



UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA  
Escuela de Ingeniería Informática



Proyecto de Fin de Carrera de la Escuela de Ingeniería Informática de la Universidad de Las Palmas de Gran Canaria presentado por los alumnos:

**Alberto Cardona López**

**José Antonio Martín García**

**Título del proyecto:** Diseño e implementación de un sistema de información para una organización social.

**Tutor:** D. Francisco Javier Carreras Riudavets

# Contenido

Contenido .....	2
1. INTRODUCCIÓN .....	7
1.1. SISTEMAS DE INFORMACIÓN .....	7
1.2. ESTADO DEL ARTE .....	7
1.3. SITUACIÓN DE PARTIDA .....	9
1.4. ESTRUCTURA DE LA MEMORIA .....	10
2. OBJETIVOS .....	11
2.1. OBJETIVOS PARA LA ORGANIZACIÓN .....	11
2.2. OBJETIVOS PARA LOS AUTORES DEL PROYECTO .....	11
3. METODOLOGÍA DE DESARROLLO SOFTWARE .....	13
3.1. DOCUMENTACIÓN .....	14
3.2. GESTIÓN .....	15
3.3. FLUJO DE TRABAJO .....	16
3.3.1. Sobre Git .....	16
3.3.2. Revisiones de código .....	16
3.4. EQUIPO DE TRABAJO .....	17
4. RECURSOS SOFTWARE Y HARDWARE .....	18
4.1. RECURSOS SOFTWARE .....	18
4.1.1. Herramientas de desarrollo .....	18
4.1.2. Extensiones de Visual Studio: .....	18
4.1.3. Librería principalñes .....	19
4.2. RECURSOS HARDWARE .....	19
5. PLAN DE TRABAJO .....	21
5.1. ESTUDIO DE LA TECNOLOGÍA .....	21
5.2. ANÁLISIS INICIAL DE REQUISITOS .....	22
5.3. DISEÑO DE LA ARQUITECTURA .....	23
5.4. DESARROLLO ITERATIVO .....	23

5.5.	FINALIZACIÓN .....	24
6.	ESTIMACIÓN DE ESFUERZO Y PRESUPUESTO ESTIMADO .....	25
6.1.	ESTIMACIÓN DE ESFUERZO .....	25
6.2.	ORGANIZACIÓN DEL ESFUERZO .....	26
6.3.	PRESUPUESTO ESTIMADO .....	27
6.3.1.	Supuesto 1: Presupuesto Junior .....	27
6.3.2.	Supuesto 2: Presupuesto Senior.....	27
7.	ESTUDIO DEL DOMINIO DEL PROBLEMA .....	28
7.1.	ACTORES .....	28
7.2.	ENTIDADES .....	29
7.2.1.	Servicio de atenciones .....	29
7.2.2.	Gestión de socios.....	30
7.2.3.	Cooperación.....	31
8.	ANÁLISIS DE REQUISITOS .....	32
8.1.	CARACTERÍSTICAS COMUNES ENTRE LOS MÓDULOS .....	32
8.2.	CARACTERÍSTICAS DEL MÓDULO DE ATENCIONES.....	35
8.3.	CARACTERÍSTICAS DEL MÓDULO DE SOCIOS.....	37
8.4.	CARACTERÍSTICAS DEL MÓDULO DE COOPERACIÓN .....	39
8.5.	CARACTERÍSTICAS DE LA EXTRANET .....	41
9.	DISEÑO ARQUITECTÓNICO.....	43
9.1.	ARQUITECTURA GLOBAL.....	44
9.2.	ORGANIZACIÓN DE ENSAMBLADOS .....	46
9.3.	DESPLIEGUE .....	48
9.3.1.	Funcionamiento del servidor de sincronización.....	49
9.3.2.	Organización de carpetas de la aplicación en el cliente.....	50
9.4.	ARRANQUE DE LA APLICACIÓN.....	50
9.5.	MVVM (MODEL-VIEW-VIEWMODEL) .....	51
9.5.1.	Motivación para usar MVVM .....	51

9.5.2.	MVVM (Model-View-ViewModel).....	52
9.6.	CLASES DE LOS ENSAMBLADOS COMUNES .....	53
9.6.1.	Ensamblado <i>Core</i> .....	54
9.6.2.	Ensamblado <i>Gama.Common</i> .....	55
9.7.	CLASES DEL MÓDULO DE ATENCIONES .....	60
9.7.1.	Capa de acceso a datos ( <i>Gama.Atenciones.DataAccess</i> ).....	61
9.7.2.	Capa de negocio ( <i>Gama.Atenciones.Business</i> ).....	62
9.7.3.	Eventos ( <i>Gama.Atenciones.Wpf.Eventos</i> ).....	63
9.8.	CLASES DEL MÓDULO DE SOCIOS .....	64
9.8.1.	Capa de acceso a datos ( <i>Gama.Socios.DataAccess</i> ).....	64
9.8.2.	Capa de negocio ( <i>Gama.Socios.Business</i> ) .....	65
9.8.3.	Eventos ( <i>Gama.Socios.Wpf.Eventos</i> ).....	66
9.9.	CLASES DEL MÓDULO DE COOPERACIÓN .....	66
9.9.1.	Capa de acceso a datos ( <i>Gama.Cooperacion.DataAccess</i> ) .....	67
9.9.2.	Capa de negocio ( <i>Gama.Cooperacion.Business</i> ) .....	68
9.9.3.	Eventos ( <i>Gama.Cooperacion.Wpf.Eventos</i> ) .....	69
9.10.	ARQUITECTURA CLIENTE-SERVIDOR .....	69
9.11.	DISEÑO DE LA BASE DE DATOS .....	70
9.11.1.	Servicio de atenciones .....	71
9.11.2.	Gestión de socios .....	72
9.11.3.	Cooperación .....	73
9.12.	DECISIONES DE RENDIMIENTO .....	74
9.12.1.	Sobre WPF.....	74
9.12.2.	Sobre el calendario .....	74
9.12.3.	Sobre el arranque de los módulos.....	74
9.12.4.	Sobre la Extranet.....	75
10.	DISEÑO DE LA INTERFAZ DE USUARIO.....	76
10.1.	SELECCIÓN DE MÓDULO .....	76

10.2.	ESQUEMA VISUAL COMPARTIDO .....	77
10.3.	ASPECTOS VISUALES COMUNES.....	78
10.3.1.	Seguimiento de cambios.....	78
10.3.2.	Validación.....	80
10.3.3.	Habilitación de botones .....	80
10.3.4.	Notificar a través del <i>Status Bar</i> .....	82
10.4.	DISEÑO DE LA INTERFAZ DEL MÓDULO DE ATENCIONES .....	82
10.4.1.	Dashboard.....	83
10.4.2.	Cuadro de búsqueda.....	85
10.4.3.	Acciones de la barra de herramientas .....	86
10.4.4.	Panel de personas.....	87
10.4.5.	Panel de citas .....	91
10.4.6.	Panel de asistentes .....	92
10.4.7.	Panel de gráficas .....	93
10.4.8.	Preferencias .....	94
10.5.	DISEÑO DE LA INTERFAZ DEL MÓDULO DE SOCIOS.....	95
10.5.1.	Dashboard.....	96
10.5.2.	Panel de Socios .....	98
10.6.	DISEÑO DE LA INTERFAZ DEL MÓDULO DE COOPERACIÓN.....	102
10.6.1.	Dashboard.....	103
10.6.2.	Actividades.....	105
10.6.3.	Cooperantes.....	108
10.6.4.	Nueva actividad. ....	110
10.6.5.	Nuevo Cooperante.....	110
10.6.6.	Nueva Tareas. ....	111
10.6.7.	Foros de discusión .....	112
10.6.8.	Seguimiento. ....	112
10.6.9.	Incidencias. ....	113

10.6.10.	Calendario.....	114
10.7.	DISEÑO DE LA INTERFAZ DE LA EXTRANET .....	115
11.	TRABAJO FUTURO.....	117
11.1.	TRABAJO FUTURO GENERAL .....	117
11.2.	TRABAJO FUTURO PARA EL MÓDULO DE ATENCIONES.....	118
11.3.	TRABAJO FUTURO PARA EL MÓDULO DE SOCIOS .....	118
11.4.	TRABAJO FURUTO PARA EL MÓDULO DE COOPERACIÓN .....	118
11.5.	TRABAJO FUTURO PARA LA EXTRANET .....	119
12.	CONCLUSIÓN .....	120
13.	BIBLIOGRAFÍA .....	122
13.1.	LIBROS Y DOCUMENTOS .....	122
13.2.	SITIOS WEB CON RECURSOS TEXTUALES Y AUDIOVISUALES .....	122
13.3.	ARTÍCULOS.....	123
14.	ANEXO I: INSTRUCCIONES DE USO .....	124
14.1.	CÓMO USAR LA APLICACIÓN .....	124
14.2.	CÓMO USAR LA APLICACIÓN CON VISUAL STUDIO .....	124
14.2.1.	Preparación de la base de datos.....	125

---

# 1. INTRODUCCIÓN

---

## 1.1. SISTEMAS DE INFORMACIÓN

Los **sistemas de información** permiten aumentar la competitividad de las organizaciones, optimizando sus procesos y permitiéndoles diferenciarse de la competencia. En el contexto de las organizaciones sin ánimo de lucro suponen también más posibilidades para colaborar y crear sinergias con otras organizaciones.

Por otra parte, el estudio y análisis continuo característicos de un desarrollo iterativo –el utilizado en este proyecto– permite la racionalización de procesos, esto es, el repensar la forma, contenido y relaciones entre los distintos procesos y roles laborales de la organización.

Este proyecto ha propuesto **analizar la situación** de la organización sin ánimo de lucro “Colectivo Gamá LGTB” (en adelante Gamá) en cuanto a la **gestión de los socios**, el **servicio de atenciones**, y las **relaciones con otras organizaciones** o instituciones, y **desarrollar un sistema de información** que lo soporte.

## 1.2. ESTADO DEL ARTE

En el mercado podemos encontrar infinidad de programas, tanto gratuitos como de pago, que cubran por separado alguna de las necesidades de gestión del sistema de información de Gamá.

Para la gestión de socios, su información, sus cuotas y pagos, encontramos programas como “GestCli”, “Gym Control”, “Asociaciones XL” y muchos más que en un principio el soporte que prestarían estaría por encima de las necesidades de Gamá, al contar con paquetes de contabilidad y facturación.

Para la gestión de las atenciones los programas del mercado que mejor cubriría las necesidades de Gamá son los diseñados para la gestión de consultorios médicos como “PsicoClinic” o “DoctorGes”. Permiten llevar un control de los usuarios y las citas con reserva online y demás cosas, pero no permite llevar el control de las atenciones y el seguimiento que realiza Gamá, así como de generar informes personalizados según la atención prestada, que es de muy distinta índole.

En lo que concierne a la gestión de proyectos encontramos muchos programas en el mercado. La mayoría están indicados para gestionar proyectos de ingeniería como “GanttProject” o “Task Juggler”, que no están orientados al tipo de actividades que desarrolla Gamá. Gestores como “BaseCamp” o “Trello” están más orientados a esa gestión de actividades o tareas generales.

La siguiente tabla indica las competencias que cubrirían esos programas del mercado frente al alcance de nuestro proyecto:

Competencia	Actualidad	Proyecto	GestCli	DoctorGes	Trello
Gestión de Socios	X	✓	✓	X	X
Gestión de Usuarios	✓	✓	✓	✓	X
Gestión de Personal	X	✓	✓	X	✓
Gestión de pagos	X	✓	✓	X	X
Informes personalizados	✓	✓	X	X	X
Gestión de citas	X	✓	X	✓	X
Gestión de atenciones	✓	✓	X	X	X
Estadísticas	✓	✓	X	✓	X
Exportar	X	✓	X	✓	✓
Notificación de Eventos	X	✓	X	✓	✓
Gestión de proyectos	X	✓	X	X	✓



Podemos ver claramente como por separado ninguno de los programas indicados podrían dar un soporte integral para la gestión de la información de la organización debido a la gran disparidad de datos con los que trabajan. Además tienen un sistema de gestión de atenciones con el que se generan informes y seguimientos de los usuarios que solicitan atención de muy distinta naturaleza, desde una atención medica hasta un asesoramiento jurídico.

### 1.3. SITUACIÓN DE PARTIDA

Previo al comienzo del desarrollo, Gamá realizaba las actividades relacionadas a la gestión de socios y a la gestión del servicio de atenciones en formato físico. En cuanto a la generación de otros documentos sí utilizan el formato digital, en concreto Microsoft Word y Excel.

En relación a los **socios**, han de gestionar información personal, cómo llegaron a conocer a Gamá, etc., así como los **pagos de las cuotas**. Deben controlar pagos atrasados (o adelantados), que se ajusten a las cantidades correspondientes, etc., en lo que dan a llamar la *gestión de la morosidad*.

Gamá ofrece un **servicio de atenciones** individuales a personas LGTB así como a organizaciones que requieran información. Estas atenciones son de distinto tipo (psicológico, jurídico, de acogida, de orientación, de prevención para la salud,...). Según el caso se **derivarán** a un profesional, a la propia organización o a una institución, estando estas *derivaciones* en correspondencia con la atención solicitada. Las atenciones cuentan con un seguimiento por parte de Gamá independientemente de la derivación que tenga lugar. Las **fichas personales** de cada atención se ajustan a una categorías específicas (identidad sexual, orientación afectivo-sexual, edad, nacionalidad, nivel de estudios,...), y se integran al resto de datos para **extraer estadísticas** personalizadas (presentadas en forma gráfica). Esta información la utilizan para la elaboración de memorias y documentos varios, algunos de los cuales requieren para pedir y justificar subvenciones.

En cuanto a la **cooperación** con otras entidades o proyectos particulares, se requiere poder gestionar proyectos y su información relacionada. Esto es: plazos, actividades,

cooperantes que participan en cada actividad y eventos. Además, se contará con un sistema de mensajería asíncrona dentro del propio módulo a modo de foro.

Estos **tres módulos** de trabajo son independientes entre sí.

Por otra parte, a Gamá le interesa poder disponer de toda esa información no sólo desde la aplicación, sino en formato de documento, por lo que se hace necesario contar con funcionalidad de exportación. Por las necesidades de la organización, esto se concreta en la exportación de distintas informaciones (fichas, listados,...) a documento de texto de Word –que es el procesador de textos que utilizan en Gamá–.

Se desarrollará por tanto una aplicación de escritorio con tres módulos, a los que se accederá mediante un único formulario de acceso común a los tres.

Se ha desarrollado también un **sitio Web** que permite a Gamá **acceder a toda la información** de la base de datos **desde cualquier dispositivo en cualquier momento**. Puesto que la base de datos está encriptada, se hace necesario contar con una herramienta de este estilo para mostrar los datos desencriptados, cuestión que hace imposible tomar la alternativa de acceder directamente con herramientas como PhpMyAdmin.

En resumen, el proyecto pretende informatizar toda esta gestión, ajustándose a unas exigencias de calidad y a los requerimientos de la LOPD donde corresponda, puesto que se tratan datos sensibles.

## 1.4. ESTRUCTURA DE LA MEMORIA

En los primeros capítulos se abordan los objetivos, la metodología de desarrollo software empleada, los requisitos hardware y software, y el plan de trabajo. Los siguientes capítulos desarrollan el análisis de requisitos, el diseño, la implementación y las pruebas. Finalmente se exponen mejoras futuras planteadas que han quedado fuera del desarrollo, y las conclusiones.

Junto a esta memoria se entrega el documento PFC-4.

## 2. OBJETIVOS

---

En este capítulo se listan los objetivos planteados para el proyecto, tanto para la organización a la que se le realiza el software como para los autores el proyecto.

### 2.1. OBJETIVOS PARA LA ORGANIZACIÓN

- **Modernizar la organización** con la implantación de tecnologías de la información, aumentado la efectividad, eficiencia y la calidad de su actividad, así como poder realizarlo con mayor comodidad. Esto les permite también ahorrar costes a largo plazo y ser más respetuosos con el medio ambiente (por la reducción en el uso de papel y demás material).
- **Disponer de una infraestructura tecnológica** que permita, en el futuro, incorporar con mayor rapidez aquello que deseen sobre una base robusta.
- **Disponer de acceso remoto** a la información de la base de datos desde cualquier dispositivo en cualquier momento.
- **Mejorar la imagen corporativa** de la organización de cara a las instituciones de las que dependen para su financiación (subvenciones).

### 2.2. OBJETIVOS PARA LOS AUTORES DEL PROYECTO

- Aprender y **poner en práctica** tecnología y técnicas avanzadas en el desarrollo de software a medida, algunas de las cuales se demandan explícitamente en el mercado laboral, otras las demandan *los clientes*. Se enfatiza *la puesta en práctica* ya que algunas cuestiones se ven ligeramente en la carrera, y en muy pocas se llega a implementar con alguna tecnología concreta:
  - **C#**, Visual Studio

- **Pruebas:** TDD, diseño de casos de prueba, pruebas automatizadas, *mocking*, *fakers*.
- **Sistemas de control de versiones:** Git, flujo de trabajo con Git.
- **Patrones de diseño:** *Model-View-ViewModel* (MVVM en adelante), *binding*, *data context*; *command*, *observer*, *dependency-injection – service provider*, *repository*, integración de datos y más. Se busca construir una aplicación con componentes débilmente acoplados.
- **Diseño y gestión de bases de datos:** MySQL, MySQL Workbench, ORM.
- **Diseño de interfaces ricas de usuario:** WPF (*Windows Presentation Foundation*), XAML (*eXtensible Application Markup Language*, un XML específico para tecnología WPF).
- **Exportación de datos** a Word.
- **Visualización de gráficas:** Visualización en formato gráfico de distintos datos.
- **Extranet:** Sitio Web para acceso remoto a la información de la base de datos desde cualquier dispositivo en cualquier momento.
- **Gestión de la configuración del usuario**
- **Gestión de copias de seguridad**
- **Framework MVC de Microsoft ASP.NET MVC**, para la herramienta en la nube.
- Disponer de un ejemplo completo de un desarrollo real que estará disponible para la comunidad académica en el que estarán presentes estas tecnologías.
- Contribuir a la sociedad civil organizada sin buscar lucro. El desarrollo será completamente gratuito para Gamá.

---

## 3. METODOLOGÍA DE DESARROLLO SOFTWARE

---

Hemos escogido un modelo de desarrollo **iterativo, incremental y parcialmente evolutivo**. No nos ajustamos completamente a metodologías ágiles conocidas como Scrum o Kanban, sino que hemos tomado ideas y lo hemos adaptado a nuestra situación particular. Lo que se expone a continuación se refiere a la **fase de desarrollo** del proyecto, pues como se verá en el capítulo 5. *Plan de Trabajo*, ha habido más fases que esta. En cualquier caso, se han seguido los principios ágiles del archiconocido *Manifiesto por el Desarrollo Ágil de Software*:

Individuos e interacciones sobre procesos y herramientas  
Software funcionando sobre documentación extensiva  
Colaboración con el cliente sobre negociación contractual  
Respuesta ante el cambio sobre seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

Es **iterativo** porque se plantean iteraciones acotadas. Hemos decidido **separar las iteraciones por funcionalidad** –por conjunto de características– **y no por tiempo**. Se ha decidido así dada la falta de experiencia en desarrollos con esta tecnología, lo que hacía imposible estimar con fundamento. Siendo esto así, se ha considerado ineficiente destinar recursos a ello.

Es **incremental** porque en cada iteración se añade funcionalidad. También se mejoran, adaptan o cambian las ya existentes si así se acuerda con Gamá al finalizar cada iteración.

Es **parcialmente evolutivo** porque aunque la **arquitectura** se plantea en las fases iniciales del proyecto, en la práctica ha ido evolucionando para poder dar soporte a todo aquello que no se supo prever en las fases previas al desarrollo en sí.

Se describe a continuación la forma en que hemos abordado la documentación, gestión y flujo de trabajo.

