a describir el proceso seguido para generar los pagos:

Primero se determinará el subconjunto de asociados para los cuales se deberá generar un pago.

Los asociados a los que se les puede generar algún pago serán únicamente aquellos que se encuentren de alta dentro del sistema y que tengan marcado el checkbox “Generar pagos” en su ficha de asociado.

La cuota puede ir dirigida a todos los asociados o puede realizar un filtrado de asociados basándose en los campos edad y perfil del asociado.

Si el asociado pertenece a una familia y tiene marcado el checkbox “Familia” se generará un único pago agrupando a todos los miembros de la familia. Dado que no se puede realizar un filtrado de familias en base a la edad o al perfil, únicamente las cuotas dirigidas a todos los asociados generarán pagos para las familias. Los descuentos relativos a la edad y al perfil de los asociados de la familia si que se aplicarán con normalidad si la cuota así lo indica.

Si la cuota tiene marcado el checkbox “Todos los asociados”, no se realiza ningún filtrado de los asociados basándose en el perfil. En caso contrario se deberá seleccionar los perfiles que determinarán el conjunto de asociados para los que se generarán pagos.

Si la cuota tiene marcado el checkbox “Todas las edades”, no se realizará ningún filtrado de los asociados basándose en la edad. En caso contrario será necesario indicar la edad mínima y máxima (ambas incluidas) de los asociados que formarán el subconjunto.

Si la cuota no tiene marcado el checkbox “Nuevos asociados”, se realizará un filtrado basándose en la fecha de creación de la cuota para impedir que los asociados introducidos en la aplicación de forma posterior a la cuota se añadan al subconjunto. Esto resultará especialmente útil para representar derramas, eventos puntuales como celebraciones o comidas y otras obligaciones de pago que sean únicamente válidas en un momento determinado.



Una vez seleccionado el subconjunto de asociados para los que se deberá generar el pago se determinará si se debe crear un pago para el asociado basándose en el campo “Tipo” de la cuota.

### Cuotas únicas:

Generarán un único pago para cada asociado dentro del subconjunto, la combinación de este tipo de cuota junto al filtrado por perfil permite representar el pago de facturas correspondientes a ciertos eventos como puede ser la compra de material,, la asistencia a un viaje programado por la organización, la celebración de una comida o el pago de una cuota de inscripción.

### Cuotas periódicas:

Las cuotas periódicas generarán pagos para los asociados basándose en la “Fecha de inicio”, la fecha actual, el campo “Periodicidad” y el número de pagos generados previamente para el asociado en cuestión.

### Cuotas anuales:

Las cuotas anuales generarán un pagos basándose en los campos “Fecha de inicio” de cuotas, la fecha actual y “Forma de pago” de asociados.

Por último calcularemos el importe aplicado a cada asociado, para ello seguimos las siguientes reglas:

Si la cuota tiene activo el checkbox “Aplicar descuentos por perfil” o “Aplicar descuentos por rango de edades”, utilizaremos los campos “Perfil” y “Edad” del asociado y de la cuota para reducir el importe de la cuota.

El descuento aplicado a cada perfil y a cada rango de edades debe ser configurado previamente desde el área de administración. El cálculo utilizado para determinar el importe final es el siguiente:

$$importeFinal = (1 - \sum \frac{descuento}{100}) * importeCuota$$

Si el importeFinal es negativo el sistema generará un error advirtiendo de que no se pueden generar facturas de importe inferior a 0 y facilitando un link con el que poder acceder a la vista de detalle del asociado que ha generado el error.

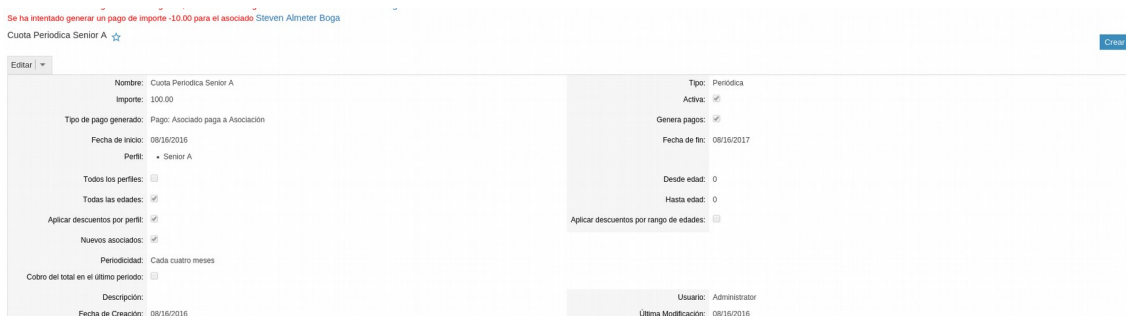


Imagen 8: Vista detalle Cuotas - Error importe negativo

Si la cuota es de tipo anual el importe indicado en la cuota deberá corresponder al importe cobrado en 1 año ya que dicho importe se fraccionará en diversos pagos dependiendo del campo “Forma de pago” del asociado.