### 3.1. DOCUMENTACIÓN

En cuanto a **documentación de la funcionalidad**, nos hemos limitado al uso de listas de características en forma no detallada. Se ha considerado ineficiente escribir casos de uso tanto en formato de diagrama como en formato de fichas. **Al tomar como premisa que los requisitos irán cambiando**, mantener actualizada esa documentación entorpecería el desarrollo al tiempo que no aportaría valor real, pues entre las entrevistas, la comunicación activa y los bocetos y apuntes en papel se ha podido enfocar el desarrollo en aquello que aporta un valor real.

Se le ha dado más importancia a la **documentación de aspectos arquitectónicos**, pues como se indicará en capítulos posteriores, se ha hecho énfasis en el diseño de la arquitectura como uno de los elementos cruciales para asegurar el éxito del proyecto. Por esto, se han creado diagramas de la estructura en ensamblados del proyecto, de las entidades de base de datos, de las clases principales de los distintos ensamblados, y del despliegue de los distintos componentes entre el servidor y los clientes.

Sobre la **documentación de casos de prueba**, siguiendo la filosofía de la documentación de la funcionalidad, se ha considerado que lo que aporta más valor sin mermar la eficiencia ha sido considerar la amplia batería de **pruebas unitarias** como la propia documentación de las pruebas. Se ha seguido un enfoque **TDD –Test Driven Development–** (hasta cierto punto). Las pruebas han formado parte integral del desarrollo, en lugar de estar relegadas a un segundo plano y realizar al final. Asimismo, estas han ido evolucionando de forma más o menos pareja al desarrollo de funcionalidad. Por todo ello, se considera que el enfoque adoptado **aporta las ventajas** derivadas de escribir pruebas al tiempo que **se evita la ineficiencia** de mantener actualizado una documentación aparte. Finalmente, dado que la complejidad del proyecto en cuanto a lógica de negocio es baja, se ha considerado que aporta poco valor crear diagramas de estos pequeños y simples procesos.

## 3.2. GESTIÓN

Las metodologías de desarrollo software no siempre cubren todos los aspectos del ciclo de vida del software. Por ejemplo, Scrum no te dice si debes hacer pruebas o no, cómo las debes hacer o cuánto debes documentar al respecto. De lo que sí se encarga Scrum, que es la metodología ágil que se ha planteado como base para el desarrollo de este proyecto, es de la gestión. La gestión se refiere a la gestión del **alcance**, a la **estimación** de esfuerzos, tiempos y costes, y a la **planificación**.

En cuanto al **alcance**, dado que no se cobra por el proyecto y que la acotación temporal viene dada sobre todo por requerimientos académicos, se establecieron unos **mínimos imprescindibles** en cuanto a funcionalidad. Estos mínimos incluyen: funciones **CRUD** de los tres módulos; gestión de **copias de seguridad**; generación de **gráficas** para el módulo de atenciones; creación del **sitio Web para acceso remoto** a los datos; **encriptación de los datos** en la base de datos por exigencias de proteger los datos sensibles con los que se trabaja.

Por otra parte, el **alcance** también viene dado por los **objetivos educativos** que se indicaron en el capítulo anterior. Estos objetivos, al traducirse en la práctica, abarcan de forma transversal las distintas actividades del desarrollo: diseño de la arquitectura, flujo de trabajo, desarrollo de pruebas, diseño de interfaces y usabilidad, despliegue de componentes, etc.

En cuanto a la **estimación** de esfuerzos, tiempos y costes hay que indicar que por las características del proyecto se hace aún más irreal de lo habitual la actividad de estimar. Por ello se ha optado, como se indicó más arriba, por un enfoque **iterativo en características** en lugar de uno **iterativo en tiempo**. De todas formas, se han realizado estimaciones para cada iteración siguiendo un **criterio de cantidad de horas** y subsiguientemente de **cantidad de días**, sólo que éstas no se han formalizado (*i.e.* no se han documentado).

En cuanto a **planificación** se ha optado por tanto por contar con un *Product Backlog* en forma de **lista de características**. Según el conjunto de estas que se decida incluir en cada iteración y las estimaciones para cada una, nos resultará una cantidad de días u otra. La planificación se detalla en el capítulo 5. *Plan de Trabajo*.

## 3.3. FLUJO DE TRABAJO

**Al comenzar la iteración**, lo primero que se hace es acordar **cómo se van a integrar** las nuevas características o modificaciones en **la arquitectura presente**. Dependiendo del caso no se requiere realizar modificaciones en la arquitectura, o se han de aplicar cambios leves, **o ésta evoluciona para dar soporte** a las nuevas necesidades.

Una vez se ha analizado el problema, se procede a la implementación de las características. Según la situación se opta por un enfoque más cercano o más alejado a TDD. Esta dinámica de **TDD se seguía sobretodo en las primeras iteraciones**, ya que permitía pensar más pausadamente sobre cada cambio además de consolidar las técnicas de pruebas pronto (para evitar el riesgo de acabar no haciendo pruebas).

**Más adelante** la dinámica pasó a ser una posición intermedia, en la que **a veces se hacía TDD** como al principio, **otras veces** se implementaban las pruebas a posteriori, **y en otras ocasiones** se implementaba un *boceto* de solución para probar si la solución tenía cabida. Si no tenía cabida, se probaba otra opción. Cuando ésta fuera apropiada, entonces se implementaba o se seguía TDD según preferencias personales.

En relación al diseño de interfaces de usuario, esto es, al diseño de vistas, respecto al desarrollo *back-end*, se siguió un orden arbitrario a la hora de implementarlos.

### 3.3.1. Sobre Git

Se usó Git desde la primera fase del proyecto, esto es, desde el estudio de la tecnología. Familiarizarse con Git y aprender a integrarlo en el flujo de trabajo ha sido un objetivo importante del proyecto. Durante el resto del desarrollo se ha seguido usando. El proyecto está alojado en GitHub: <https://github.com/PFC-acl-amg/GamaPFC>.

### 3.3.2. Revisiones de código

Las **revisiones de código**. Se hacían de manera informal, normalmente aprovechando momentos de menos energía. Resultaron ser beneficiosos, ya que servían para tener una comprensión mayor del creciente sistema, así como para detectar *code smells*, mejoras varias refactorizando de forma ligera o incluso para plantear cambios arquitectónicos, que los hubo.

---



## 3.4. EQUIPO DE TRABAJO

Esta metodología de trabajo requiere de un equipo de más de una persona. Por ello, y por las aspiraciones de alcance de la aplicación, se decidió formar el equipo con dos personas.

Ser dos personas ha permitido una organización y unos resultados más realistas, además de poder poner en práctica técnicas como la programación por parejas.

---

## 4. RECURSOS SOFTWARE Y HARDWARE

---

En este capítulo se describen las dependencias hardware y software que se requirieron para la realización del proyecto.

### 4.1. RECURSOS SOFTWARE

El proyecto se ha desarrollado usando la tecnología WPF de Microsoft para la aplicación cliente, ASP.NET MVC 5 para la extranet, y el lenguaje de programación C# 6 para ambos.

#### 4.1.1. Herramientas de desarrollo

HERRAMIENTA	DESCRIPCIÓN Y USO
<b>Microsoft Visual Studio 2015 Community Edition</b>	IDE ( <i>Integrated Development Environment</i> ) usado.
<b>phpMyAdmin</b>	Herramienta de gestión de bases de datos MySQL.
<b>MySQL</b>	DBMS (DataBase Management System) usado.
<b>IIS (Internet Information Services)</b>	Servidor web escogido para alojar la Extranet.
<b>MySQL Workbench</b>	Herramienta de gestión de bases de datos MySQL con múltiples utilidades. Entre ellas generar diagramas sofisticados.
<b>Git</b>	VCS ( <i>Version Control System</i> ) usado.

#### 4.1.2. Extensiones de Visual Studio:

HERRAMIENTA	DESCRIPCIÓN Y USO
<b>Git for Visual Studio</b>	Herramienta de Git integrada en Visual Studio. Facilita el flujo de trabajo con git.
<b>Inline Color Picker</b>	Permite seleccionar un color personalizado desde el editor de código XAML.
<b>Visual Studio Installer Project</b>	Permite crear instaladores simples con facilidad.
<b>Nuget Package Manager</b>	Gestor de paquetes/dependencias.

### 4.1.3. Librería principales

LIBRERÍA	DESCRIPCIÓN Y USO
<b>.NET Framework 4.6.1</b>	Framework de .NET.
<b>Prism</b>	Framework de MVVM de código libre, desarrollado hasta su versión 5 por Microsoft. Usamos la versión 6. Ofrece todo tipo de facilidades a la hora de implementar MVVM. Se han usado algunas de estas funciones.
<b>Unity</b>	Contenedor de gestión de dependencia para implementar el patrón <i>Dependency Injection / Inversion Of Control</i> .
<b>MahApps.Metro</b>	Conjunto de estilos, iconos, recursos y utilidades para aplicaciones basadas en XAML. Se ha tomado como base sobre la que personalizar el aspecto visual de la aplicación. Se han usado algunas de estas utilidades.
<b>NHibernate</b>	ORM basado en el famoso ORM de Java <i>Hibernate</i> .
<b>FluentNHibernate</b>	Extensión de NHibernate que basa su configuración en código en forma <i>Fluent</i> , en lugar de
<b>MySql.Data</b>	Conector de MySQL para .NET (C# en este caso)
<b>xUnit</b>	Utilidad para facilitar el desarrollo de pruebas unitarias.
<b>Faker</b>	Utilidad para generar datos falsos. Facilita el popular la base de datos de desarrollo con nombres, fechas, direcciones, etc..
<b>DocX</b>	Permite exportar a formato docx sin pasar por las librerías de InterOP de Microsoft.
<b>MySqlBackup</b>	Facilita la creación de copias de seguridad y la restauración de las mismas.
<b>Modern UI Charts</b>	Librería usada para la generación de gráficos.

## 4.2. RECURSOS HARDWARE

RECURSO	DESCRIPCIÓN
<b>Servidor dedicado</b>	Ordenador propiedad de Gamá donde alojar el servidor de sincronización y la base de datos MySQL. Se contó

	con uno específico para el desarrollo también.
<b>Portátil personal</b>	Ordenadores personales de los autores del proyecto.

---

## 5. PLAN DE TRABAJO

---

El desarrollo del proyecto se ha dividido en **cinco fases: estudio de la tecnología, toma de requisitos, diseño de la arquitectura base, proceso iterativo de desarrollo, finalización**. En este capítulo se describe brevemente en qué consistió cada fase, señalando las particularidades dadas por nuestro caso y justificando algunas decisiones tomadas.

Las tres primeras se solaparon en el tiempo de una forma u otra. La fase de finalización atiende a las exigencias académicas, pues de no existir éstas se habría continuado con el proceso iterativo habitual de desarrollo mientras así se acordara. En el enfoque ágil que se ha planteado no hay tal cosa como *iteración final*. Cuestiones como la elaboración de esta memoria y las modificaciones finales (de cara a la entrega académica) han hecho que estas últimas modificaciones no cuenten con todo el respaldo de pruebas unitarias convenientemente actualizadas (pruebas que hay que ir actualizando), sustituyéndose estas pruebas automatizadas por pruebas manuales. Pruebas manuales que, por otra parte, siempre hay que llevar a cabo, ya que todo lo concerniente a la interfaz gráfica (*layout, estilos, bindings*) no es objeto de trabajo de las pruebas unitarias.

### 5.1. ESTUDIO DE LA TECNOLOGÍA

Esta fase comenzó antes de iniciarse formalmente el proyecto. En ella se siguieron distintos recursos educativos (véase la bibliografía). Lo primero fue familiarizarse con el código XAML y las formas más sencillas de implementar MVVM. Se siguió aumentando la complejidad a medida que se iba profundizando en el aprendizaje. Con esto se pretendió conocer la potencia, limitaciones y riesgos de esta tecnología.

En cuanto a **potencia o alcance** cumple con las expectativas de desarrollo de una interfaz de usuario *sofisticada (rich user interface)*.

La limitación principal que encontramos en WPF es que puede volverse pesado (*i.e.* lento para transitar entre vistas, para cargarlas, y para refrescarlas) con facilidad.

Esta limitación crea el riesgo de tener que reformular el diseño gráfico e incluso la arquitectura para hacer frente a las ineficiencias.

Hay que señalar que es raro que un proyecto software no requiera de profundizar en algún aspecto o aprender algo nuevo independientemente de la experiencia previa. En esta fase nos referimos al estudio inicial, que como decimos tuvo sus inicios bastante antes de la aprobación del proyecto. Desde la fase iterativa hasta la finalización del proyecto se ha continuado aprendiendo, pero no consideramos parte de esta fase todo ese aprendizaje habitual a cualquier desarrollo software.

## 5.2. ANÁLISIS INICIAL DE REQUISITOS

Una vez se toma conciencia del alcance, limitaciones y riesgos de WPF se puede comenzar la actividad de análisis sabiendo asesorar al cliente. Esta fase consistió en la realización de varias entrevistas donde se plantearon los actores, lista de características inicial, y aspectos de calidad y usabilidad.

Como buena parte de lo desarrollado ya era llevado a cabo por Gamá en formato físico, las necesidades de información estuvieron a nuestra disposición desde un primer momento.

Se enfocó la actividad de obtención de requisitos en facilitar el posterior diseño de una arquitectura que diera soporte al conjunto de funcionalidades. Es decir, se llevó a cabo con vistas a reducir riesgos futuros. Los cambios arquitectónicos pueden tener un impacto impredecible y en ocasiones catastrófico para el conjunto del software y el desarrollo.

### 5.3. DISEÑO DE LA ARQUITECTURA

Esta fase tiene bastante de *estudio de la tecnología*, aunque enfocado en los aspectos arquitectónicos, guiados estos por algunos de los objetivos del proyecto. Para ello, se profundizó en MVVM y en el diseño de aplicaciones débilmente acopladas. Todo esto se traduce en el aprendizaje, diseño y desarrollo tanto de MVVM como de distintos patrones de diseño, entre ellos: *Command Pattern, Composite View, Dependency Injectio, Invesion of Control, Event Aggregator, Observer, Registry, Repository*.

El diseño de la arquitectura depende del resultado obtenido en el análisis en relación a los actores y a las características (funcionales y de calidad).

### 5.4. DESARROLLO ITERATIVO

Una vez planteada la arquitectura, y después de un tiempo de formación continua en las tecnologías a usar, comienza el proceso iterativo que se describió en *Capítulo 3. Metodología de Desarrollo Software*.

Este proceso consiste en

1. Tomar un subconjunto de características de la lista de características para ser desarrolladas en la iteración.
2. Detallar los requisitos seleccionados y hacer prototipos (en papel o directamente en código según conveniencia, como por ejemplo si se está en una entrevista con el cliente o no). Estos se aceptan o modifican hasta considerarse satisfactorios.
3. Implementar y probar las características seleccionadas. La comunicación con el cliente es activa cuando se requiere.
4. Entregar y mostrar la nueva versión al cliente. Se toma *feedback*.
5. Retrospectiva del equipo de desarrollo, vuelta al paso 1.

## 5.5. FINALIZACIÓN

Para terminar el proyecto se hizo necesario acotar el alcance del proyecto de forma temporal. Esto es, establecer un límite de cara a la entrega académica. Es en esta fase donde se desarrolla la memoria.

Debido a ello, se han dado aún más prioridad a los principios ágiles que han regido el desarrollo, en particular el principio de “*Software funcionando frente a documentación extensiva*”. Esto ha tenido impacto en las pruebas unitarias. Siendo las pruebas unitarias una forma de documentación activa (nos dicen, en *tiempo real*, que un cierto porcentaje del código no contiene fallos) consideramos coherente haberlas descuidado en esta última fase, máxime cuando se han realizado extensas pruebas manuales (que por otra parte son necesarias igualmente, ya que las pruebas unitarias no cubren todas las capas, como la visual o la de acceso real a la base de datos –*pruebas funcionales*–).

Para finalizar, es importante señalar que el software desarrollado no es un producto final, sino que se seguirá desarrollando. Por ello, es probable que entre el día de presentación de esta memoria junto con el código, hasta el día de la defensa ante un tribunal, se hayan introducido cambios en los programas. Hay que tener en cuenta que de cara a la presentación, con el cambio de prioridades que ello implica, algunos aspectos se han dejado un poco de lado, como la batería de pruebas unitarias. Se siguen manteniendo las del código de infraestructura pero muchas de los *ViewModels* están desactualizadas.

La primera *iteración* tras consistirá precisamente en recuperar toda esa batería actualizada de pruebas y volver a sentar las bases para reiniciar la fase iterativa.



---

## 6. ESTIMACIÓN DE ESFUERZO Y PRESUPUESTO ESTIMADO

---

Una de las características del proyecto es que **el desarrollo ha sido gratuito**. Con ello pretendíamos eliminar algunos riesgos inherentes a nuestro contexto (falta de experiencia en la tecnología, etc.) al tiempo de contar con más flexibilidad, tanto para los desarrolladores como para el cliente. De todas formas, se expone a continuación un análisis presupuestario que cubriría las fases dos, tres y cuatro: análisis de requisitos, diseño de la arquitectura, desarrollo iterativo.

### 6.1. ESTIMACIÓN DE ESFUERZO

Con un esfuerzo medio dedicado de **cuatro horas diarias** (entre los dos desarrolladores) durante un periodo aproximado de **dos años** (tomando **cuarenta y cuatro** semanas en cada uno) a razón de **cinco días por semana**, y teniendo en cuenta los porcentajes estimados dedicados a cada actividad, nos resulta la siguiente estimación:

$$4 \times 2 \times 44 \times 5 = 1.760 \text{ horas}$$

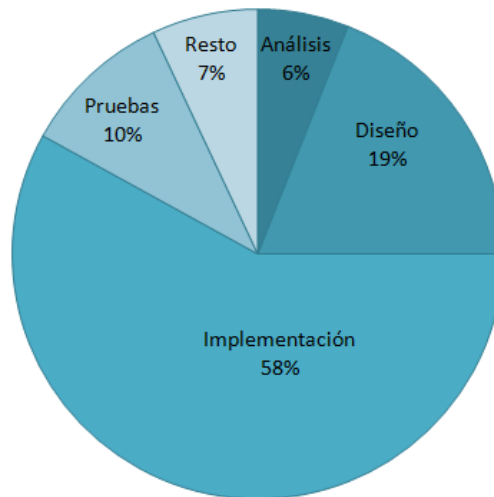


Ilustración 1. Estimación de costes

Donde:

Actividad	Descripción	Horas	Porcentaje
Análisis	Obtención y análisis de requisitos.	106	6%
Diseño	Diseño arquitectónico, de la interfaz de usuario y de despliegue.	335	19%
Implementación	Codificar <i>front-end</i> y <i>back-end</i> .	1020	58%
Pruebas	Codificar las pruebas	176	10%
Resto	Resto de actividades: gestión, comunicación, control de versiones, etc.	123	7%
Total		1.760	100%

## 6.2. ORGANIZACIÓN DEL ESFUERZO

En términos generales, ambos alumnos se encargaron tanto de las actividades de análisis como las de diseño arquitectónico, diseño del *layout* general, y despliegue. El trabajo se dividió a partir de ahí con la implementación, las pruebas, el diseño de la interfaz de usuario de los tres módulos más la extranet.

El alumno Alberto Cardona López se encargó del módulo de atenciones y de la extranet, así como del servidor de sincronización.

El alumno José Antonio Martín García se encargó del módulo de socios y del módulo de cooperación.

## 6.3. PRESUPUESTO ESTIMADO

El presupuesto se ha estimado en base al número de horas totales invertidas. Se trata de un cálculo bastante arbitrario dada la falta de experiencia en el desarrollo con las tecnologías seleccionadas así como de datos reales de sueldos y de productividad con esta tecnología en el mercado real.

### 6.3.1. Supuesto 1: Presupuesto Junior

Si asumimos unos costes laborales de un **desarrollador junior sin experiencia**, recién entrado al mercado laboral, de **700 € al mes**, se tiene que el coste laboral por hora es de  $(700 \text{ €} / 4 \text{ semanas} / 40 \text{ horas semanales}) = 4,4 \text{ € por hora}$ .

Esto resultaría en un coste total de  $(1.760 \text{ horas a } 4,4\text{€ la hora}) = 7.744,00 \text{ €}$

### 6.3.2. Supuesto 2: Presupuesto Senior

Si asumimos unos costes laborales de un **desarrollador senior** (más de dos años de experiencia) de **2.800 € al mes**, se tiene que el coste laboral por hora es de  $(2.800 \text{ €} / 4 \text{ semanas} / 40 \text{ horas semanales}) = 17,5 \text{ € por hora}$ .

Si asumimos una productividad tal que le permite desarrollar el proyecto en **un tercio del tiempo** (se pasaría de **1.760** horas a **587** horas), se obtendría un **coste total de**  $(587 \text{ horas a } 17,5 \text{ € la hora}) = 10.272,50 \text{ €}$ .

---

## 7. ESTUDIO DEL DOMINIO DEL PROBLEMA

---

En este capítulo se describen los actores que actúan con el sistema y las entidades del dominio del problema. También se incluye aquí el diagrama entidad-relación de la base de datos, que coincide con lo que sería un diagrama entidad-relación de análisis.

### 7.1. ACTORES

Los actores del sistema se han clasificado según el módulo que utilicen y las funciones que realizan dentro del mismo.

Actor	Descripción
Contable	Contable de la asociación Gamá.
Asistente	Persona que trabaja en Gamá entre cuyas funciones está el atender a personas.
Personal	Todo el personal de Gamá.

Tabla 1. Descripción de actores

Actor	Módulo que utiliza	Descripción de uso
Contable	Gestión de Socios	Consulta datos, imprime listados contables y realiza las acciones que correspondan sobre los datos contables introducidos en la aplicación, como la cuestión de la morosidad.

Personal	Gestión de Socios	Gestiona la información de los socios, realiza las acciones contables que correspondan.
Personal	Servicio de Atenciones	Consulta datos, imprime listados o fichas personales, añade nuevas citas al atender a una persona, etc. Utiliza la información para realizar informes y memorias aparte del sistema.
Asistente	Servicio de Atenciones	Consulta información de las citas, rellena atenciones durante o después de tener lugar la cita.
Personal	Extranet	Consulta información sobre personas, citas y socios en cualquier momento en cualquier lugar.
Personal	Cooperación	Introduce nuevas actividades y las gestiona. Consulta y participa en los foros. Consulta y gestiona información varia sobre tareas y eventos asociados a las actividades. Imprime listados varios o fichas.
Cooperantes	Cooperación	Consulta información, gestiona actividades, participa en los foros.

Tabla 2. Relación entre actores y uso de los módulos

## 7.2. ENTIDADES

### 7.2.1. Servicio de atenciones

Entidad	Descripción
Persona	Persona que acude a Gamá para recibir asesoramiento sobre una cuestión LGTB.
Asistente	Persona que trabaja en Gamá entre cuyas funciones está asesorar a personas.
Cita	Para que el <b>asistente</b> pueda asesorar a una <b>persona</b> , primero se ha de concertar una <b>cita</b> . Esta se puede concertar contactando con Gamá por distintas vías:

	telefónica, por email, y personarse en su sede.
<b>Atención</b>	Informe asociado a una cita. Cuando la cita tiene lugar, se escribe este informe, que consta de información sobre el tipo de atención solicitada (social, psicológica, jurídica, de acogida, de orientación laboral, de prevención para la salud, de formación, de participación en Gamá, de otro tipo –se indica cual–), sobre la <b>derivación</b> asociada, y un seguimiento.
<b>Derivación</b>	Asociada a cada atención, se trata de información sobre qué opciones de terceros se proponen a la <b>persona</b> al tener lugar la <b>cita</b> en caso de que no sea Gamá quién se encargue del caso. Asimismo, se incluye información sobre cuáles de estas opciones son llevadas a cabo por la <b>persona</b> . Las derivaciones, al igual que las atenciones, cuentan con una categorización cerrada, a saber: social, jurídica, psicológica, de formación, de orientación laboral, otro tipo –se indica cual–.

### 7.2.2. Gestión de socios

Entidad	Descripción
<b>Socio</b>	Socios de la asociación Gamá
<b>Periodo de alta</b>	Intervalo de tiempo en el que un <b>socio</b> está dado de alta en la asociación. Un <b>socio</b> puede tener estados de alta intermitentes, por lo que podrá tener más de uno asociado.
<b>Cuota</b>	Las <b>cuotas</b> que pagan los <b>socios</b> mensualmente.
<b>Moroso</b>	<b>Socio</b> con más de un cierto número de meses cuyas <b>cuotas</b> no ha pagado.

### 7.2.3. Cooperación

Entidad	Descripción
Actividad	Engloba a trabajos, proyectos, talleres,... que realiza la asociación.
Cooperante	Persona que trabaja en Gamá encargada de realizar las Actividades de la asociación.
Tarea	Cada uno de los trabajos en los que subdivide una actividad. Tendrán un responsable asignado a la tarea y la Actividad s dará por terminada al finalizar todas las tareas.
Foro	Cada Actividad se dota de entorno donde los Cooperantes encargados de la misma podrán intercambiar opiniones, ideas, ocurrencias,..., sobre la Actividad y las Tareas en que se subdivide.
Evento	Es el encargado de llevar un registro de actuación en el módulo de cooperación. Recogerá cada suceso acaecido que implique alguna acción que modifique la Base de Datos.
Seguimiento	Cada Tarea de una actividad dispone de un entorno donde los cooperantes asignados a la Actividad y a la realización de dicha Tarea, pueden anotar cada una de las actuaciones que han llevado a cabo para completar las tareas macadas. El seguimiento de cada Tarea llevará entonces un historial de cada uno de los trabajos realizados.
Incidencia	Con el mismo principio de Seguimiento, las incidencias están asociadas a cada Tarea. Funciona a modo de historial para almacenar los problemas y contratiempos encontrados por los Cooperantes al realizar la Tarea.
Mensaje	Mensajes que se envían a cada uno de los foros creados.

---

## 8. ANÁLISIS DE REQUISITOS

---

El análisis se ha dividido en las siguientes secciones:

- Lista de características funcionales y de calidad comunes entre los módulos
- Lista de características del módulo de atenciones
- Lista de características del módulo de socios
- Lista de características del módulo de cooperación
- Lista de característica de la extranet

### 8.1. CARACTERÍSTICAS COMUNES ENTRE LOS MÓDULOS

Característica	Descripción
Seguimiento de cambios	<p>Seguimiento de los cambios que el usuario introduce en los modelos con los que trabaja. Permite:</p> <ul style="list-style-type: none"><li>▪ Mostrar en la interfaz qué campos han sido modificados – cambiando el color de fondo del campo– y su valor original –a través del <i>tooltip</i>–</li><li>▪ Revertir las modificaciones introducidas, devolviendo el modelo tratado a su estado original, antes de introducir cambios.</li></ul>
Validación	<p>Validación en el lado <i>cliente</i>. Cada vez que se introduce un cambio, se somete el modelo a una validación.</p> <ul style="list-style-type: none"><li>▪ Se muestran en la interfaz qué campos son inválidos, cambiando el estilo del campo y mostrando el mensaje de error específico en cada uno.</li></ul>
Habilitación de botones	<p>Se utiliza la información que nos proporcionan las dos características anteriores (seguimiento de cambios y validación) para habilitar o deshabilitar botones cuyas acciones impliquen persistir datos. El modelo deberá haber sido cambiado y además ser válido para que se</p>



	<p>permita llevar a cabo la acción. La primera condición atiende a la usabilidad. La segunda a la consistencia de los datos y robustez de la aplicación (al imposibilitar un conjunto de excepciones en tiempo de ejecución). El estado del botón (habilitado o deshabilitado) hace que se muestre con un estilo visual u otro.</p>
<p><b>Preferencias</b></p>	<p>Preferencias de usuario sobre varios aspectos de usabilidad, así como para indicar si deben realizarse copias de seguridad automáticas o no y en qué carpeta guardarlas.</p>
<p><b>Gestión de copias de seguridad</b></p>	<p>Todos los módulos cuentan con funcionalidad de generación automática de copias de seguridad. También se permite realizar una en cualquier momento.</p>
<p><b>Cambio de módulo</b></p>	<p>Pequeña facilidad para cambiar entre módulos desde la propia aplicación. La acción consiste en cerrar el módulo actual y relanzar el selector de módulo. Por otra parte, se permite tener varias instancias de uno o varios módulos al mismo tiempo en un mismo terminal.</p>
<p><b>Sincronización de datos</b></p>	<p>Los clientes lanzados notificarán, a través del servidor central de sincronización, las modificaciones que realicen, instando al resto de cliente a actualizar sus datos en tiempo real. Este requisito implica que se debe poder visualizar el estado de conexión al servidor.</p>
<p><b>Control de acceso</b></p>	<p>Cada módulo cuenta con sus propios usuarios para el control de acceso. Dentro de cada módulo, sus usuarios gozan de los mismos privilegios.</p>
<p><b>Exportación de datos</b></p>	<p>Exportar fichas y listados a docx.</p>
<p><b>Generación de gráficas</b></p>	<p>Donde se ha considerado necesario o deseable se han introducido gráficas mostrando datos estadísticos varios.</p>
<p><b>Navegación</b></p>	<p>Facilidad para navegar por las distintas vistas. Se traduce en que se presentan casos en lo que desde distintos sitios se pueden realizar las mismas acciones, así como navegar cómodamente entre vistas. Esto se consigue con dos tipos de acciones: acciones <i>Ir a</i> y acciones <i>Editar</i>. La primera permite navegar hacia el modelo relacionado seleccionado (e.g. a la persona asociada a una cierta cita). La segunda permite editar algunos modelos sin tener que navegar a una vista específica para ello, sino abriéndose un diálogo en esa misma vista.</p>

**Notificación de cambios**

La barra de estado se ilumina y muestra un mensaje durante unos segundos cada vez que se ha realiza una acción que implica persistir datos en la base de datos.

**Propagación de cambios**

Al realizarse un cambio en el modelo a través de una vista, los cambios se propagan a aquellas vistas donde corresponda.

## 8.2. CARACTERÍSTICAS DEL MÓDULO DE ATENCIONES

Característica	Descripción	Detalle
Añadir persona	Añade a una persona nueva a la base de datos.	La acción debe ser accesible desde la barra de herramientas ( <i>Toolbar</i> ).
Editar persona	Modifica los datos personales de una persona existente.	Esta edición ha de compartir vista con la edición de citas y atenciones de la persona. Este requisito incluye la visualización de los datos de la persona así como de sus citas (con su calendario personal) y las atenciones asociadas a las citas.
Eliminar persona	Elimina a la persona y todos sus registros asociados de la base de datos.	Esto significa que se eliminan todas sus citas y atenciones asociadas, por eso esta lista carece de funciones de eliminar citas y atenciones.
Añadir asistente	Añade un nuevo asistente a la base de datos.	La acción debe ser accesible desde la barra de herramientas ( <i>Toolbar</i> ).
Editar asistente	Modifica los datos personales de un asistente existente.	Esta edición comparte la vista con el listado del resto de asistentes así como de las citas en las que el asistente en edición está presente.
Eliminar asistente	<i>No implementado.</i>	<i>Una cita requiere que se establezca un asistente. Por ello, aunque un asistente deje de formar parte de la organización o de colaborar con ella, es necesario que permanezca en la base de datos.</i>
Añadir cita	Añade una cita en la base de datos.	Las citas se podrán añadir desde el calendario general, y desde el calendario específico de cada persona. La creación requiere que se indique la persona a la que afecta y el asistente que asistirá en la cita cuando esta se efectúe. Si el asistente ya

		<p>tiene una cita cercana (entre una hora menos y una hora más), se indica al usuario que solaparían y se impide la acción.</p> <p>Al crearse la cita se crea automáticamente una atención asociada nueva y vacía, por eso no existe la función explícita de <i>añadir atención</i>.</p>
<b>Editar cita</b>	Modifica los datos de una cita existente.	La acción debe poder realizarse tanto desde la vista de citas general, como las particulares de cada persona, como en el registro de citas asociadas a cada asistente.
<b>Editar atención</b>	Modifica la información de una atención.	Cuando se crea una cita, esta ya contiene una atención, pues es lo que se rellena al tener lugar la cita. Por ello, sólo existe la función para el usuario de <i>editar atención</i> , y no las de crear o eliminar.
<b>Buscar persona</b>	Buscar persona por nombre.	Se podrá buscar a una persona por nombre, listando los resultados que encajen, y pudiendo navegar a la persona que se seleccione. El formulario de búsqueda debe ser accesible y permanecer visible en todo momento.
<b>Visualizar personas</b>	Visualizar de forma paginada al conjunto de personas.	Disponer de un panel donde se vean todas las personas <i>listadas</i> en forma de matriz mostrando su imagen, nombre y NIF.
<b>Visualizar atenciones</b>	Visualizar el conjunto de atenciones	Visualizar un listado de todas las atenciones del sistema
<b>Calendario de citas</b>	Visualización de todas las citas del sistema en formato de calendario y formato tabular.	Se trata de un calendario mensual navegable por semanas desde el que se deben poder crear citas y visualizar todas las citas del sistema. El formato tabular, al que se accederá como opción de cambio de vista,

		mostrará todas las citas en una tabla y permitirá filtrar por fecha.
Visualizar asistentes	Visualizar al conjunto de los asistentes	Estará conformado por un listado de asistentes junto a una sección dinámica de contenido donde se mostrará la información y citas asociadas al asistente seleccionado.
Visualizar gráficas	Visualizar gráficos de ciertos datos sobre personas y atenciones	Se deberán poder visualizar gráficas de estilo pastel para los campos de edad, identidad sexual, orientación sexual, estado civil, y tipo de atención solicitada. Las derivaciones propuestas y realizadas se mostrarán integradas en un gráfico de barras.
Exportar	Exportar fichas y listados	Dependiendo de la pantalla donde se esté situado, se exportará un listado de todas las personas, o la ficha de la persona en pantalla junto a todas sus citas.

### 8.3. CARACTERÍSTICAS DEL MÓDULO DE SOCIOS

Característica	Descripción	Detalle
Añadir socio	Añade a un socio nuevo a la base de datos.	La acción debe ser accesible desde la barra de herramientas ( <i>Toolbar</i> ).
Editar socio	Modifica los datos personales de un socio existente.	Esta edición ha de compartir vista con la edición de periodos de alta y cuotas del socio. Este requisito incluye la visualización de los datos del socio así como de sus periodos de alta y e información sobre cuotas.
Eliminar socio	No implementado	<i>Un socio puede estar dado de alta o de baja, pero en tanto que información contable que puede tener validez incluso mucho tiempo después de que un socio se dé de baja, sólo</i>

		<i>se podrá eliminar un socio accediendo directamente a la base de datos como administrador.</i>
<b>Añadir periodo de alta</b>	Asigna un nuevo periodo de alta a un socio existente	Al añadirse un periodo de alta, éste se crea con la fecha actual y sin fecha de fin.
<b>Editar periodo de alta</b>	Modifica los datos personales de un asistente existente	Cuando se edita la fecha de inicio o de fin del intervalo, se deben generar tantas cuotas vacías como nuevos meses haya, manteniéndose la información de cuotas preexistentes. Si el periodo de alta se acorta, las cuotas que queden fuera del intervalo sí serán eliminadas.
<b>Añadir cuota</b>	Añade una cuota a un periodo de alta	Las cuotas se autogeneran al editarse los periodos de alta.
<b>Editar cuota</b>	Modifica los datos de una cuota existente.	
<b>Eliminar cuota</b>	Elimina una cuota del sistema.	Esto tiene lugar de forma indirecta al editarse un periodo de alta. Véase este requisito para más información.
<b>Buscar socio</b>	Buscar socio por nombre.	Se podrá buscar a un socio por nombre, listando los resultados que encajen, y pudiendo navegar al socio que se seleccione. El formulario de búsqueda debe ser accesible y permanecer visible en todo momento.
<b>Visualizar socios</b>	Visualizar de forma paginada al conjunto de socios.	Disponer de un panel donde se vean todas los socios <i>listadas</i> en forma de matriz mostrando su imagen, nombre y NIF.

## 8.4. CARACTERÍSTICAS DEL MÓDULO DE COOPERACIÓN

Característica	Descripción	Detalle
Añadir Actividad	Añade una nueva Actividad a la base de datos.	La acción está accesible desde la barra de herramientas ( <i>Toolbar</i> ).
Visualizar Actividades	Visualizar de forma paginada al conjunto de actividades.	Disponer de un panel donde se vean todas las actividades <i>listadas</i> en forma de matriz mostrando su título y descripción.
Añadir Cooperante	Añade a un nuevo Cooperante a la base de datos.	La acción está accesible desde la barra de herramientas ( <i>Toolbar</i> ).
Editar Actividad	Modifica los datos de una Actividad existente en la base de datos.	La edición de una Actividad se permite desde varias vistas. Está disponible en la vista principal ( <i>Dashboard</i> ), así como en la vista donde se visualizan los datos de los Cooperantes, donde se indican en las Actividades en que participan. Los cambios en los Datos de una Actividad se verán reflejados en todas las vistas que incluyan dicha información.
Borrar Actividad	Borra una actividad de la base de datos. Con ésta se borran las tareas y los foros asociados a dicha actividad.	Borrar una Actividad se permite desde varias vistas. Está disponible en la vista principal ( <i>Dashboard</i> ), así como en la vista donde se visualizan los datos de los Cooperantes, donde se indican en las Actividades en que participan. Los cambios en los Datos de una Actividad se verán reflejados en todas las vistas que incluyan dicha información.
Ir Actividad	Permite cargar los datos de una actividad.	Al ir a una Actividad tendremos disponible en la interfaz sus datos de creación y el entorno necesario para la creación y modificación de las tareas y foros de la actividad. Esta acción se permite desde la vista de

		gestión de los cooperantes, donde vemos las Actividades en que trabajan y desde el gestor de eventos, que con la información del evento nos permite ir a la actividad donde se generó dicho evento
<b>Filtrar Actividades</b>	Permite filtrar la lista de actividades.	Las opciones para filtrar una actividad están en función de dos parámetros. El primero según el estado de la actividad y el segundo por meses, indicando las actividades que finalizan en los meses seleccionados.
<b>Filtrar Eventos</b>	Permite filtrar los eventos por fechas.	Permite filtrar todos los eventos desde una fecha o filtrar los eventos en un intervalo de tiempo concreto.
<b>Editar Cooperante</b>	Modifica los datos personales de un cooperante.	Sólo disponible desde el entorno de visualización del listado de cooperantes con que cuenta la asociación.
<b>Exportar Cooperante</b>	Exporta los datos del cooperante seleccionado.	Permite pasar todos los datos personales del cooperante así como las actividades en que participa a formato Word.
<b>Añadir Tarea</b>	Añade una tarea a la actividad seleccionada.	Acción disponible en el desarrollo de la actividad, permite crear las diferentes tareas en las que se subdivide una actividad.
<b>Tareas Finalizadas</b>	Visualizar una lista con las tareas finalizadas.	Permite ver cada una de las tareas con toda la información asociada que ya se han finalizado.
<b>Finalizar Tarea</b>	Permite dar por concluida una tarea.	Al dar por terminada una tarea, se modificará en la base de datos su estado a finalizado.
<b>Editar Tarea</b>	Modifica los datos de una tarea.	Nos permite modificar los datos asociados a una tarea como son su descripción, el responsable de llevar a cabo la tarea o la fecha de finalización.
<b>Borrar Tarea</b>	Borra una tarea de la base de datos	Borra la tarea seleccionada junto con todo el historial de desarrollo de la tarea.