Una vez generados los pagos, nos fijaremos en el campo “Generar facturas” del asociado y, en caso de que esté activo, generaremos una factura basada en la información del pago recién generado. En el caso contrario los usuarios tendrán la posibilidad de validar los pagos generados y convertirlos manualmente a facturas a través de la acción con el mismo nombre, accesible desde el menú contextual de la vista de detalle del pago. Este diseño permite a los usuarios poder marcar y tratar los casos individuales, que tengan unas necesidades de facturación que no hayan quedado cubiertas en el desarrollo de este proyecto a la vez que facilita la automatización para los casos mayoritarios.

Una vez generadas las facturas los usuarios tendrán la posibilidad de utilizar el módulo de Remesas para generar el documento XML correspondiente y poder enviarlo al banco.

Para realizarlo se deberá crear un nuevo registro en el módulo remesas, y dependiendo de las necesidades de la asociación se ofrecen dos posibilidades.

La acción “Buscar todas las facturas por tipo” relaciona con esta remesa todas las facturas en estado emitido, que no tengan relación con ninguna otra remesa y que sean del mismo tipo que la remesa. Para poder utilizarla es necesario que la remesa no esté relacionada con ninguna cuota, en caso contrario la acción no será visible.

Por otro lado, al crear una remesa y relacionarla con una cuota tenemos visible la acción “Buscar facturas por cuota”. Dicha acción relaciona con esta remesa todas las facturas, en estado emitido y sin otra relación con otra remesa, que hayan sido generadas por la cuota relacionada.

El usuario podrá añadir manualmente cualquier factura generada a esta remesa antes de generar el documento XML definitivo.

Al añadir o eliminar una factura de la remesa se recalcula el valor del campo “Importe previsto” de la misma, lo que facilitará la validación de los datos por parte del usuario antes de generar el documento final.

Cuando se haya generado un documento válido se visualizará la acción “Validar remesa” la cual cambia el estado de todas las facturas relacionadas a “Pagadas”.



#### **4.4.2 Cambios realizados**

Originalmente el campo “Perfil” del asociado estaba pensado como una lista en la que el usuario tendría que asignar un único valor de los configurados desde el área de administración.

Esto ha sido modificado para convertir el campo en “multienum” el cual permite seleccionar todos los valores que se deseen de dicha lista. Esto se ha realizado para permitir más flexibilidad a la hora de programar las cuotas y poder representar con una mayor facilidad las distintas cuotas a las cuales se puede suscribir el asociado.

Se ha añadido un nuevo tipo de cuota “Libre” para permitir que el usuario genere pagos para los asociados dentro del subconjunto indicado sin ningún tipo de restricción temporal. La tarea programada “Generar pagos” no creará pagos relativos a este tipo de cuotas.

#### **4.4.3 Fichero de log**

Para facilitar el testeo y la depuración de la aplicación todas las clases implementadas en este proyecto utilizan el sistema de logueo de SuiteCRM basado en niveles. Adicionalmente se ha diseñado una clase que escribe todos los mensajes generados por las funciones implementadas en este desarrollo en un fichero personalizado. Esto facilita la trazabilidad del código para las labores de monitorización y el desarrollo de futuras actualizaciones.

El nombre y la utilización de este log adicional se puede configurar desde el área de administración, su uso no es obligatorio y está pensado únicamente para entornos de pruebas o para instancias recién instaladas y en ningún caso interfiere con el logueo habitual de SuiteCRM.

## **5 Casos de Estudio**

En este capítulo veremos dos escenarios en los que la aplicación puede configurarse para adaptarse a las necesidades de un club deportivo y de una falla.

Para ello primero comentaremos las necesidades que presenta cada organización (centrándonos en los distintos tipos de facturación que realizan) y a continuación mostraremos las soluciones propuestas para cada una de ellas.

### **5.1 Club deportivo**

Un club deportivo es una organización dedicada a la práctica de un deporte que usualmente gestiona más de un equipo, divididos por edades o en categorías reguladas por una federación. Adicionalmente los clubes deportivos pueden ofrecer otros servicios complementarios, como puede ser el acceso a un gimnasio, entrenadores particulares, salas de estudio o consulta con un fisioterapeuta.

Los miembros del club deportivo deberán siempre realizar una serie de pagos dependientes del reglamento de la propia organización, las características personales del asociado y de los servicios que quiera hacer uso.

#### **5.1.1 Necesidades de la asociación**

- Separación de asociados por equipos.
- Pagos puntuales por equipos.
- Cuota anual: cada asociado escoge la periodicidad de los pagos.
- Gimnasio: el asociado se suscribe durante un periodo de tiempo y los pagos se producen mensualmente.
- Alquiler de pista de padel por horas.
- Cuota de inscripción.
- Descuentos por familia numerosa y por pertenecer al equipo principal.

#### **5.1.2 Soluciones**

Para lograr satisfacer todas las necesidades hemos utilizado el campo “Perfil” para representar varios casos.