<b>Añadir Incidencia</b>	Añade Incidencias en la actividad seleccionada	Acción disponible en el desarrollo de la actividad, permite crear un historial de incidencias donde se indica cada uno de los problemas surgidos durante el desarrollo de la actividad, informando al resto de cooperantes de las complicaciones surgidas.
<b>Ir Tarea</b>	Carga los trabajos realizados en la tarea.	Permite cargar todos los trabajos realizados necesarios para el desarrollo de la tarea seleccionada.
<b>Añadir Foro</b>	Añade un Foro de discusión a la actividad seleccionada	Acción disponible en el desarrollo de la actividad, permite crear diferentes Foros de discusión donde los Cooperantes pueden intercambiar opiniones sobre la Actividad o alguna de sus Tareas.
<b>Borrar Foro</b>	Borrar un foro de discusión de la actividad seleccionada.	Borra el Foro seleccionado junto con todos los mensajes que se han enviado al foro.
<b>Ir Foro</b>	Permite cargar los mensajes del foro.	Permite ir al entorno de gestión del foro. Estarán disponibles todos los mensajes enviados al foro de discusión así como la posibilidad de enviar nuevos mensajes.
<b>Añadir trabajo Tarea</b>	Inserta un trabajo en el historial de la Tarea	Permite añadir un nuevo trabajo el historial de desarrollo de la tarea seleccionada.
<b>Añadir Incidencia</b>	Inserta una nueva incidencia en el historial de la tarea.	Permite añadir al historial de problemas encontrados durante el desarrollo de la tarea nuevas incidencias.

## 8.5. CARACTERÍSTICAS DE LA EXTRANET

Características	Descripción	Detalle
-----------------	-------------	---------

<b>Visualizar personas</b>	Listado de personas	Se muestra un listado de personas con todos sus campos menos la imagen para aumentar la eficiencia. Hay que entender que cuando se consulte aquí información la imagen no es relevante. Sí es en cambio importante poder agilizar el proceso.
<b>Visual citas</b>	Listado de citas	Listado tabular de todas las citas desde el día actual. El uso que se le da a esta consulta hace que sea innecesario consultar fechas anteriores.
<b>Visual socios</b>	Listado de socios	Listado de todos los socios. Al igual que con las personas, se excluye la imagen.

---

## 9. DISEÑO ARQUITECTÓNICO

---

La arquitectura de un sistema informático es de crucial importancia. De ella depende la dificultad para ir añadiendo funcionalidad y requisitos de calidad. Debe ser robusta como para soportar los requisitos del análisis inicial, pero suficientemente flexible como para que pueda evolucionar para dar soporte a las nuevas necesidades sin alterar significativamente al resto de código existente. En el caso de este proyecto, esto significa tomar como referencia los **principios SOLID** de diseño de software, lo cual se traduce en plantear una arquitectura de componentes débilmente acoplados que, mediante el uso de patrones de composición (tanto para componentes visuales como para el resto de componentes) se integren y den lugar al resultado completo, al tiempo que nos permita desarrollar y probar los componentes de forma individual.

Concretando, se ha adoptado el patrón de diseño arquitectónico MVVM, que se explicará más adelante en este capítulo, en connivencia a un conjunto de patrones de diseño comúnmente asociados a MVVM.

También debemos exigir a la arquitectura que permita a los desarrolladores **trabajar de forma independiente** en distintos aspectos. Para ello se ha seguido la estructura y relación entre ensamblados que se indica más abajo. De esta manera es posible desarrollar, probar y lanzar algún módulo (atenciones, socios o cooperación) incluso si los otros dos módulos no compilan. También se solucionan así problemas de **dependencias circulares** entre ensamblados.

Por impracticabilidad, no se han incluido todos los diagramas de clases de cada módulo ni de todos los ensamblados. Sí se incluyen los que se han considerado más importantes.

## 9.1. ARQUITECTURA GLOBAL

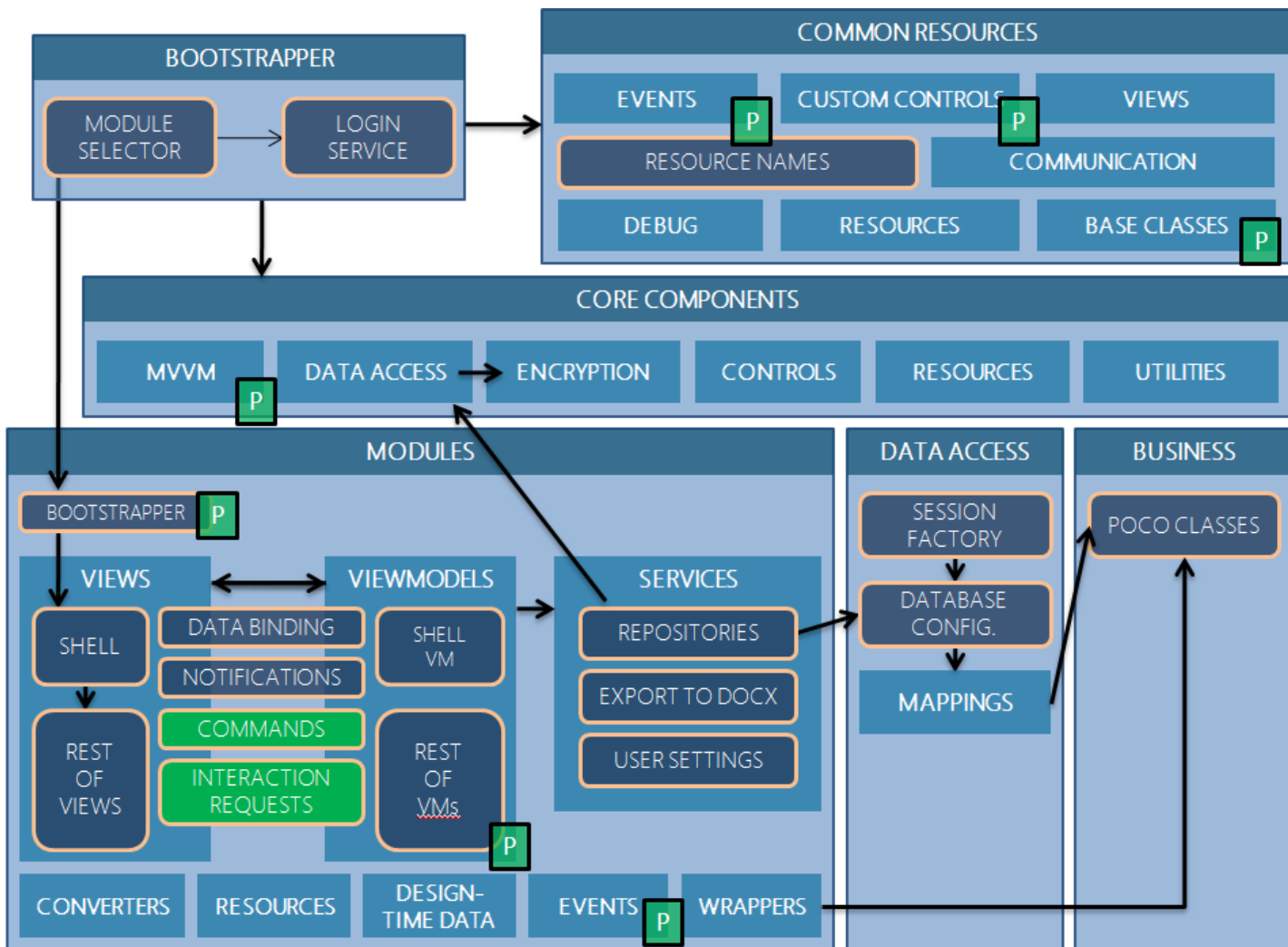


Ilustración 2. Arquitectura global

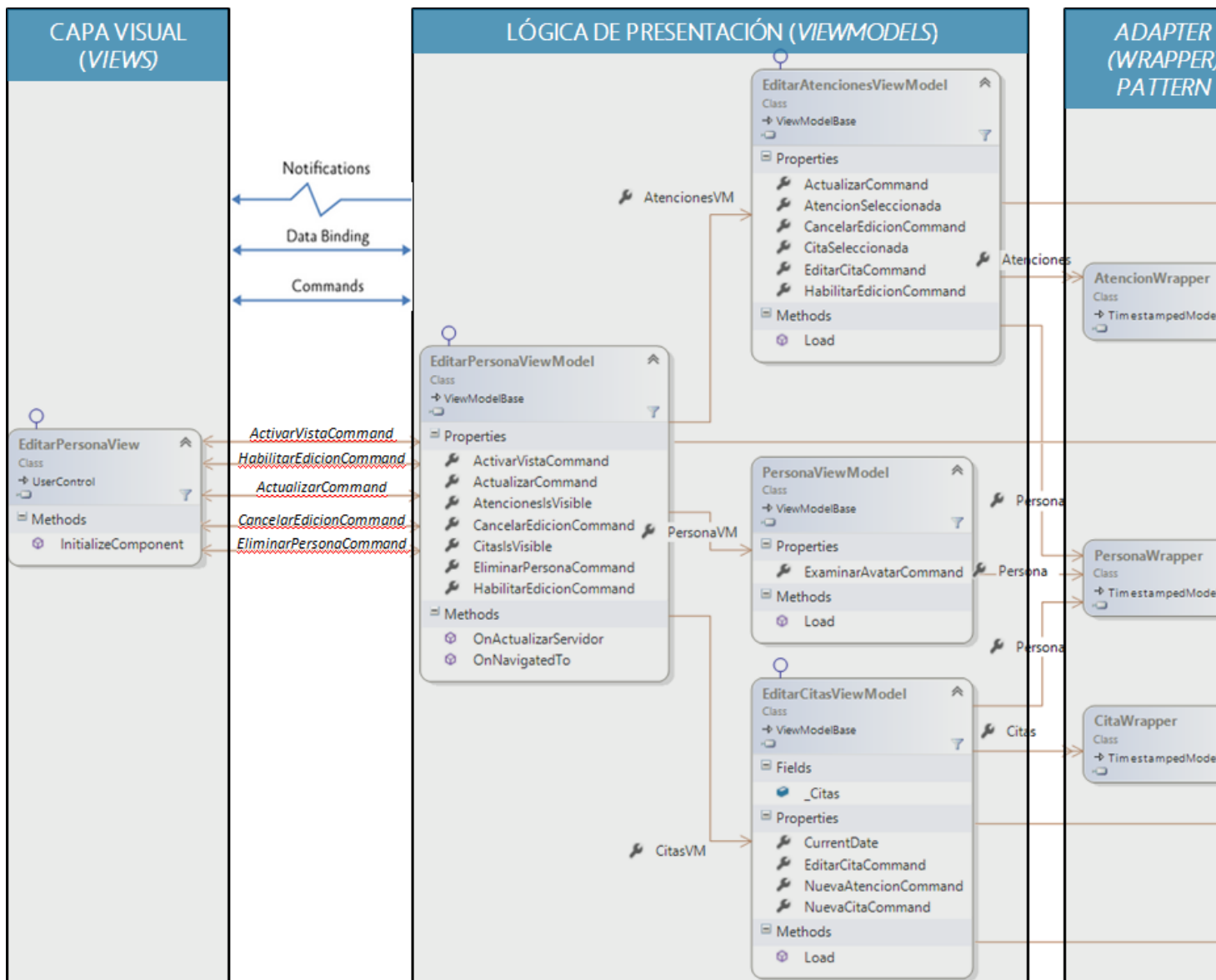


Ilustración 3. Ejemplo concreto de relación entre capas

## 9.2. ORGANIZACIÓN DE ENSAMBLADOS

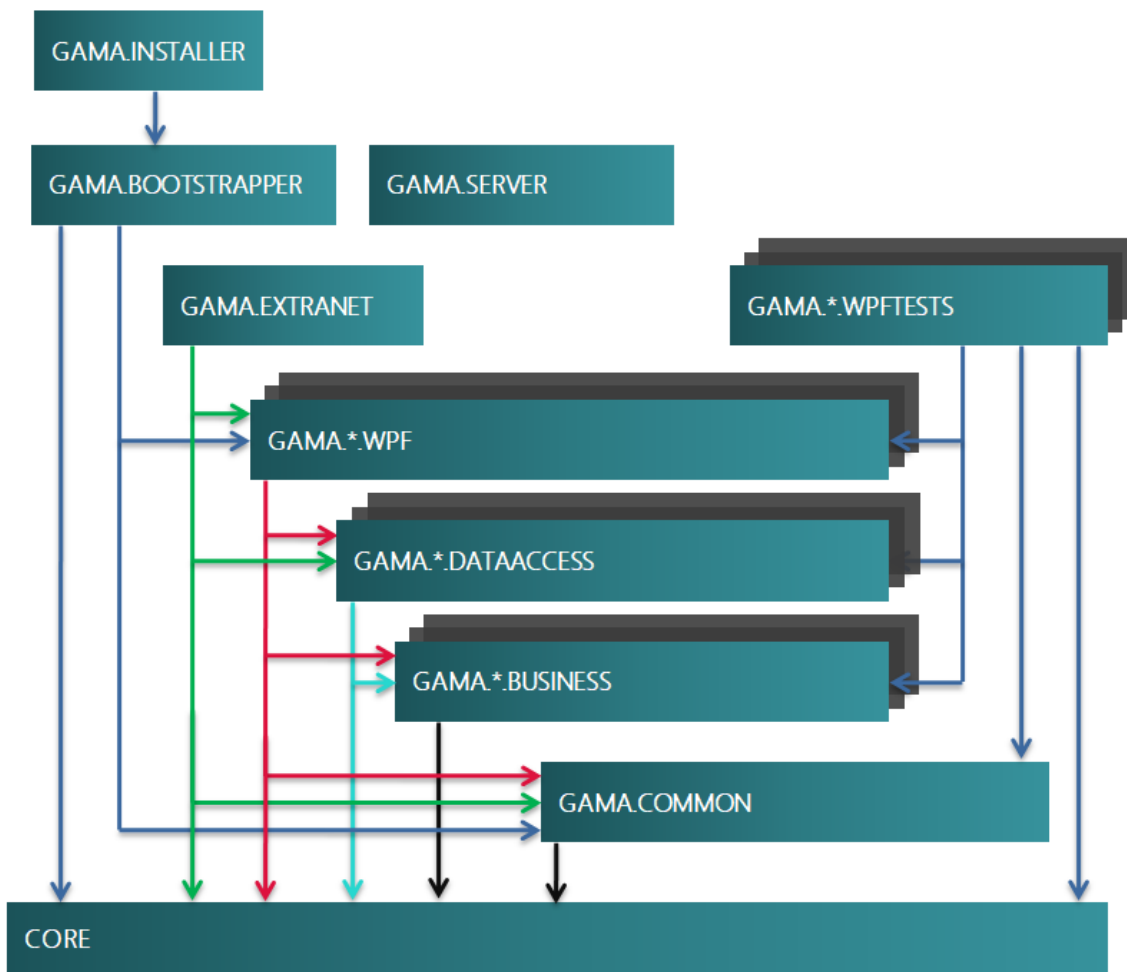
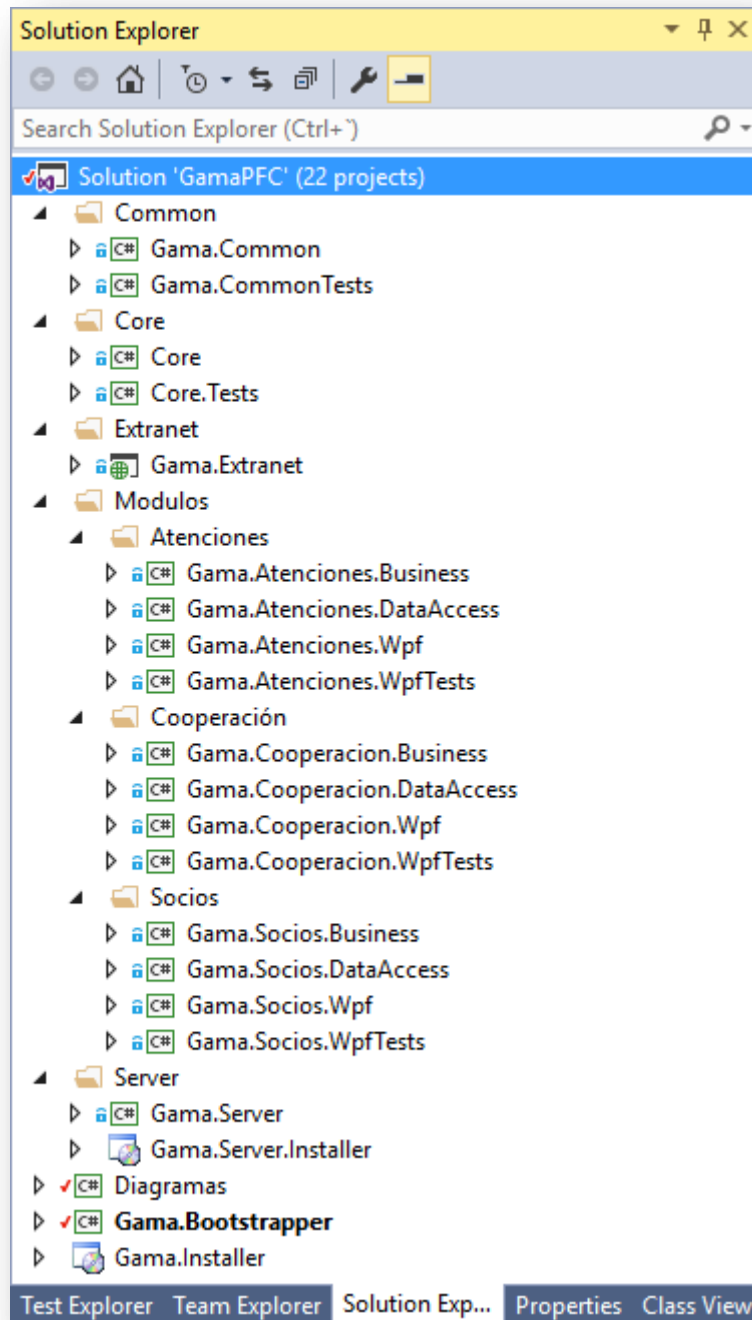


Ilustración 4. Organización de ensamblados

Se han dividido las responsabilidades como sigue:

- **Gama.Installer:** Instalador para Windows.
- **Gama.Server:** Servidor de sincronización.
- **Gama.Bootstrapper:** Ensamblado de inicio. Se encarga de lanzar el selector de módulo y posteriormente de lanzar el módulo seleccionado.
- **Gama.Extranet:** Sitio Web alojado en el servidor dedicado para acceder con cualquier dispositivo a la información de la base de datos en forma de listados.
- **Gama.\*.WpfTests:** Pruebas unitarias de los proyectos *Gama.\*.Wpf*
- **Gama.\*.Wpf:** Ensamblado donde se alojan las vistas, *ViewModels*, servicios, *Wrappers* y componentes de apoyo específicos a cada módulo (acciones, conversores, datos para tiempo de diseño, eventos, estilos, recursos gráficos, ...).

- **Gama.\*.DataAccess:** Contiene las clases de mapeo *fluent* así como el *Session Factory* –donde se crea la configuración de acceso para el ORM) específicos a cada módulo. El mapeo basado en código facilita el desarrollo y mantenimiento de dichos mapeos. La alternativa y única forma en el pasado, es usar ficheros XML específicos de NHibernate. Se hace más complicado gestionar ficheros XML que código altamente legible.
- **Gama.\*.Business:** Clases *POCO (Plain Old CLR Object)* del dominio del problema. Encapsularlo de esta forma permite su fácil reutilización, como ha sucedido al desarrollar la Extranet.
- **Gama.Common:** Clases y recursos comunes a más de un módulo.
- **Core:** Clases y recursos comunes a distintos proyectos desarrollados o por desarrollar con WPF.

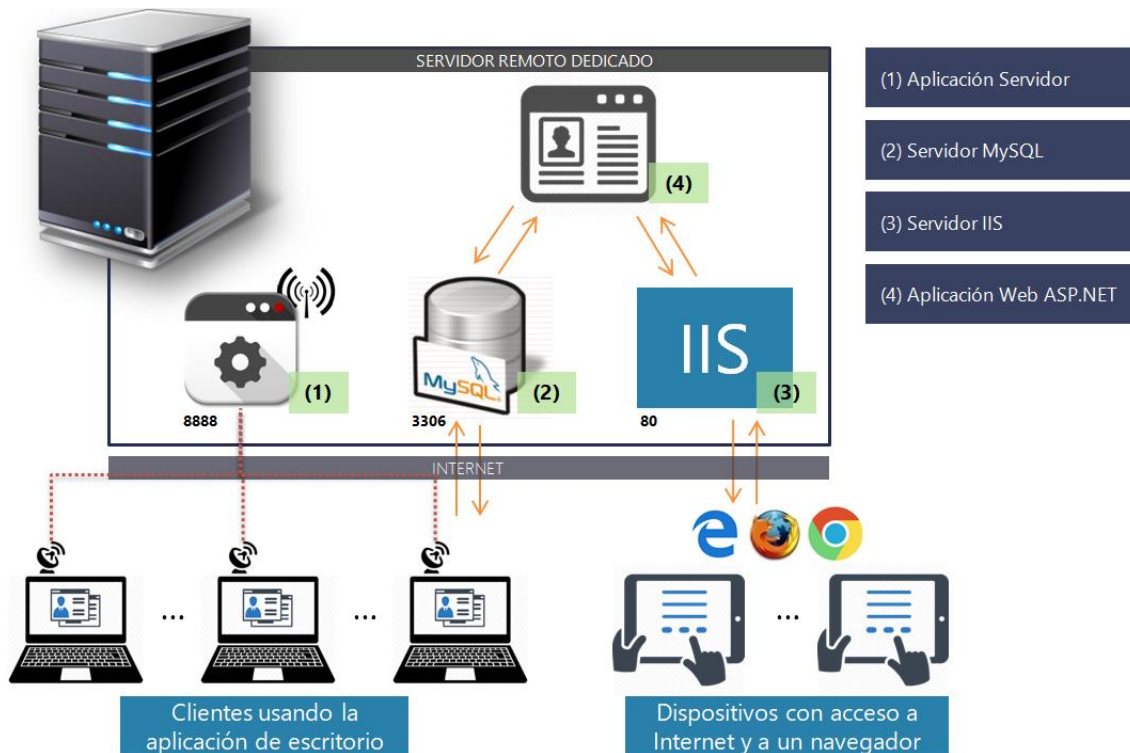


5. Organización de ensamblados en Visu al Studio

### 9.3. DESPLIEGUE

En este capítulo se muestra el diagrama de despliegue y se describen brevemente su funcionamiento.





6. Diagrama de despliegue

Donde:

1. **Aplicación servidor:** Aplicación en constante funcionamiento donde corre el programa que permite la sincronización en tiempo real de las actualizaciones que emiten los clientes.
2. **Servidor MySQL:** Servidor de base de datos.
3. **Servidor IIS:** Servidor Web para permitir peticiones a la extranet.
4. **Aplicación Web ASP.NET:** La extranet.

### 9.3.1. Funcionamiento del servidor de sincronización

El servidor de sincronización está diseñado con tecnología de sockets, de forma que está siempre esperando a recibir nuevas conexiones (que solicitan los clientes al lanzarse un módulo). Cuando un cliente realiza una actualización, emite un mensaje al servidor, el cual emite un broadcast al resto de clientes para que se actualice el módulo correspondiente (se controla que si emite un cliente de socios, no se actualice un módulo de atenciones receptor del mensaje, por ejemplo).

En caso de que el servidor no esté activo, el cliente se intentará conectar en segundo plano cada varios segundos. Una vez vuelva a estar en funcionamiento el servidor de sincronización, el módulo se conectará rápidamente.

### 9.3.2. Organización de carpetas de la aplicación en el cliente

Debido a que hay que guardar en el cliente una copia de la serialización de la configuración de acceso a la base de datos, las preferencias de usuario, las imágenes cacheadas de la base de datos por cuestiones de rendimiento y las imágenes por defecto, para cada módulo, se ha tenido que plantear una estructura de carpetas que lo soporte.

Esta estructura se comprueba cada vez que se lanza la aplicación, creando las carpetas y archivos que por alguna razón no estuvieran presentes (porque los ha borrado el usuario, por ejemplo).

## 9.4. ARRANQUE DE LA APLICACIÓN

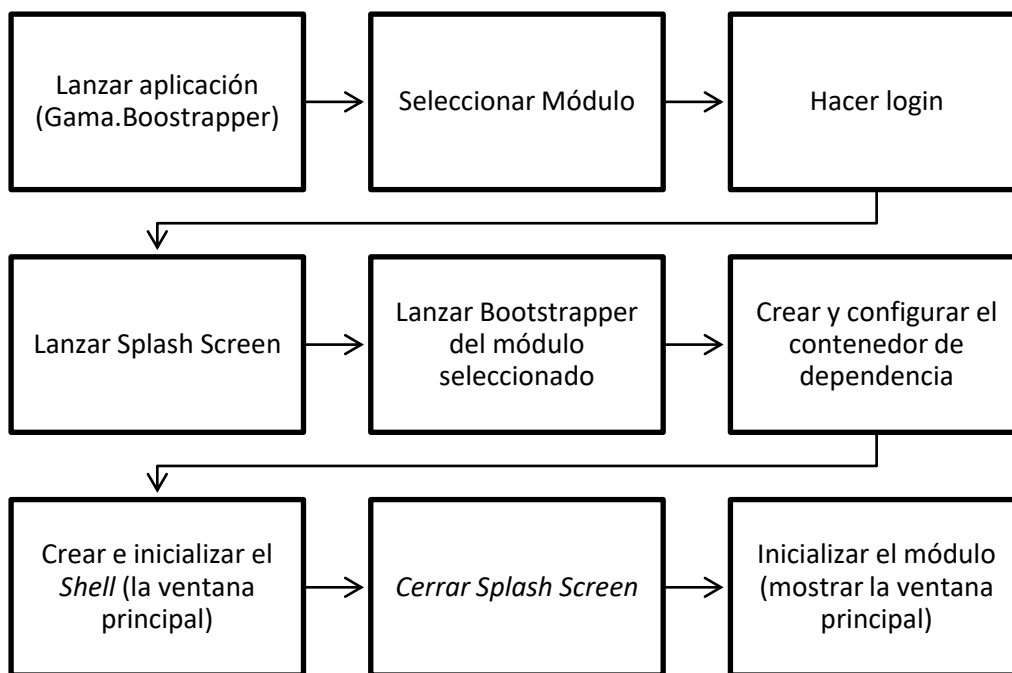


Ilustración 7. Flujo de etapas del arranque de la aplicación

## 9.5. MVVM (MODEL-VIEW-VIEWMODEL)

Sin pretender ser exhaustivos, sí se ha considerado relevante dedicar un apartado a los aspectos más importantes del patrón de diseño arquitectónico que se ha adoptado, debido a la relación directa que tiene con varios de los objetivos planteados para este proyecto, así como porque no resulta difícil desestimar superficialmente, por excesivo, el uso de un patrón como este. Esto último puede atender a varias razones. Entre ellas se encuentra la acusada curva de aprendizaje que supone desarrollar software con WPF (*Windows Presentation Foundation*) junto a MVVM. Otro motivo podría tener que ver con la tendencia general en la comunidad de desarrolladores a no dedicar mucho tiempo a las pruebas automatizadas. Esto se puede deber a que las consideran una pérdida de tiempo, pero también hay que considerar que escribir pruebas requiere tener otro *mindset*, que hay que aprender teoría y técnicas específicas, y que el impacto se hace mayor ya que te obliga a diseñar y programar el resto de código de otra manera. Todos estos elementos generan resistencia.

A continuación se explica la motivación que hay para usar MVVM, y las características principales de este patrón de diseño arquitectónico.

### 9.5.1. Motivación para usar MVVM

Cuando una aplicación que se ha desarrollado sin atender a la arquitectura crece en tamaño y alcance, el testeo y el mantenimiento se vuelven complejos. Es entonces cuando aparecen problemas, como el alto acoplamiento entre las vistas y la lógica de negocio. Esto dificulta la introducción de modificaciones en la capa visual así como el desarrollo de pruebas unitarias.

MVVM fue desarrollado por Microsoft con estas cuestiones en mente, tratando además de aprovecharse de las características específicas de WPF.

Los **beneficios se MVVM** para el flujo de trabajo se pueden son:

- Durante el proceso de desarrollo, **las actividades de desarrollo y diseño (de interfaces de usuario) se pueden lleva a cabo de forma más independiente y concurrente** en sus componentes. Los diseñadores pueden concentrarse en la

vista, usando por ejemplo *Expression Blend* para ayudarse, mientras los desarrolladores programan los *view models* y el resto de componentes.

- Se pueden **crear pruebas unitarias para el modelo y los *view models* sin utilizar la vista**. Esta podría ni existir y aún así se podrían desarrollar el modelo y los *view models* completamente (aunque se recomienda un flujo donde haya retroalimentación entre actividades).
- **Es fácil rediseñar la interfaz de usuario** de la aplicación **sin tocar el código** porque las vistas están implementadas completamente en XAML. Nuevas versiones de las vistas funcionarán con los *view models* existentes en el caso general.
- Encapsular, adaptando, a través del *view model* las clases del modelo para las vistas permite **dejar intacto el modelo**, lo cual podría resultar difícil, arriesgado o incluso imposible (si por cuestiones organizativas, de compatibilidad o de otro tipo no es posible modificarlo) modificarlo.

Otros motivos para adoptar MVVM vienen dados por las expectativas futuras sobre MVVM y WPF en general. El mecanismo de *Data Binding* parece que va encontrado su lugar en el mundo del desarrollo software. Por ejemplo, tenemos algunos *frameworks* de Javascript que lo usan (como KnockoutJS, Kendo MVVM, Knockback.js), entre los cuales se encuentra Angular 2, que es mantenido por Google en código abierto. El mecanismo de *Data Binding* facilita el desarrollo de componentes desacoplados, en particular entre las vistas y la lógica de negocio. Siendo este un sempiterno objetivo de la ingeniería del software, se entiende que es razonable que la industria esté evolucionando en esa dirección.

### 9.5.2. MVVM (Model-View-ViewModel)

MVVM es un patrón de diseño arquitectónico que facilita la separación entre la interfaz gráfica de usuario (front-end) y la lógica de negocio (back-end) en cuanto a su desarrollo se refiere. Es decir, que permite desarrollarlos de forma independiente.

Para hacer esto posible, MVVM ha de contar con los siguientes componentes:

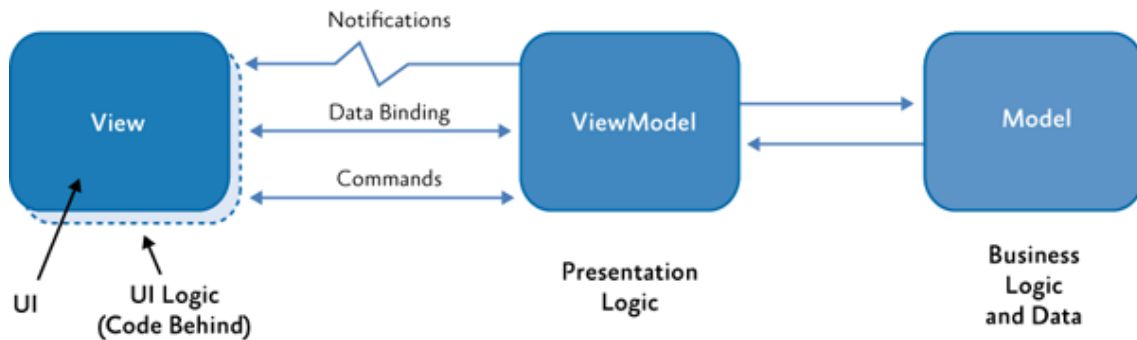


Ilustración 8. Arquitectura MVVM

Donde, además de los componentes habituales (*views* y *models*), encontramos:

- El **ViewModel** es una interfaz o intermediario entre el back-end y el front-end que se encarga de preparar los datos que recibe del back-end (transformándolos o disponiendo de ellos en distintas formas) de forma que facilite a la capa visual los datos consumir y presentar los datos de forma cómoda. Se puede entender como que un *view model* es, en efecto, un *modelo* para una *vista*. Es práctica extendida el asociar cada vista con un único *view model*.
- **Notifications** y **Data Binding** se refieren a las notificaciones que se envían entre el ViewModel y la vista cuando alguno de los dos ha actualizado algún objeto que esté enlazado (que tenga un *binding* entre el ViewModel y la vista).
- **Commands** como forma de encapsular las acciones disponibles para el usuario. Basta con hacer referencia a éstos en la vista e implementarlos después en el ViewModel. Incluso si no se implementan en absoluto, no tendrá lugar ningún error, si bien esas acciones no surtirán efecto alguno.

Para finalizar, cabe mencionar que tanto MVVM como la librería Prism han sido desarrollados por Microsoft, si bien a día de hoy Prism es de código abierto y [se aloja en GitHub](#).

## 9.6. CLASES DE LOS ENSAMBLADOS COMUNES

En este capítulo se muestran los componentes más relevantes de los ensamblados comunes (*Core* y *Gama.Common*).

## 9.6.1. Ensamblado Core

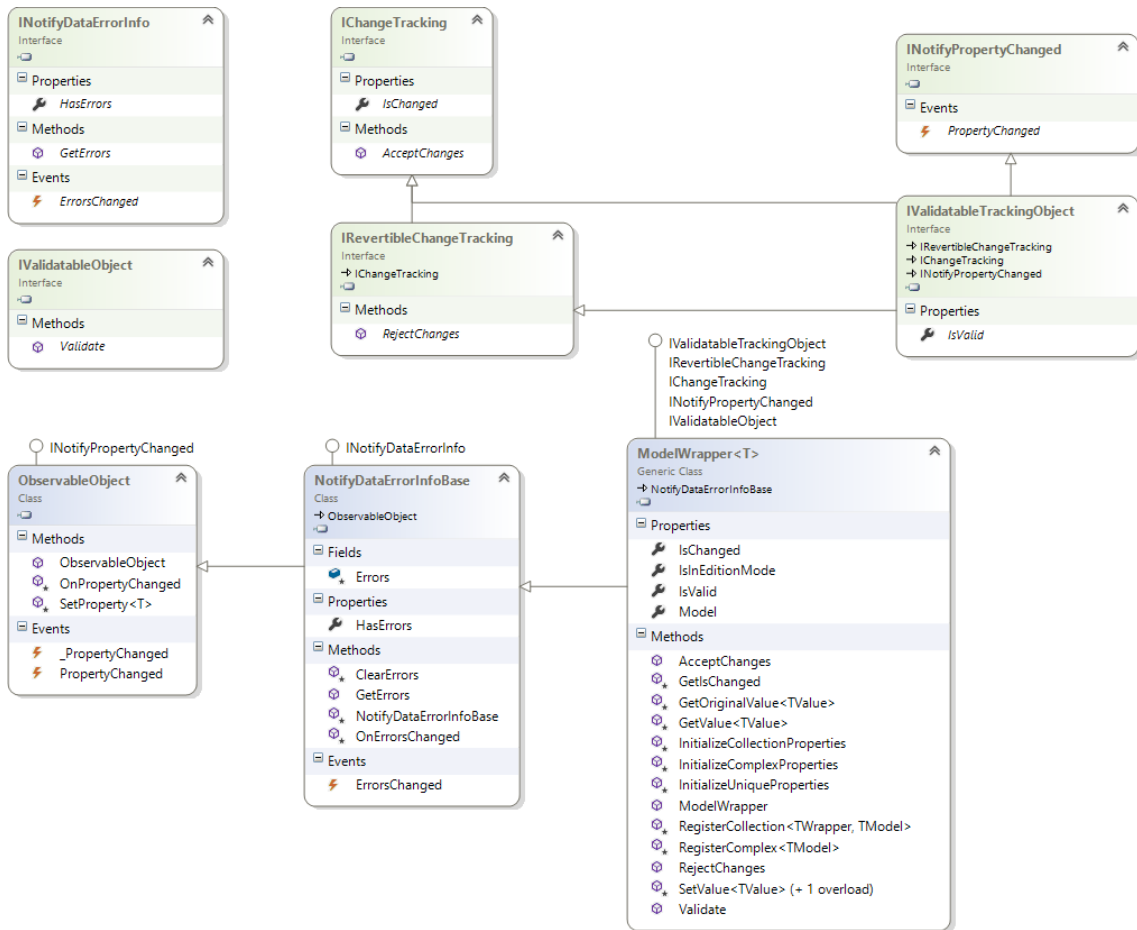


Ilustración 9. Clases comunes en Core

## 9.6.2. Ensamblado *Gama.Common*

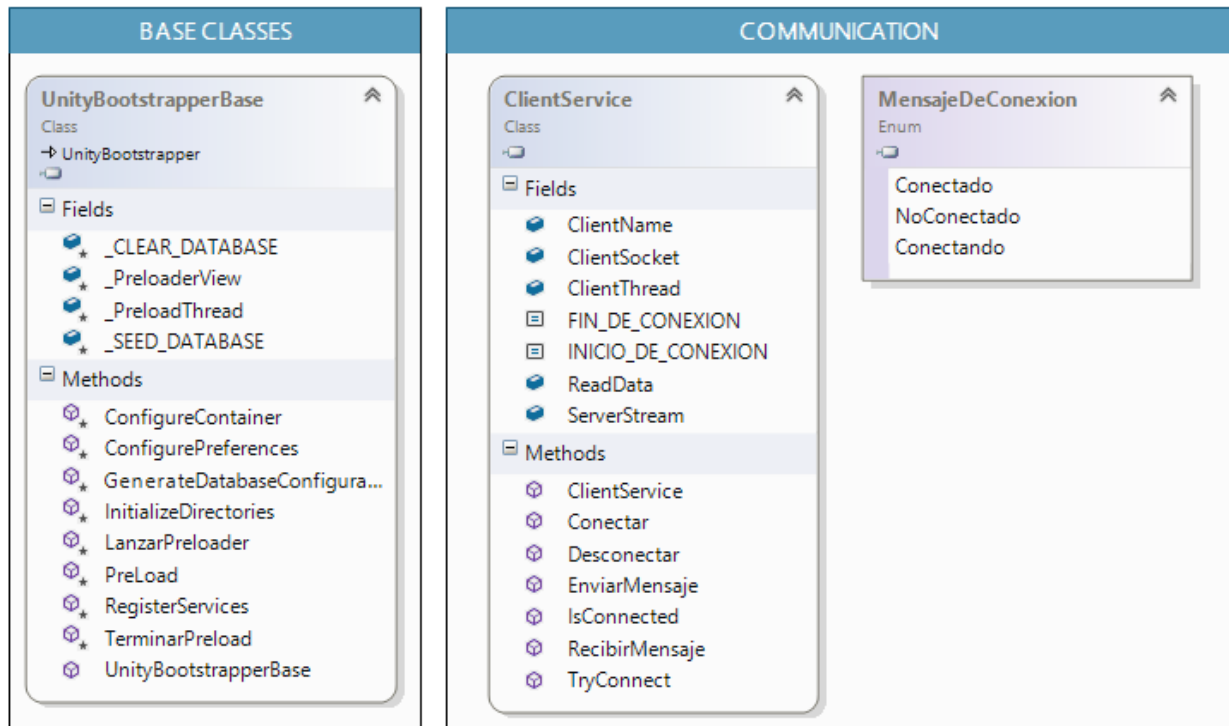


Ilustración 10. Clases base y comunicación

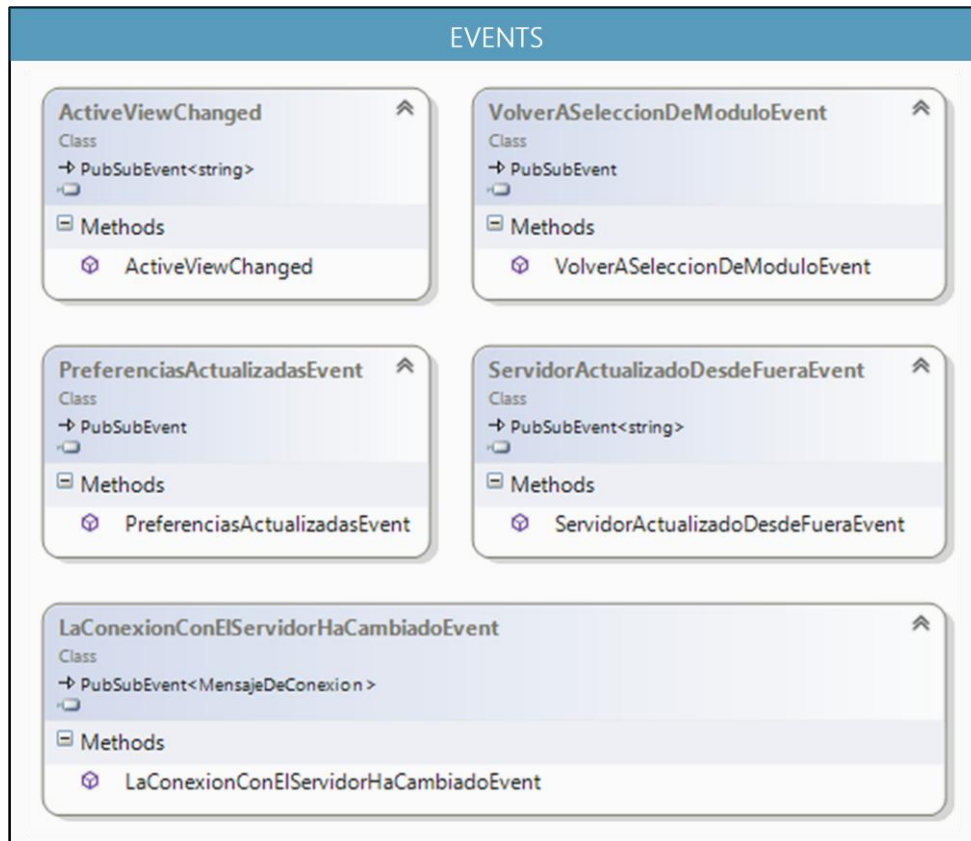


Ilustración 11. Eventos comunes



## CUSTOM CONTROLS: SEARCHBOX

The image displays the Visual Studio IDE showing the class hierarchy for a custom SearchBox control. The main window on the left shows the SearchBox class, which inherits from TextBox. It lists various fields, properties, methods, and events. Three smaller windows on the right provide details for related classes: SearchMode (an Enum), LookupItem (a class inheriting from BindableBase), and SearchBoxDataTemplateSelector (a class inheriting from DataTemplateSelector).