Por un lado introduciremos los distintos equipos que encontramos en el club, indicando en todos los casos descuento 0 excepto en el equipo principal (Senior A) que tiene un descuento del 40%.

Por otro lado para poder representar la suscripción a la cuota de Gimnasio hemos creado también una entrada en esta lista.

Por último hemos añadido el perfil “Familia numerosa” para poder aplicar el descuento correspondiente a los asociados con dicho perfil.

### Lista de perfiles

Añadir nuevo perfil

Nombre	Descuento aplicado
Senior A	20
Senior B	0
Senior C	0
S18	0
S16	0
Familia numerosa	40
Gimnasio	0

[Editar Perfiles](#)

*Imagen 9: Lista perfiles club deportivo*

Una vez que la lista de perfiles ha sido creada podemos editar los asociados para indicar el equipo del que forma parte, si es parte de una familia números y si está suscrito a la cuota de gimnasio.



*Imagen 10: Club deportivo - Campo perfil asociado*

Por último hemos configurado las cuotas necesarias para generar todos los pagos de un club deportivo.



### ***Cuota de inscripción***

Cada asociado pagará una cantidad fija en el momento de darse de alta por primera vez en el club. Para representar este escenario hemos creado una cuota con las siguientes características:

- Tipo: única
- Aplicar descuentos por perfil: No
- Todos los perfiles: Si
- Nuevos asociados: Si

### ***Cuota anual***

Representa la cantidad anual que debe pagar cada jugador, la periodicidad y el importe de los pagos generados dependerá de la forma de pago del asociado.

- Tipo: Anual
- Aplica descuentos por perfil: Si
- Todos los perfiles: Si

### ***Cuota Gimnasio***

Esta cuota sólo deberá generar pagos una vez al mes para los asociados que tengan entre sus perfiles el valor "Gimnasio". La configuración que hemos realizado es la siguiente:

- Tipo: Periódica
- Periodicidad: Una vez al mes
- Todos los asociados: No
- Perfil: Gimnasio
- Aplicar descuentos por perfil: No

### ***Una hora de padel***

Esta cuota no debe generar pagos automáticamente ni va dirigida a asociados con ningún perfil en particular. Está pensada para que los usuarios puedan crear rápidamente los pagos correspondientes a la reserva de la pista de padel utilizando la acción "Crear pagos manualmente" accesible desde el menú contextual de la cuota.

El único punto importante de su configuración es que el checkbox "Generar pagos" debe estar desactivado.



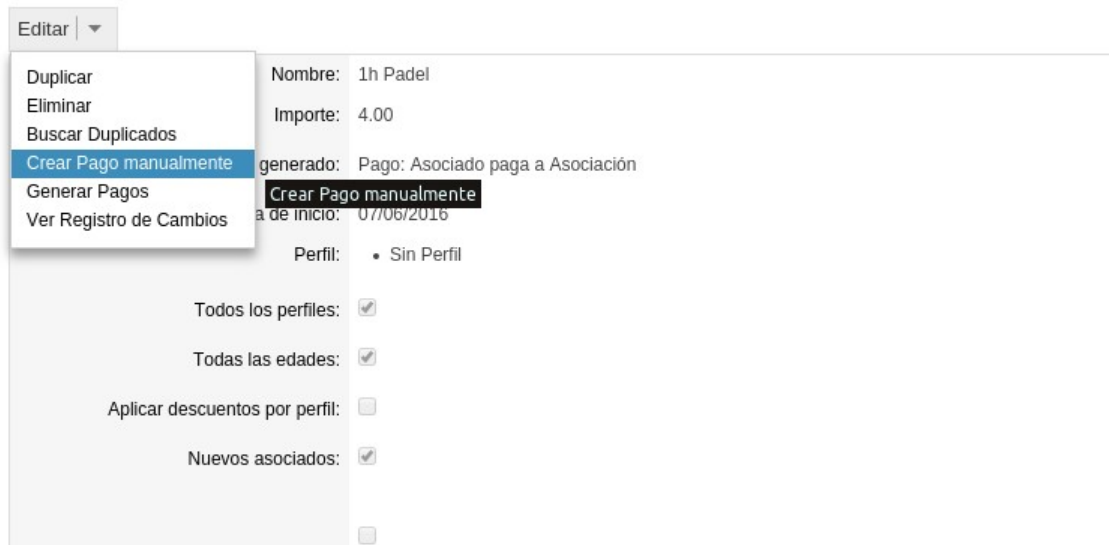


Imagen 11: Vista detalle Cuotas - Crear pago manualmente

### Eventos

Por último los usuarios podrán crear cuotas para representar eventos como comidas, viajes, compra de material. Para ello se deberá crear una cuota similar a la siguiente, pensada para generar los pagos relativos a una comida celebrada por el equipo Senior B, que generará un único pago para los integrantes del equipo.

- Tipo: Única
- Nuevos asociados: No
- Todos los asociados: No
- Perfil: Senior B
- Aplicar descuentos por perfil: No

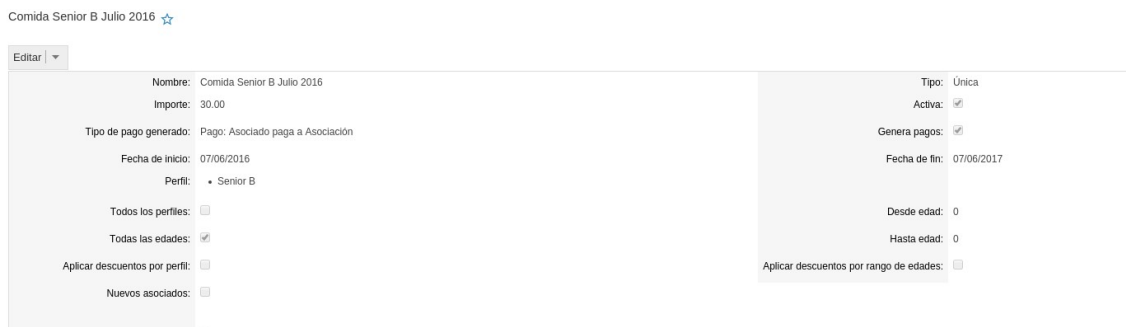


Imagen 12: Vista detalle Cuotas - Comida club deportivo

## 5.2 Comisiones falleras

Una comisión fallera, comúnmente llamada falla, es una agrupación de personas (falleros) que se reúnen para celebrar este famoso evento. Aunque la mayor parte de sus actividades se concentran en el mes de marzo, estas organizaciones realizan distintos tipos de eventos durante todo el año. Para permitir el desarrollo de estas actividades, así como costear el alquiler del casal y los monumentos falleros, las fallas buscan constantemente distintas fuentes de ingresos. Parte de estos ingresos los obtienen del cobro de una cuota aplicada a todos los miembros de la asociación.

En este ejemplo veremos, además de como configurar la cuota para generar los pagos necesarios, como utilizar los módulos de gestión de cargos para representar la estructura organizativa de la falla y como configurar el módulo de informes propio de SuiteCRM combinado con los módulos desarrollados en este proyecto.

### 5.2.1 Necesidades de la asociación

#### ***Cobro de cuotas:***

Las fallas necesitan realizar un cálculo del importe de las cuotas personalizado por cada asociado en base a su edad y, si forma o no parte de una familia. La falla cobra la cuota de todo el año entre los meses de Julio y Enero, realizando los cobros el 1 de Julio, 1 de Septiembre, 1 de Noviembre y 1 de Enero.

Los descuentos que aplica la falla son los siguientes:

- Bebés: 90%
- Infantiles: 50%
- Juveniles: 25%
- Jubilados mayores de 65: 10%

#### ***Cargos:***

Anualmente los miembros de la falla se reúnen para realizar la asignación de los cargos para el siguiente ejercicio.

A continuación presentamos los cargos más comunes englobados en distintas categorías:

- Presidencia: Presidente, Vicepresidente 1, Vicepresidente 2, Asesor presidente 1, Asesor presidente 2.
- Secretaría: Secretario, Vicesecretario 1 y Vicesecretario 2.
- Economía: Tesorería, Delegado Lotería, Delegado Cuotas y Delegado Tribuciones.



- Festejos: Delegado 1, Delegado 2, Vicedelegado 1 y Vicedelegado 2.
- Infantiles: Delegado 1, Delegado 2, Vicedelegado 1 y Vicedelegado 2.
- Cultura: Delegado 1, Delegado 2, Vicedelegado 1 y Vicedelegado 2.
- Media: Delegado web, Redes sociales y Medios de comunicación.

### 5.2.2 Soluciones

Para representar los descuentos por pertenecer a una familia introduciremos los descuentos correspondientes en el campo perfil del área de administración. En este caso el descuento se aplica a partir del tercer miembro de la familia, gozando este de un 10% y el cuarto y siguientes de un 15%.

#### Lista de perfiles

Añadir nuevo perfil

Nombre	Descuento aplicado
Segundo miembro	10
Tercer miembro	15
Cuarto miembro y siguiente:	25

[Editar Perfiles](#)

*Imagen 13: Lista perfiles Falla*

Para incluir los descuentos por edad aplicados a los asociados rellenaremos la lista de edades accesible en el área de administración.

#### Lista de Edades

Añadir nuevo rango de edad

From	To	Descuento aplicad
01	06	90
07	12	50
13	18	25
65	99	10

[Editar rangos de Edad](#)

*Imagen 14: Lista edades Falla*

Para la representación de la estructura organizativa de la falla utilizaremos los módulos órganos, donde crearemos las categorías presidencia, secretaría, economía, etc., cargos, donde definiremos cada uno de los cargos y lo relacionaremos con el órgano correspondiente, y organización, en el que indicaremos que fallero ocupara cada cargo durante este ejercicio.