**SearchBox Class**

- Class
  - ↳ TextBox
- Fields
  - HasTextProperty
  - IsMouseButtonDownProperty
  - ItemContainerStyleProperty
  - ItemsSourceProperty
  - ItemTemplateProperty
  - ItemTemplateSelectorProperty
  - LabelTextColorProperty
  - LabelTextProperty
  - MaxCompletionsProperty
  - SearchEvent
  - SearchEventTimeDelayProperty
  - SearchModeProperty
  - SelectedItemProperty
  - SelectResultEvent
- Properties
  - HasText
  - IsMouseButtonDown
  - ItemContainerStyle
  - ItemsSource
  - ItemTemplate
  - ItemTemplateSelector
  - LabelText
  - LabelTextColor
  - MaxCompletions
  - SearchEventTimeDelay
  - SearchMode
  - SelectedItem
- Methods
  - OnItemsSourceChanged
  - OnKeyDown
  - OnMouseDoubleClick
  - OnPreviewKeyDown
  - OnTextChanged
  - ShowPopup
- Events
  - Search
  - SelectResult

**SearchMode Enum**

- Enum
  - Delayed
  - Instant

**LookupItem Class**

- Class
  - ↳ BindableBase
- Properties
  - DisplayMember1
  - DisplayMember2
  - DisplayMember3
  - Email
  - EstaDadoDeAlta
  - FechaDeInicioActividad
  - FechaDeNacimiento
  - IconSource
  - Id
  - Id\_Coordinador
  - Imagen
  - Nacionalidad
  - Telefono
- Methods
  - LookupItem
  - ShortenStringForDisplay

**SearchBoxDataTemplateSelector Class**

- Class
  - ↳ DataTemplateSelector
- Methods
  - SearchBoxDataTemplateSelector
  - SelectTemplate

Ilustración 12. Controles personalizados (1/3)

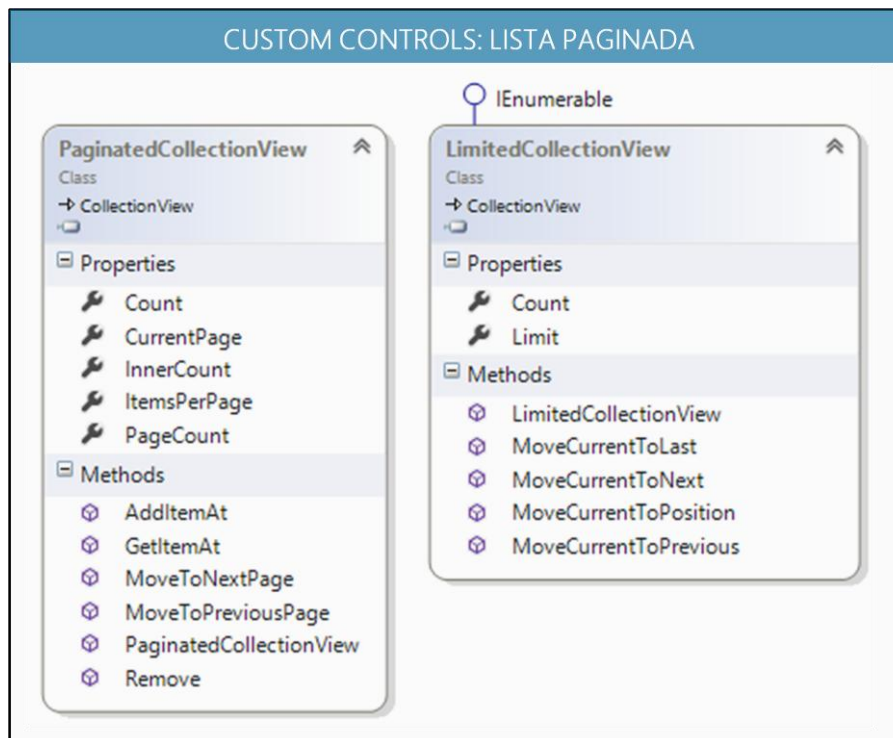


Ilustración 13. Controles personalizados (2/3)

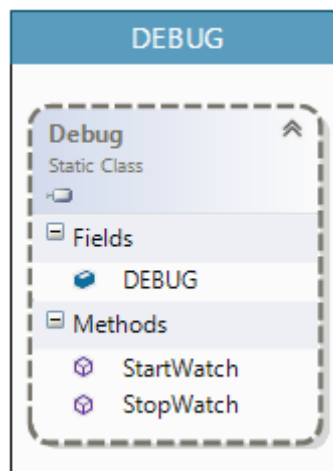


Ilustración 14. Depuración

## CUSTOM CONTROLS: CALENDARIO

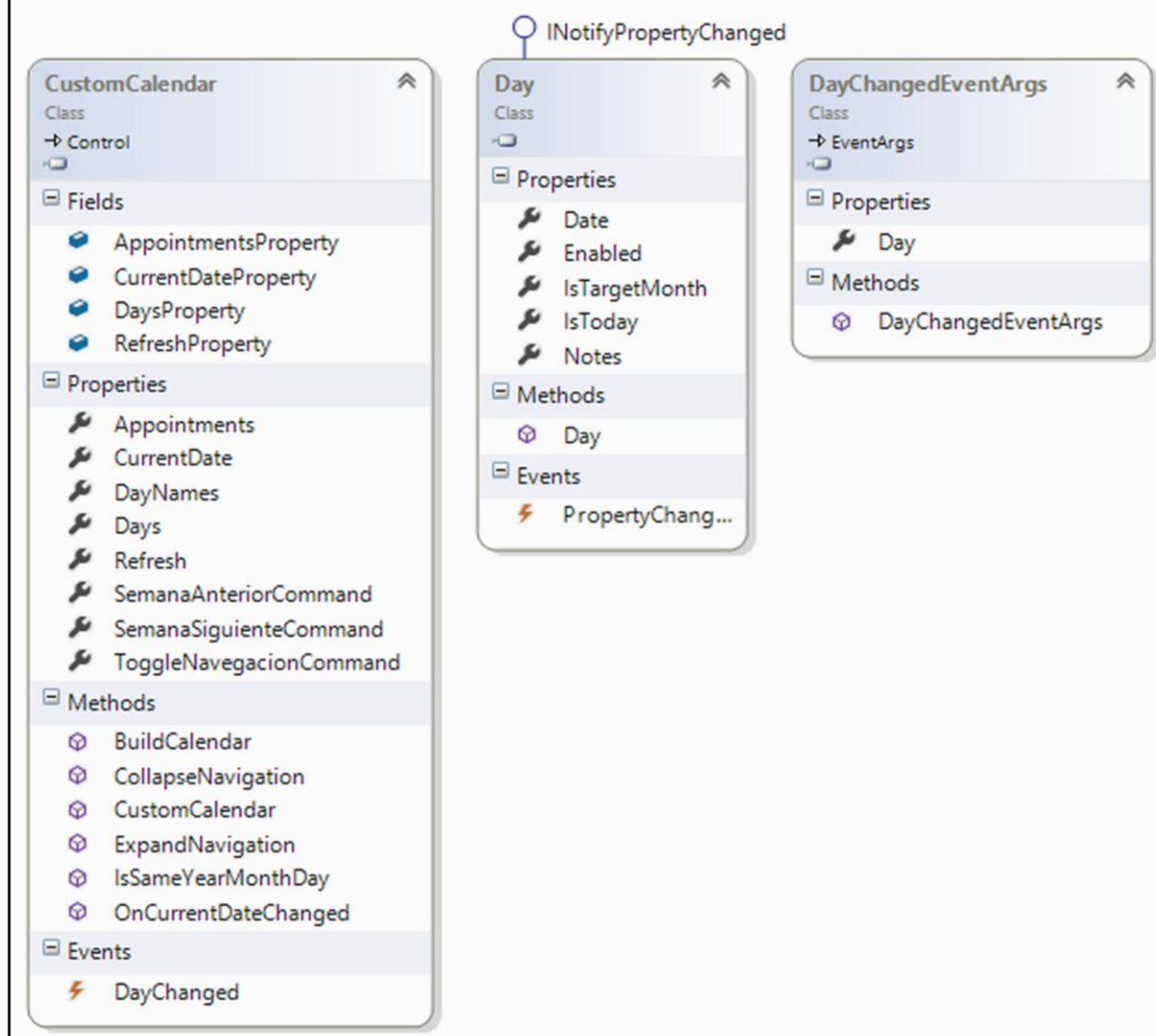


Ilustración 15. Controles personalizados (3/3)

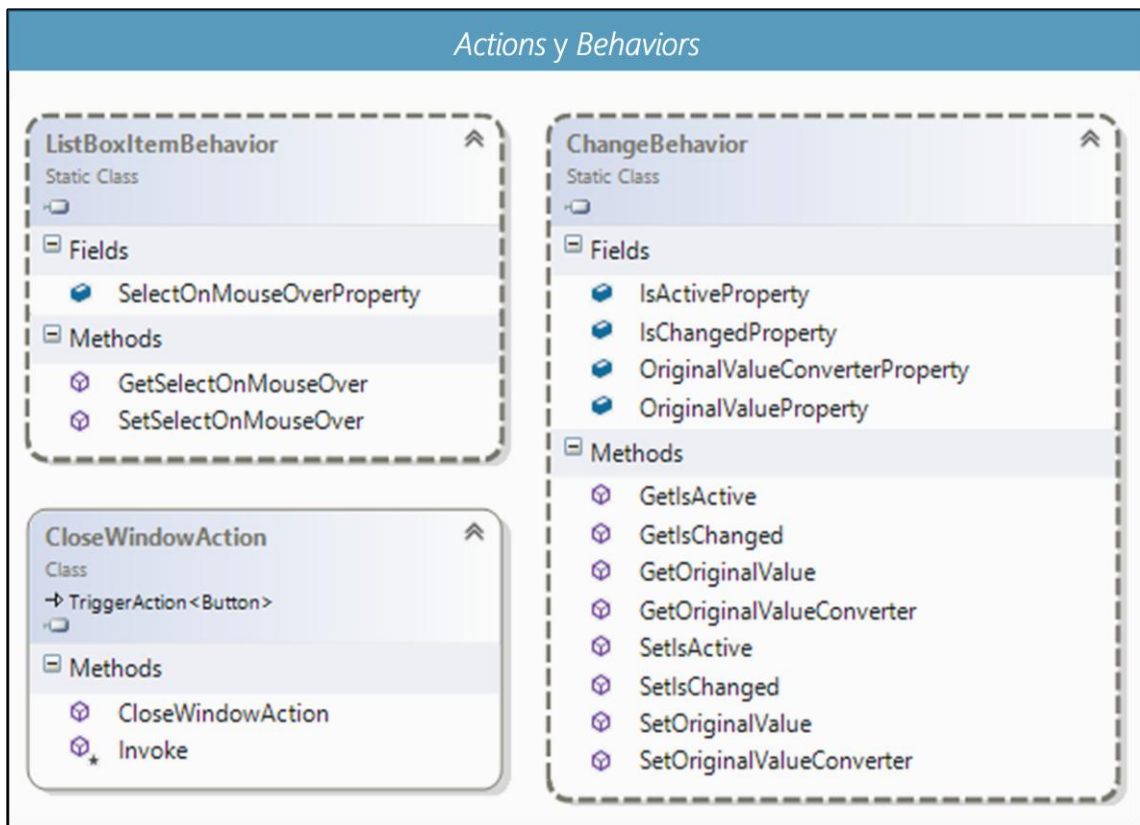
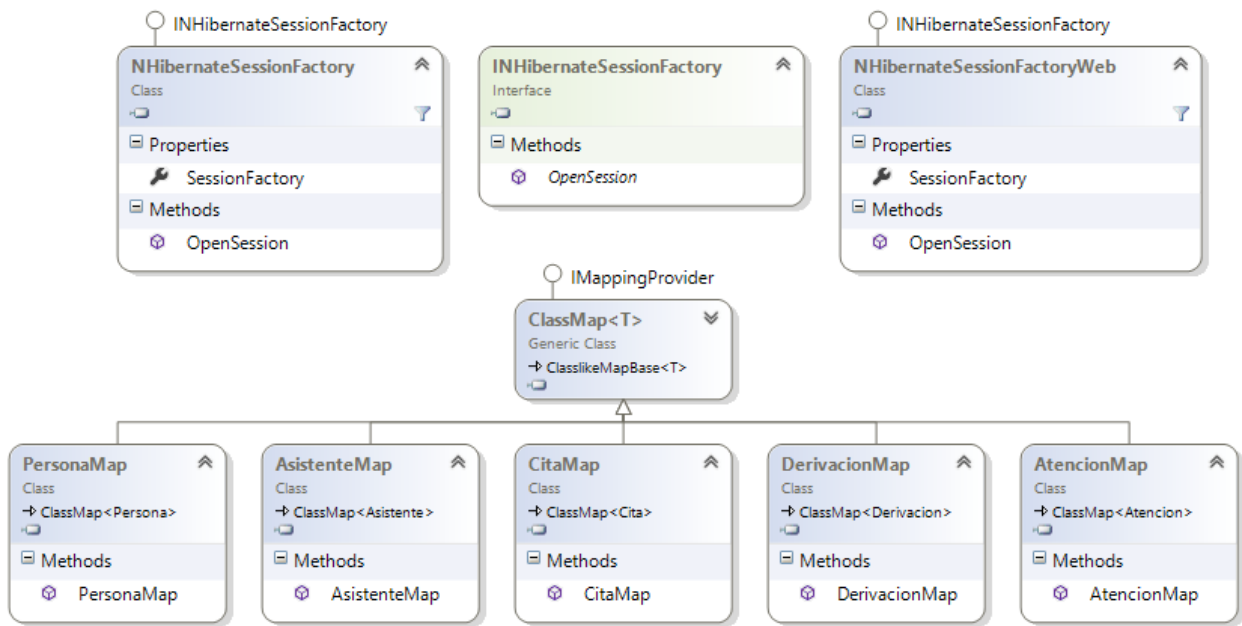


Ilustración 16. Acciones y comportamientos

## 9.7. CLASES DEL MÓDULO DE ATENCIONES

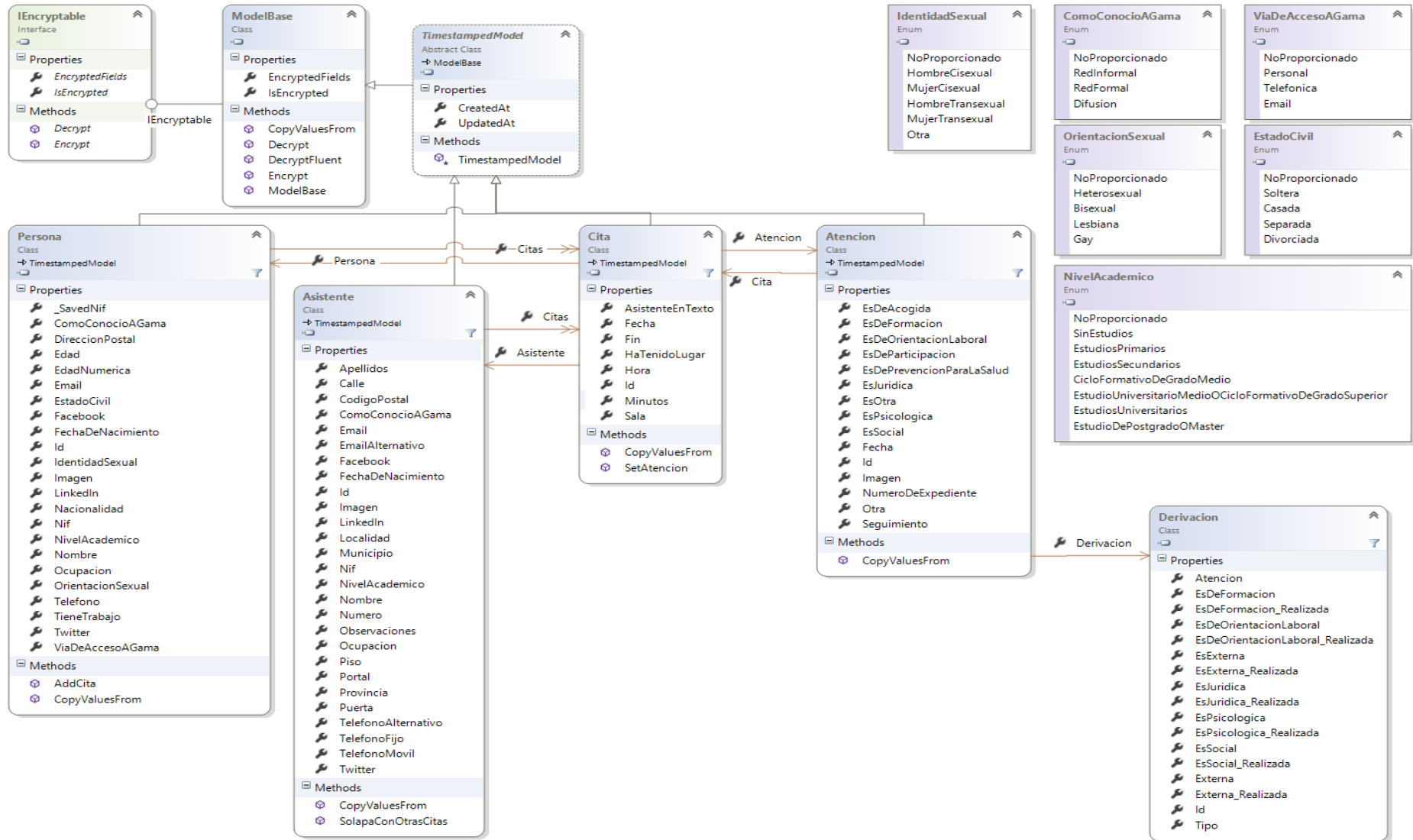
Se muestran en esta sección los diagramas de clases para los ensamblados del módulo de atenciones. No se incluyen todas las clases por una cuestión de presentación. El conjunto de vistas, *view-models* y resto de clases específicas quedan ya reflejadas en el esquema global de la arquitectura al principio de este capítulo.