### **Cuota**

Para poder automatizar la generación de los pagos relativos a la cuota anual configuraremos la cuota de la siguiente manera:

- Tipo: Periódica.
- Importe: Ya que el cobro de la cuota se realiza en cuatro pagos el importe deberá ser una cuarta parte del importe total. En este caso anualmente cada fallero debe pagar 280€ por lo que estableceremos el importe de la cuota a 70€.
- Para que se puedan aplicar los descuentos deberemos marcar las casillas “Aplicar descuentos por perfil” y “Aplicar descuentos por rango de edades”.
- Definimos la fecha de inicio el 1 de Julio y la fecha de fin el 2 de Enero (el 1 de Enero se generarán los últimos pagos)
- Estableceremos la periodicidad cada dos meses.

Cuota anual 2016 - 2017 ☆ Crear

Editar ▾

Nombre:	Cuota anual 2016 - 2017	Tipo:	Periódica
Importe:	70.00	Activa:	<input checked="" type="checkbox"/>
Tipo de pago generado:	Pago: Asociado paga a Asociación	Genera pagos:	<input checked="" type="checkbox"/>
Fecha de inicio:	07/01/2016	Fecha de fin:	01/02/2018
Perfil:	• Sin Perfil	Desde edad:	0
Todos los perfiles:	<input checked="" type="checkbox"/>	Hasta edad:	0
Todas las edades:	<input checked="" type="checkbox"/>	Aplicar descuentos por rango de edades:	<input checked="" type="checkbox"/>
Aplicar descuentos por perfil:	<input checked="" type="checkbox"/>		
Nuevos asociados:	<input checked="" type="checkbox"/>		
Periodicidad:	Cada dos meses		
Cobro del total en el último periodo:	<input type="checkbox"/>		
Descripción:		Usuario:	Administrator
Fecha de Creación:	08/23/2016	Última Modificación:	08/23/2016

*Imagen 15: Cuota anual Falla*

## Cargos

Para introducir la información primero deberemos crear todos los órganos descritos anteriormente. Esto lo haremos desde el módulo de órganos accesible desde la pestaña de Organización. Una vez creados podemos crear los cargos relativos a cada órgano desde la vista de detalle, utilizando para ello el subpanel. Para ello clicaremos sobre el botón “Nuevo” del subpanel de cargos y rellenaremos el formulario para introducir el nombre del cargo.

Media ☆ Crear

Editar | (5 de 7)

Nombre: Media Usuario: Administrator  
Fecha de Creación: 08/23/2016 Última Modificación: 08/23/2016  
Descripción:

**^ Cargos**

Guardar Cancelar Formulario Completo

Nombre: \* Medios de comunicación Usuario: Administrator

Guardar Cancelar Formulario Completo

Nombre	Última Modificación
Redes Sociales	08/23/2016 10:20
Delegado web	08/23/2016 10:20

Imagen 16: Vista detalle órganos, insertando cargos

A continuación deberemos asignar a cada fallero el cargo que desarrolla utilizando el módulo de organización. Esto podemos hacerlo bien desde el propio módulo o mediante el subpanel accesible desde la vista de detalle de cada cargo.

SuiteCRM Cargos Gestión de Asociados Facturación Organización Todo

Acciones  
Crear Cargos  
Vista Cargos  
Importar

Recientes  
Vicedelegado 1  
Vicedelegado 2  
Delegado 2  
Delegado 1  
Secretaría  
Presidencia  
Media  
Infantiles  
Festejos  
Economía

Favoritos

Cargos

Nombre: [ ] Órganos: Economía Mis Elementos [ ] Buscar Limpiar Búsqueda Avanzada

Eliminar (1 - 4 de 4)

Nombre	Órganos	Usuario
Delegado Cuotas	Economía	Administrator
Delegado Tribuciones	Economía	Administrator
Delegado Lotería	Economía	Administrator
Tesorería	Economía	Administrator

Eliminar (1 - 4 de 4)

Grupos de Seguridad: Asignación Masiva  
Asignar Remover

Grupo: -ninguno-

⌚ Tiempo de respuesta del servidor: 0.51 segundos © Supercharged by SuiteCRM © Desarrollado por SugarCRM 🖨️ Imprimir ↑ Volver al parte superior

Imagen 17: Vista de lista Cargos



Mediante el formulario de búsqueda de la vista de lista podemos filtrar los resultado por el órgano deseado y acceder a cada uno a través del enlace facilitado en el nombre.

Realizar la asignación de esta forma resultará de utilidad para casos en los que el asociado vaya a repetir el cargo durante otro ejercicio, ya que permitirá observar cómodamente el histórico de los distintos asociados que han desarrollado el cargo anteriormente.