### 9.7.1. Capa de acceso a datos (*Gama.Atenciones.DataAccess*)



17. Diagrama de clases de *Gama.Atenciones.DataAccess*

### 9.7.2. Capa de negocio (Gama.Atenciones.Business)



### 9.7.3. Eventos (*Gama.Atenciones.Wpf.Eventos*)

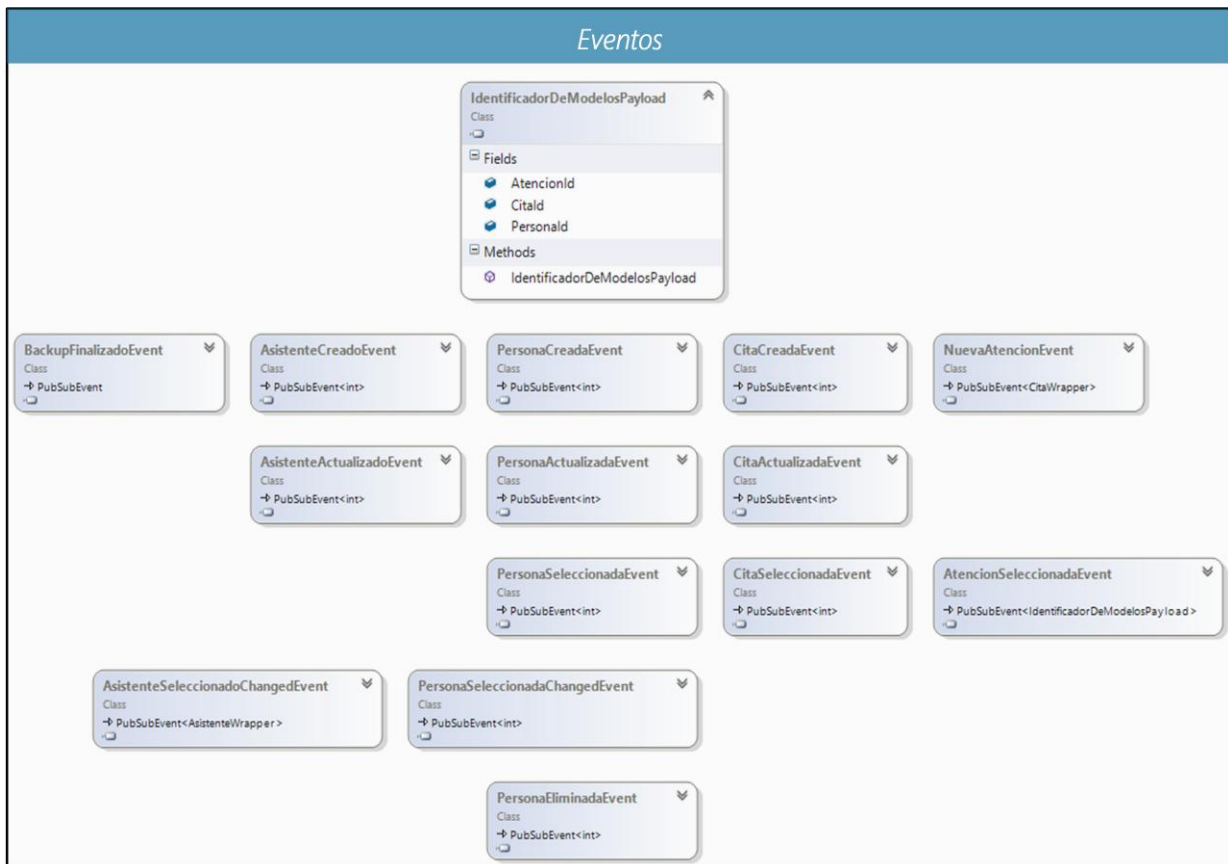


Ilustración 18. Eventos específicos al módulo de atenciones

## 9.8. CLASES DEL MÓDULO DE SOCIOS

Se muestran en esta sección los diagramas de clases para los ensamblados del módulo de socios. No se incluyen todas las clases por una cuestión de presentación. El conjunto de vistas, *view-models* y resto de clases específicas quedan ya reflejadas en el esquema global de la arquitectura al principio de este capítulo.

### 9.8.1. Capa de acceso a datos (*Gama.Socios.DataAccess*)

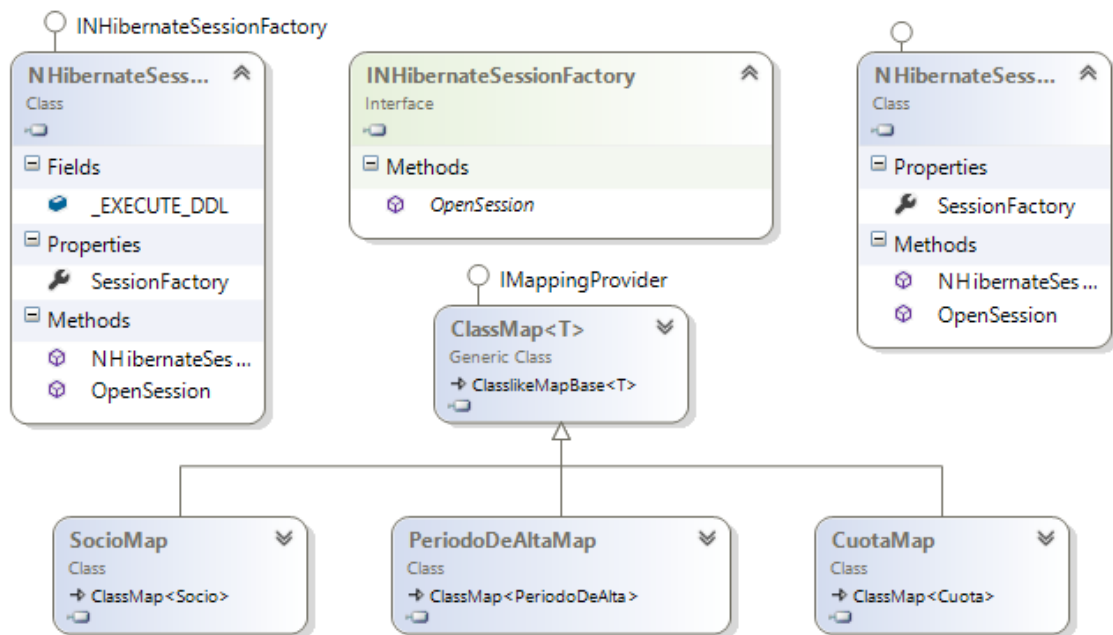


Ilustración 19. Diagrama de clases de *Gama.Socios.DataAccess*



## 9.8.2. Capa de negocio (*Gama.Socios.Business*)

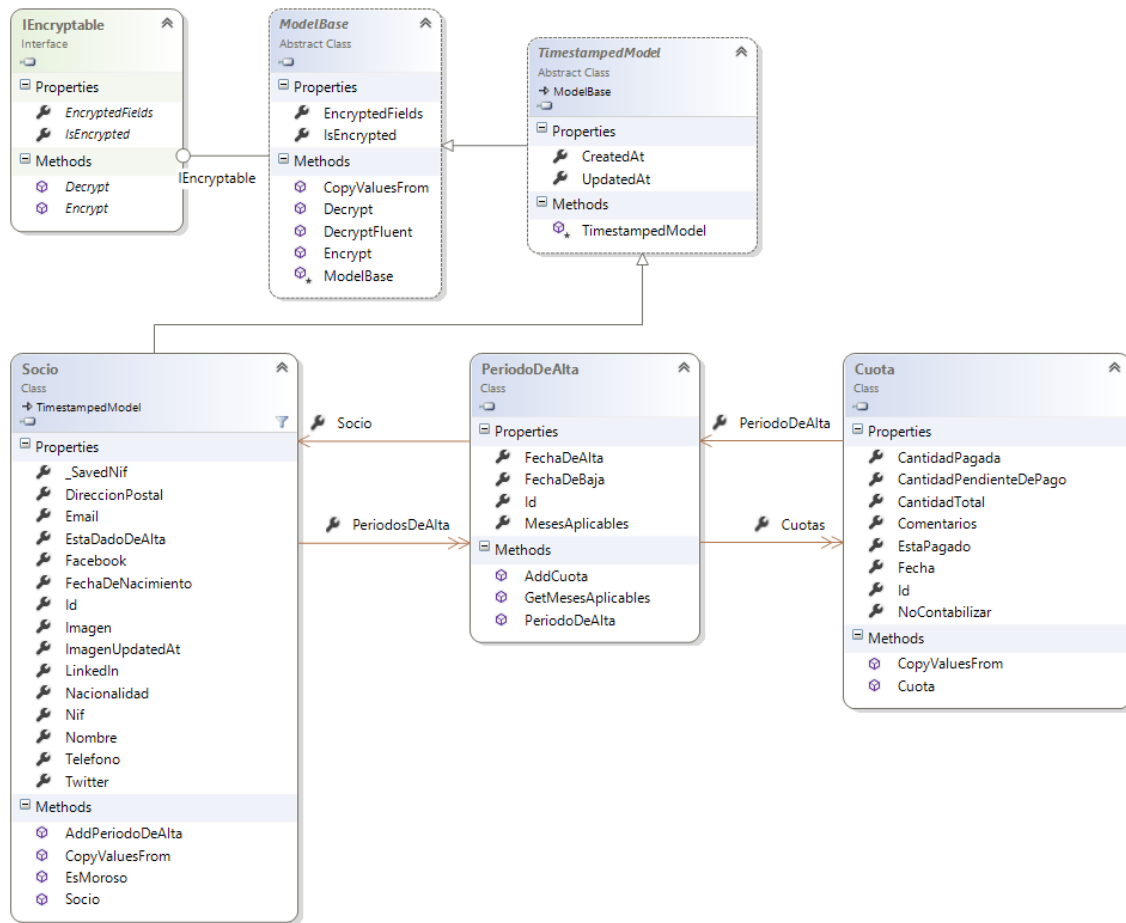


Ilustración 20. Diagrama de clases de *Gama.Socios.Business*

### 9.8.3. Eventos (*Gama.Socios.Wpf.Eventos*)

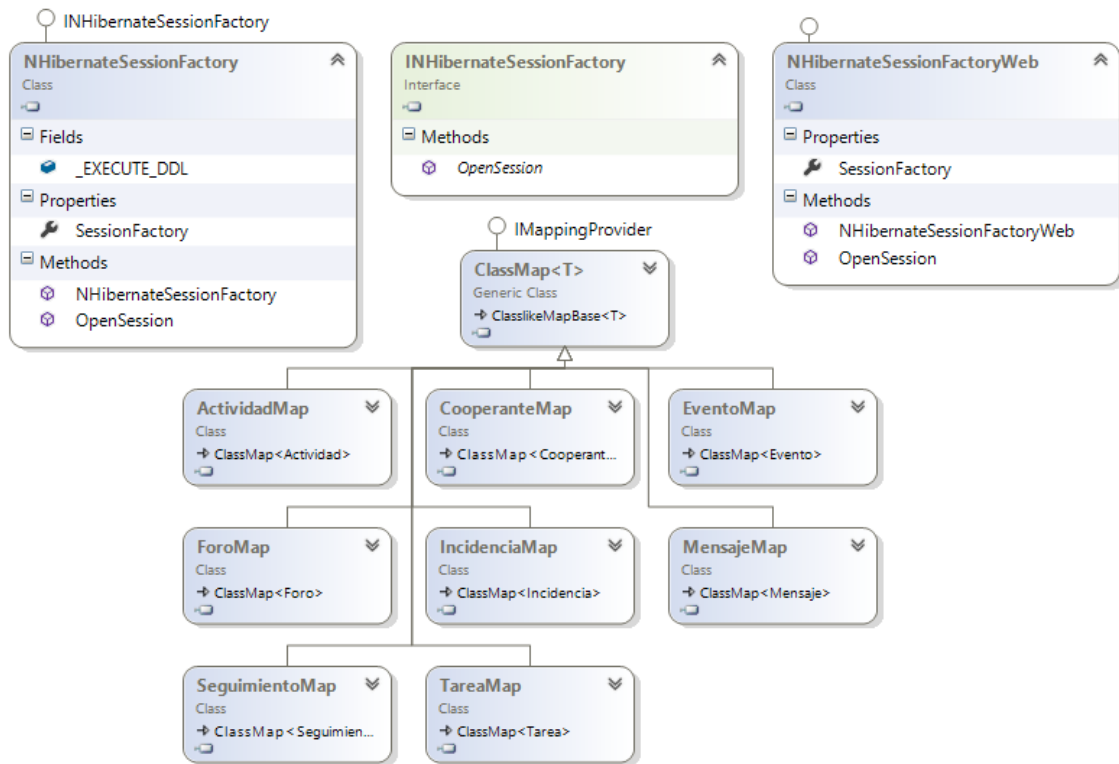


Ilustración 21. Eventos específicos para el módulo de socios

## 9.9. CLASES DEL MÓDULO DE COOPERACIÓN

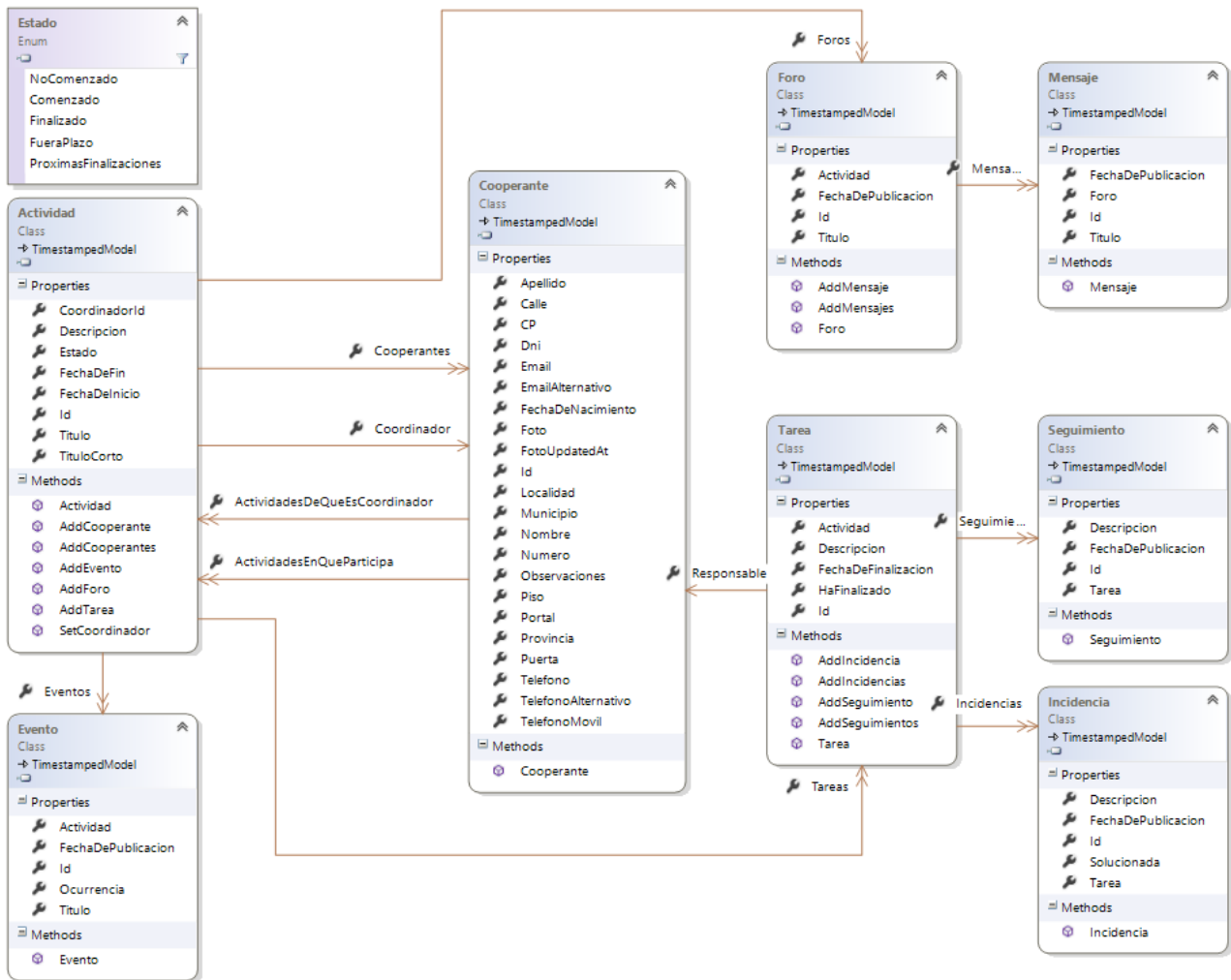
Se muestran en esta sección los diagramas de clases para los ensamblados del módulo de cooperación. No se incluyen todas las clases por una cuestión de presentación. El conjunto de vistas, *view-models* y resto de clases específicas quedan ya reflejadas en el esquema global de la arquitectura al principio de este capítulo.

### 9.9.1. Capa de acceso a datos (*Gama.Cooperacion.DataAccess*)



22. Diagrama de clases de *Gama.Cooperacion.DataAccess*

## 9.9.2. Capa de negocio (*Gama.Cooperacion.Business*)



23. Diagrama de clases de *Gama.Cooperacion.Business*

### 9.9.3. Eventos (*Gama.Cooperacion.Wpf.Eventos*)



24. Eventos específicos para el módulo de cooperacion

## 9.10. ARQUITECTURA CLIENTE-SERVIDOR

No se considera oportuno explicar en qué consiste y cuál es el funcionamiento de una arquitectura cliente-servidor basada en sockets. No obstante, se incluye el siguiente diagrama de clases de la implementación concreta.

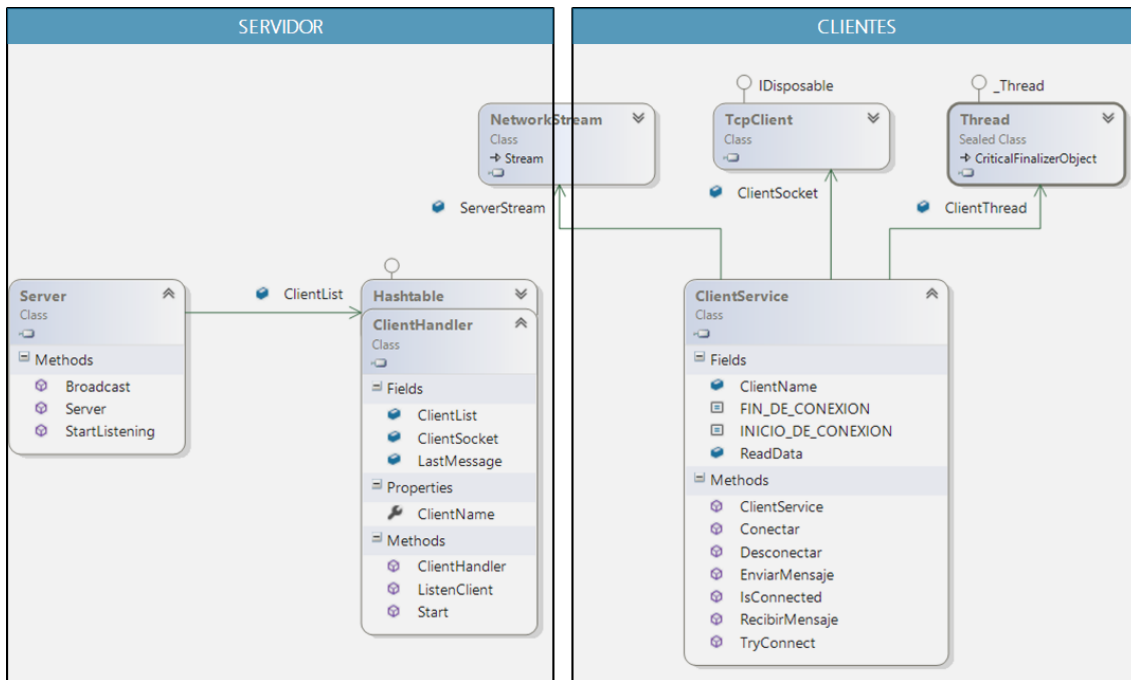


Ilustración 25. Diagrama de las clases que intervienen en la dinámica cliente-servidor

## 9.11. DISEÑO DE LA BASE DE DATOS

En este capítulo se muestran los diagramas entidad-relación de la base de datos.