The screenshot shows the SuiteCRM interface for the 'Delegado Cuotas' record. The top navigation bar includes 'SuiteCRM', 'Cargos', 'Gestión de Asociados', 'Facturación', 'Organización', and 'Todo'. The sidebar on the left lists 'Acciones' such as 'Crear Cargos', 'Vista Cargos', 'Importar', and 'Recientes' with various roles like 'Delegado Cuotas', 'Tesorería', etc. The main content area shows the 'Delegado Cuotas' detail view with a 'Crear' button and an 'Editar' dropdown. The form displays fields for 'Nombre: Delegado Cuotas', 'Fecha de Creación: 08/23/2016', 'Descripción', 'Usuario: Administrator', 'Última Modificación: 08/23/2016', and 'Órganos: Economía'. Below the form is the 'Organización' subpanel, which contains a table of assignments. The table has columns for 'Nombre', 'Asociados', 'Cargos', 'Fecha de inicio', and 'Fecha de fin'. The data rows are:

Nombre	Asociados	Cargos	Fecha de inicio	Fecha de fin	
Delegado Cuotas Economía 2016-2017	Van Awkard Basso	Delegado Cuotas	04/01/2016	03/31/2017	editar
Delegado Cuotas Economía 2015-2016	Van Awkard Basso	Delegado Cuotas	04/01/2015	03/31/2016	editar
Delegado Cuotas Economía 2014-2015	Van Awkard Basso	Delegado Cuotas	04/01/2014	03/31/2015	editar

Imagen 18: Vista detalle Cargos - Subpanel Organización

## Familias

Al crear un nueva familia se deberá indicar si esta utiliza la funcionalidad de generación automática de pagos mediante los campos "Generar Facturas", "Remesar" y "Generar Pagos". También se deberá indicar un miembro de la familia que actuará como contacto de facturación.

Para añadir al resto de miembros de la familia podemos hacerlo tanto desde la vista de detalle del fallero en cuestión, rellenando el campo "familia" o desde el subpanel de Asociados accesible desde la vista de detalle de la familia.

Para añadir nuevos miembros a utilizando el subpanel deberemos abrir el menú contextual situado a la derecha del botón "nuevo" y clicar en "Seleccionar". Esto abrirá una ventana emergente en la que se mostrará un pequeño formulario de búsqueda para localizar al asociado.

## Búsqueda de Asociados

Nombre	<input type="text"/>	Apellidos	<input type="text"/>
Cuenta	<input type="text"/>	Cargo	<input type="text"/>
Toma de contacto	<div style="border: 1px solid #ccc; padding: 2px;"><ul style="list-style-type: none"><li>Llamada en Frío</li><li>Cliente Existente</li><li>Auto Generado</li><li>Empleado</li><li>Socio</li></ul></div>	Cualquier email	<input type="text"/>
Campaña	<input type="text"/>	Asignado a	<div style="border: 1px solid #ccc; padding: 2px;"><ul style="list-style-type: none"><li>Administrator</li></ul></div>

## Lista de Asociados

<input type="checkbox"/>	Nombre	Cuenta	Cargo	Toma de contacto
<input type="checkbox"/>	<b>i</b> <a href="#">Roxy Boden Barcena</a>			
<input type="checkbox"/>	<b>i</b> <a href="#">Tarsha Blakes Alvarado</a>			
<input type="checkbox"/>	<b>i</b> <a href="#">Verline Bolanos Ashmead</a>			
<input type="checkbox"/>	<b>i</b> <a href="#">Wilma Baptista Albu</a>			

(1 - 20 de 100)

Imagen 19: Ventana emergente búsqueda asociados

Por último, si la familia tiene 3 o mas miembros se deberá seleccionar la opción correspondiente en el campo “Perfil” de cada asociado para que se le aplique el descuento correcto.

Al añadir un asociado a una familia se dejarán de generar pagos para el asociado ya que, a partir de ese momento, se generará un único pago conjunto para toda la familia.

## Informes

Para la generación de informes personalizados se podrá hacer uso del módulo “Informes” de SuiteCRM. Este módulo permite configurar a través de una interfaz amigable distintos tipos de informes e incluso proporciona un mecanismo para que se ejecuten automáticamente cada cierto tiempo. En este ejemplo hemos configurado un informe para buscar asociados que se hayan dado de baja (se pueden buscar altas también) en un periodo de tiempo determinado.



Para este ejemplo hemos creado un informe que permite buscar los asociados que se han dado de baja en un periodo de tiempo determinado. Este tipo de informes se suele generar a principios de cada mes para analizar los resultados del mes anterior.

Lo primero que deberemos hacer es crear un nuevo informe, para ello estableceremos el módulo principal y fijaremos los campos que aparecerán en el informe. Desplegando los módulos de la parte izquierda podremos añadir campos de los distintos módulos relacionados.

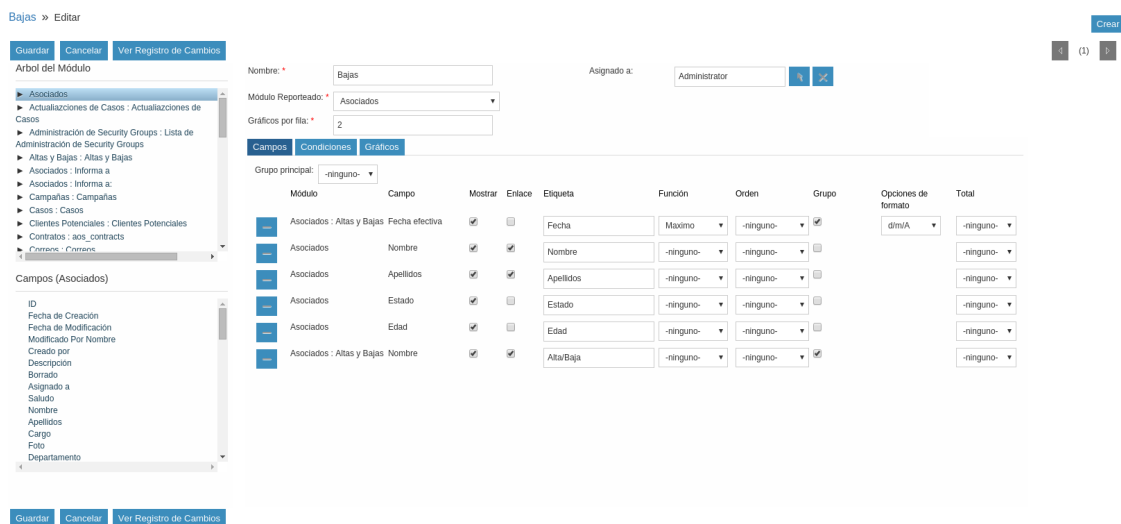


Imagen 20: Vista edición Informes. Seleccionando campos

A continuación, pinchando sobre el botón “Condiciones”, accederemos a la siguiente vista, donde podremos parametrizar los criterios de búsqueda del informe.

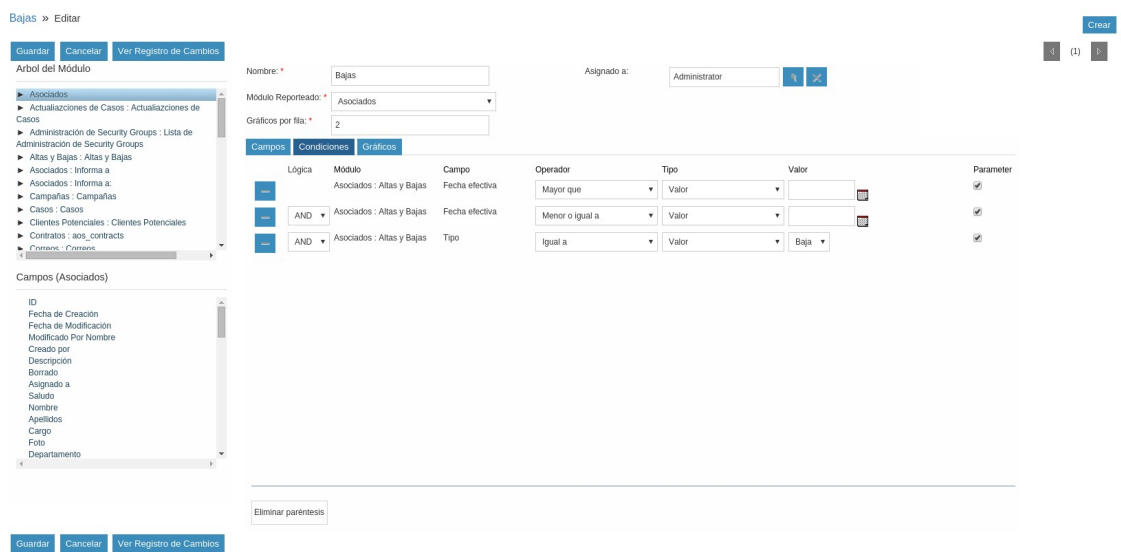


Imagen 21: Vista edición Informes. Condiciones



Para acceder al informe podemos hacerlo de la manera habitual, buscándolo desde la vista de lista del módulo. Una vez dentro deberemos rellenar los campos que hemos seleccionado como parámetros y pinchar en el botón “Ejecutar” para poder ver los resultados del informe.

The screenshot displays the 'Bajas' report configuration and results in SuiteCRM. At the top, there are navigation buttons: 'Editar' (dropdown), 'Crear', and a count '(1)'. Below this is the 'PARAMETERS' section with three rows of filters:

Modulo	Campo	Operador	Tipo	Valor
Asociados : Altas y Bajas	Fecha efectiva	Mayor que	Valor	08/01/2016
Asociados : Altas y Bajas	Fecha efectiva	Menor o igual a	Valor	08/31/2016
Asociados : Altas y Bajas	Tipo	Igual a	Valor	Baja

Below the parameters is an 'Actualizar' button and a 'REPORT' section. The report table has the following data:

Fecha	Nombre	Apellidos	Estado	Edad	Alta/Baja
19/08/2016	Roxy	Boden Barcena	Baja		Baja Roxy Boden Barcena 2016-08-23
21/08/2016	Tarsha	Blakes Alvarado	Baja	26	Baja Tarsha Blakes Alvarado 2016-08-23

Below the report is the 'Informes programados' section, which is currently empty, showing 'Sin Datos'.

At the bottom of the page, there is a footer with the following text: 'Tiempo de respuesta del servidor: 0.69 segundos', '© Supercharged by SuiteCRM © Desarrollado por SugarCRM', and 'Imprimir + Volver al parte superior'.

Imagen 22: Vista detalle Informes

# 6 Conclusiones

## 6.1 Estado del proyecto

El proyecto desarrollado satisface las funcionalidades básicas descritas anteriormente y añade otras que permiten al usuario final mejorar la adaptación con esta aplicación.

El sistema de cuotas ha demostrado tener la flexibilidad necesaria para representar los pagos y los sistemas de facturación de asociaciones con necesidades diferentes.

A través de las distintas vistas del módulo de Asociados, los usuarios pueden controlar cómodamente su estado y realizar búsquedas variadas.

A través del módulo de organización se permite realizar búsquedas sobre los distintos cargos desarrollados por los asociados para consultar el estado de los mismos en distintos instantes de la vida de la asociación.

Todo lo descrito en la especificación por módulos está desarrollado y en funcionamiento a excepción de los siguientes puntos, los cuales están siendo desarrollados para incluir en una actualización futura.

- La inclusión en el módulo Histórico de cambios de los cambios producidos en el Perfil del asociado.
- En el módulo de cuotas, la funcionalidad asignada al checkbox “Cobro total en el último periodo”, diseñada para cuotas anuales, no está implementada todavía.
- La generación de documentos XML SEPA CREDIT TRANSFER.
- La tarea programada para cambiar el estado de las cuotas.

## 6.2 Futuras mejoras

El producto desarrollado se considera una primera versión funcional sobre la cual se van a realizar futuras modificaciones, en algunos casos para adaptar algunos aspectos para cumplir con necesidades de algún cliente en particular, y en otros para aumentar las funcionalidades propias a todas las asociaciones.

Entre los próximos desarrollos que se realizarán sobre el producto encontramos:

- SuiteCRM está diseñado para permitir que cada usuario escoja un idioma distinto. Este proyecto se ha desarrollado teniendo en cuenta esta funcionalidad y se han preparado todas las etiquetas de texto necesarias para poder traducir la aplicación a otros idiomas.



- En la próxima actualización de la aplicación se añadirán los campos necesarios al módulo de histórico de cambios para permitir representar la información relativa a la posesión de cargos (representada a través del módulo de organización) justo al resto de información sobre la vida del asociado dentro de la organización.
- SuiteCRM permite la creación de funciones que se ejecuten antes de que se instale el paquete (`pre_install`) y después de que se haya instalado (`post_install`). En una próxima actualización se aprovechará esta funcionalidad para que se realice automáticamente la configuración inicial.

### 6.3 Valoración personal

La entrega final del proyecto se ha visto retrasada respecto a la estimación inicial, debido a que las fases de análisis e implementación resultaron ser más costosas de lo esperado y el tiempo que se le pudo dedicar fue inferior a lo previsto. Así pues, la versión actual de la aplicación se finalizó a mediados de junio, dejando muy poco margen para la redacción de la memoria. Esta situación motivó que la entrega se realizase en septiembre en lugar de en julio.

Durante el desarrollo de este proyecto se han ido superando las distintas dificultades presentadas durante todo el ciclo de vida de una aplicación. Para lograr finalizarlo con éxito se han utilizado gran parte de los conocimientos adquiridos en el Grado de Ingeniería Informática, así como algunas técnicas obtenidas durante las prácticas de empresa realizadas en Ticnux.

Asignaturas como “Ingeniería del Software” o “Gestión de proyectos” han sido vitales para poder afrontar los primeros pasos del desarrollo y permitir sintetizar la información de una forma ordenada.

Aunque PHP no es un lenguaje de programación que se aprenda en esta titulación, los conocimientos adquiridos en asignaturas como “Programación”, “Estructuras de datos y algoritmos” y “Desarrollo web” han podido ser extrapolados, permitiendo un aprendizaje rápido del lenguaje. La combinación de estas asignaturas, con lo aprendido en otras relativas a las bases de datos, ha proporcionado los mecanismos necesarios para poder dar un soporte completo a una aplicación web.

La elaboración de este proyecto a supuesto un reto muy interesante y ha proporcionado una visión realista de como utilizar las destrezas adquiridas durante la formación en la universidad dentro del entorno laboral.

# 7 Bibliografía

## 7.1 Recursos consultados

- <http://support.sugarcrm.com/>
- <https://aws.amazon.com/es/>
- <http://www.inforworld.com/>
- <http://es.wikipedia.org/wiki>
- <http://www.capterra.com/>
- <http://www.weblinkinternational.com/>
- <https://www.agpd.es/>
- <https://hello.raklet.com/>
- <http://www.wildapricot.com/>
- <http://www.baarasoft.com/socius.htm>
- <http://www.idesoft.es/software-para-asociaciones/>
- <http://www.smarty.net/>
- <http://www.asociaciones.org/>
- <http://www.sepaesp.es/>
- <https://www.mysql.com/>
- <http://www.php.net/>
- <https://jquery.com/>
- <http://www.w3c.es/>
- <https://news.netcraft.com/>