### 9.11.1. Servicio de atenciones

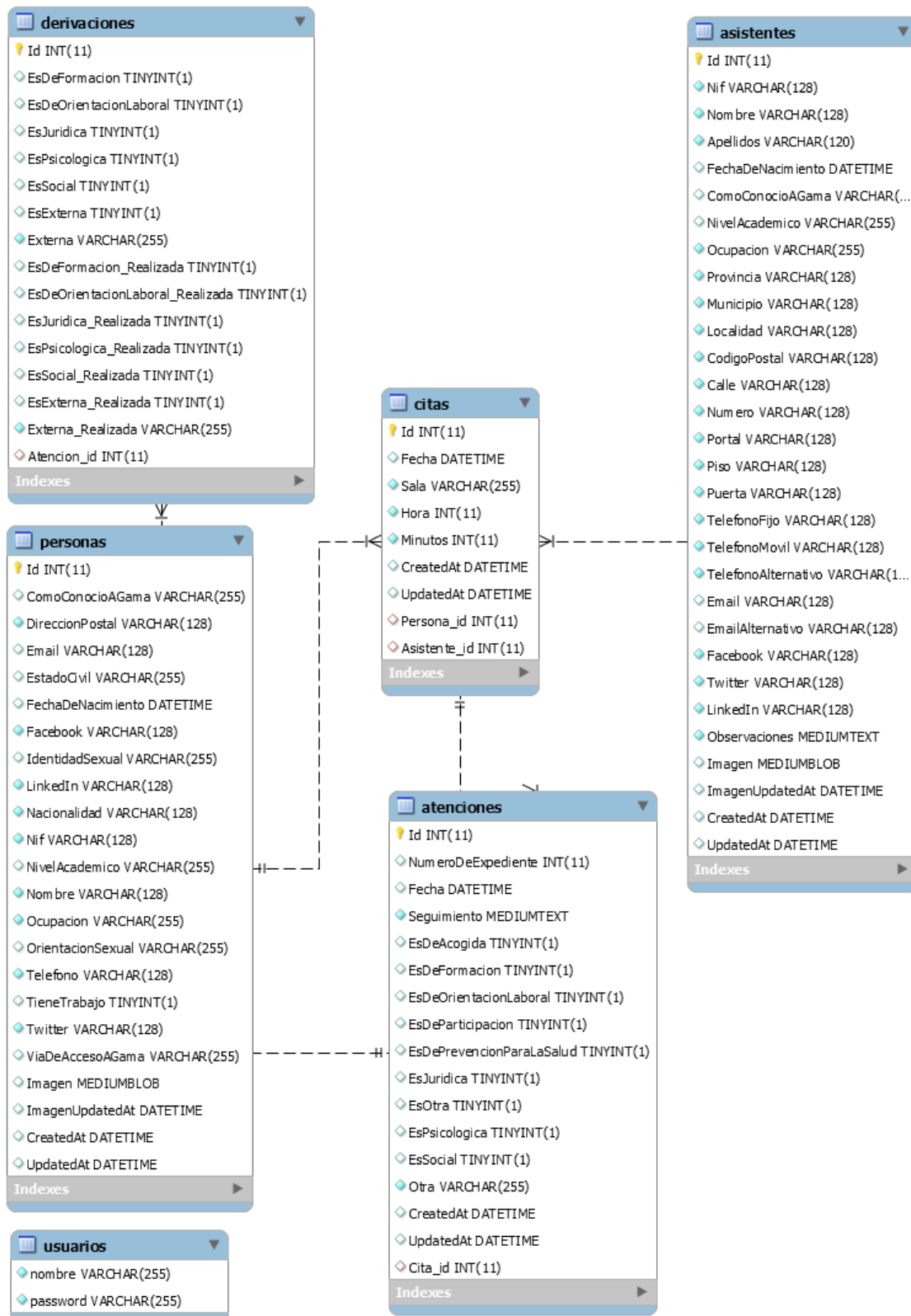


Ilustración 26. Diagrama entidad-relación de la base de datos de atenciones

### 9.11.2. Gestión de socios

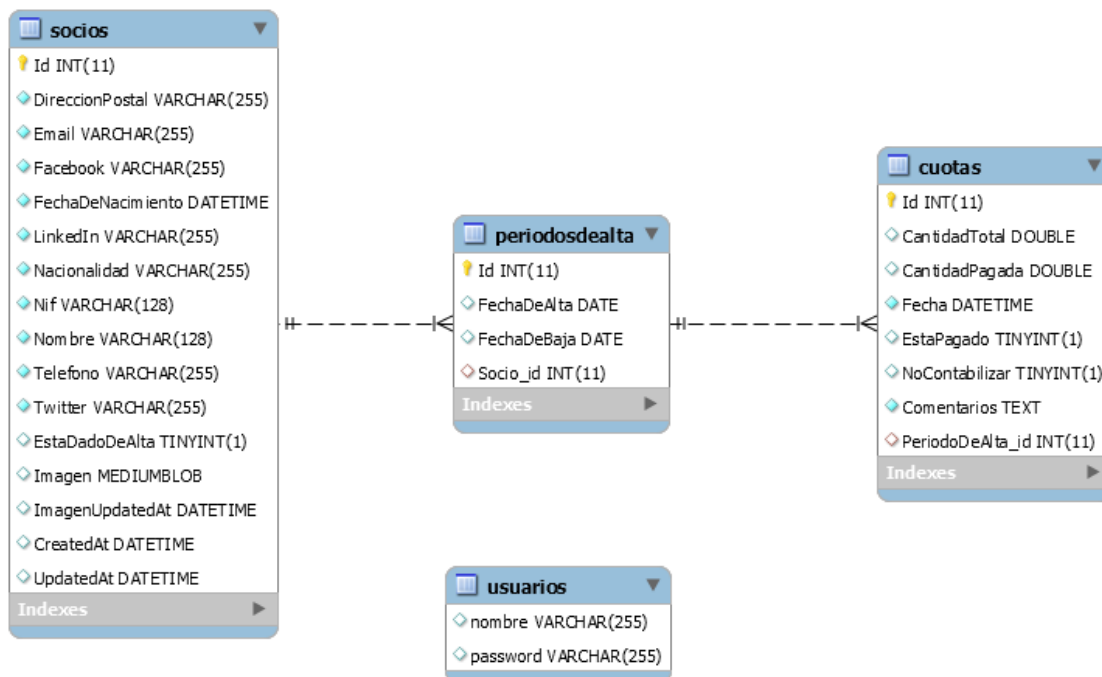


Ilustración 27. Diagrama entidad-relación de la base de datos de socios



### 9.11.3. Cooperación

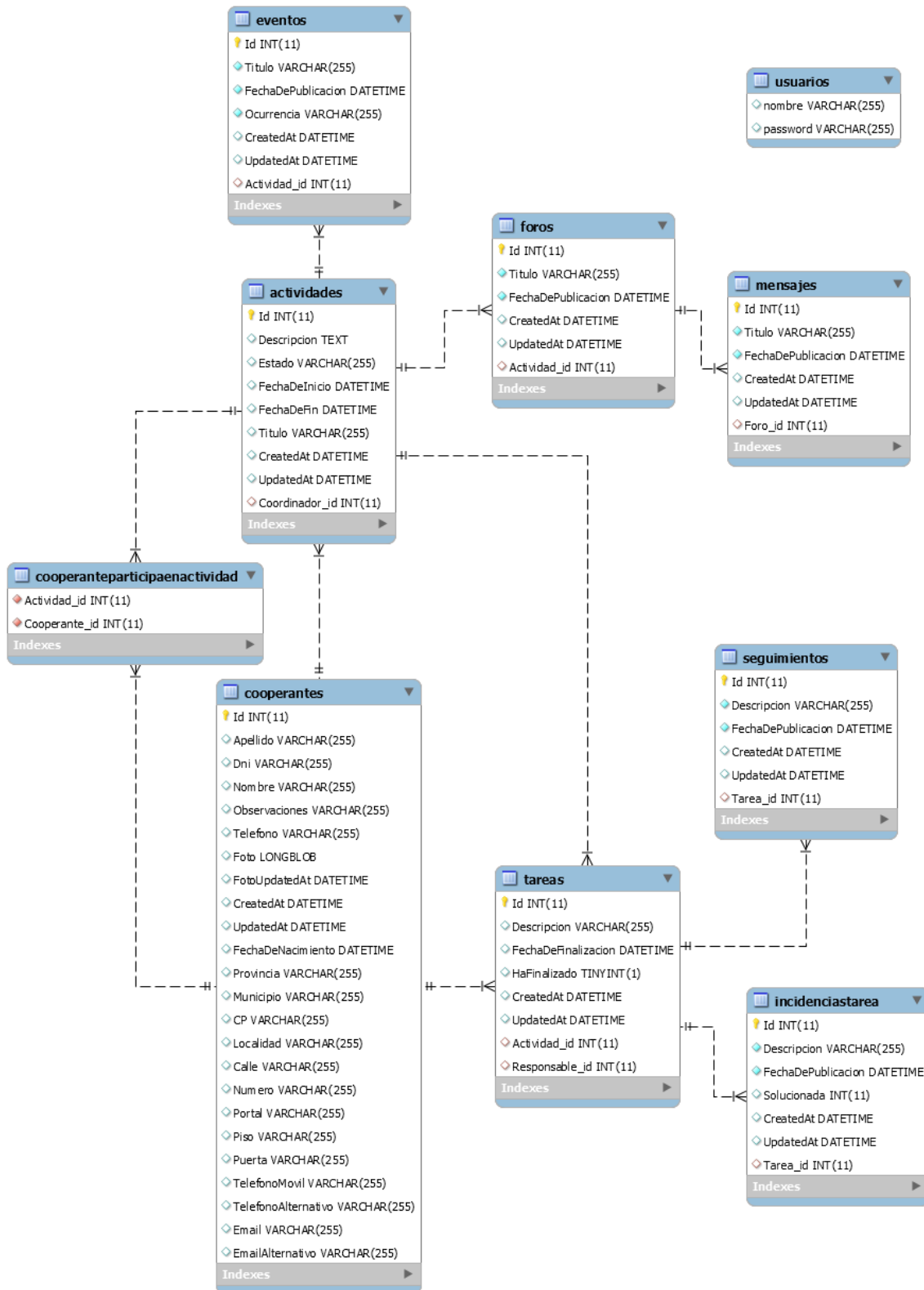


Ilustración 28. Diagrama entidad-relación de la base de datos de cooperación

## 9.12. DECISIONES DE RENDIMIENTO

Se tomaron varias decisiones relativas al rendimiento de la aplicación. Se conocía de antemano que WPF puede ser pesado en memoria. A esto hay que añadirle el uso de ciertos componentes que, si no se usan con cuidado, pueden afectar notablemente el rendimiento de la aplicación.

### 9.12.1. Sobre WPF

En relación a WPF, se hubo **que cambiar por completo el sistema navegación** llegados a un punto del desarrollo, ya que en ciertos escenarios las transiciones entre vistas podían tomar hasta varios segundos. Esto ocurría cuando usábamos la prestación de la librería Prism para navegar. Tras consultar en [su página de GitHub](#) cómo funcionaba dicho sistema de navegación y dado el problema de rendimiento, quedó cuál era el origen del problema –el sistema de navegación de Prism– y se hubo de plantear una nueva forma. Los resultados fueron muy satisfactorios, ya que se consiguió mejorar rendimiento muy notablemente.

### 9.12.2. Sobre el calendario

Aunque hoy en día pueda parecer que hay librerías gratuitas y de código libre para todo, no es así. Concretamente, no se ha podido encontrar ni una sola librería para generar calendarios que no sea de pago. Las opciones de pago ([DevExpress](#) e [Infragistics](#)) quedaron descartadas desde el primer momento, por lo que tuvimos que explorar otras opciones. Estas fueron: utilizar las características de WPF para personalizar el pequeño control de calendario propio de WPF, y desarrollar desde cero un calendario, usando para ello las características de WPF para desarrollar controles personalizados. Tras varias pruebas, se optó por la segunda opción, ya que nos ofrecía más libertad de diseño.

No obstante, hay que destacar el [excelente artículo](#) de Charles Petzold en la *Microsoft Magazine* sobre cómo personalizar el control de calendario de WPF.

### 9.12.3. Sobre el arranque de los módulos

Esta cuestión es una de las más importantes, ya que de no haberse realizado los cambios de diseño necesarios para tratar este tema, el arranque de un módulo podría llegar a tomar cerca de un minuto. Sin embargo ahora no toma más que entre unos instantes y varios segundos. Esto se ha conseguido gracias a:

- Se hacen **consultas directas** a la base de datos durante el arranque para evitar la innecesaria sobrecarga del ORM.
- Se **serializa la configuración que crea el ORM** con toda la información de mapeos (mapeos que han sido diseñados con el rendimiento en mente) y de conexión. El proceso de creación de la configuración es pesado, de ahí la conveniencia de persistirla. Esto se descubrió haciendo uso de **herramientas de análisis de rendimiento** de Visual Studio.
- Se **cachean localmente las imágenes** de la base de datos, de forma que sólo se vuelven a recoger de ésta si han sido actualizadas. Esto se debe a la sobrecarga que implica el recoger todas las imágenes en el arranque, incluso si éstas son poco pesadas (menos de 10 KB).

Además de esto, se ha considerado el impacto que tiene sobre el usuario esta espera, por lo que se ha incluido una vista de precarga a modo de *Splash Screen* que hace más agradable el arranque. De todas formas, todos los módulos están diseñados de forma que pueden permanecer abiertos indefinidamente, lo cual aumenta las opciones para el usuario.



Ilustración 29. *Splash Screen* (precarga de módulos)

#### 9.12.4. Sobre la Extranet

La dinámica de trabajo en Gamá hace necesario que se requiera poder acceder a cierta información cuando no se tiene acceso a la aplicación. No se trata esta de una cuestión de rendimiento del sistema en sí, pero sí hace que consultar datos sea más eficiente en ciertos escenarios.

---

## 10. DISEÑO DE LA INTERFAZ DE USUARIO

---

Este capítulo trata las siguientes cuestiones: Selección de módulo y control de acceso; La disposición general de regiones que comparten los tres módulos; La relación entre las distintas vistas y los requisitos que satisfacen, acompañado de capturas de pantalla de dichas vistas.

### 10.1. SELECCIÓN DE MÓDULO

El selector de módulo hace las veces de control de acceso por usuarios.



Ilustración 30. Selección de módulo

En caso de introducirse datos de acceso erróneos, se informará al usuario como se muestra a continuación, con un mensaje de error que desaparece al cabo de unos segundos:

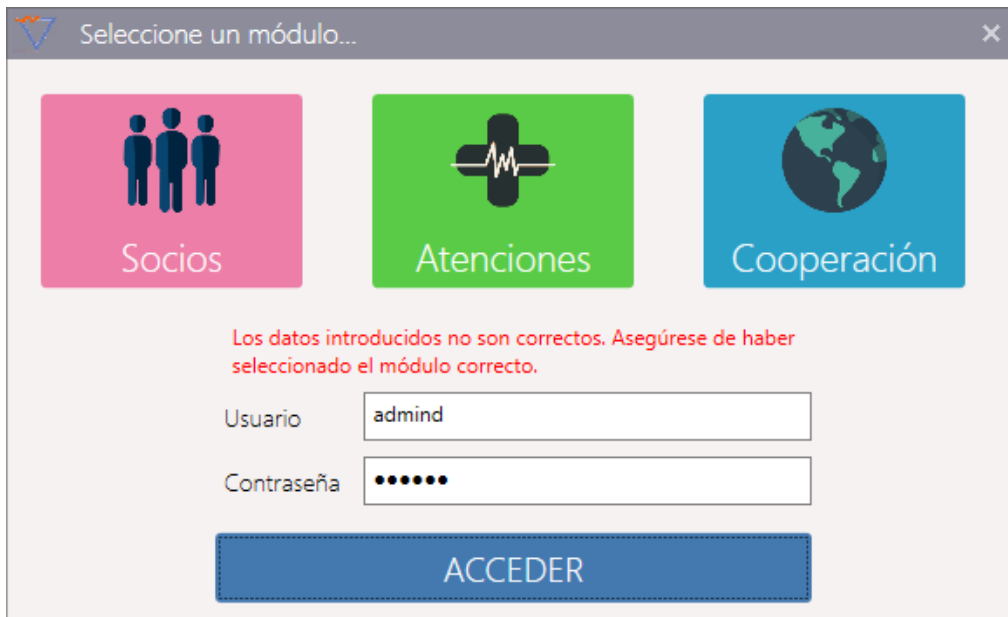


Ilustración 31. Login incorrecto en la selección de módulo

## 10.2. ESQUEMA VISUAL COMPARTIDO

Los tres módulos comparten un mismo esquema visual:



Tabla 3. Layout base para los tres módulos\*

---

(1)	<ul style="list-style-type: none"> <li>▪ Icono del programa</li> <li>▪ Accesos directos al sitio Web del colectivo, su página en Facebook y su página en Twitter</li> <li>▪ Título del módulo</li> </ul>
(2)	<ul style="list-style-type: none"> <li>▪ Funciones para cambio de módulo</li> <li>▪ Acceso a preferencias</li> <li>▪ Rótulo con el estado de conexión al servidor de sincronización</li> </ul>
(3)	<ul style="list-style-type: none"> <li>▪ Panel de navegación</li> </ul>
(4)	<ul style="list-style-type: none"> <li>▪ Cuadro de búsqueda</li> </ul>
(5)	<ul style="list-style-type: none"> <li>▪ Barra de herramientas (<i>Toolbar</i>)</li> </ul>
(6)	<ul style="list-style-type: none"> <li>▪ Sección dinámica de contenido. El contenido cambia según el panel al que se acceda desde el panel de navegación o navegando tras realizar una acción.</li> </ul>
(7)	<ul style="list-style-type: none"> <li>▪ Barra de estado donde notificar mensajes al usuario cuando se ha completado una acción con éxito.</li> </ul>

---

*\* Se ha adoptado el patrón *Balanced Whitespace* para tratar el tema de los márgenes. Seguir un orden en esta cuestión otorga uniformidad y facilita la modificación de vistas, ya que todas siguen una misma filosofía de márgenes. Esta consiste en asignar márgenes solamente a los controles hoja del árbol visual, es decir, a los controles en sí y no a los contenedores de otros controles.*

## 10.3. ASPECTOS VISUALES COMUNES

Algunos aspectos relacionados al diseño de la interfaz son compartidos por todos los módulos.

### 10.3.1. Seguimiento de cambios

Cuando se introduce un cambio que aún no ha sido persistido, el color de fondo del campo se ilumina para indicarlo. El valor original se indica en el *tooltip* que aparece al pasar el ratón por encima del campo. Si se cancela la edición, los valores vuelven a su estado original.

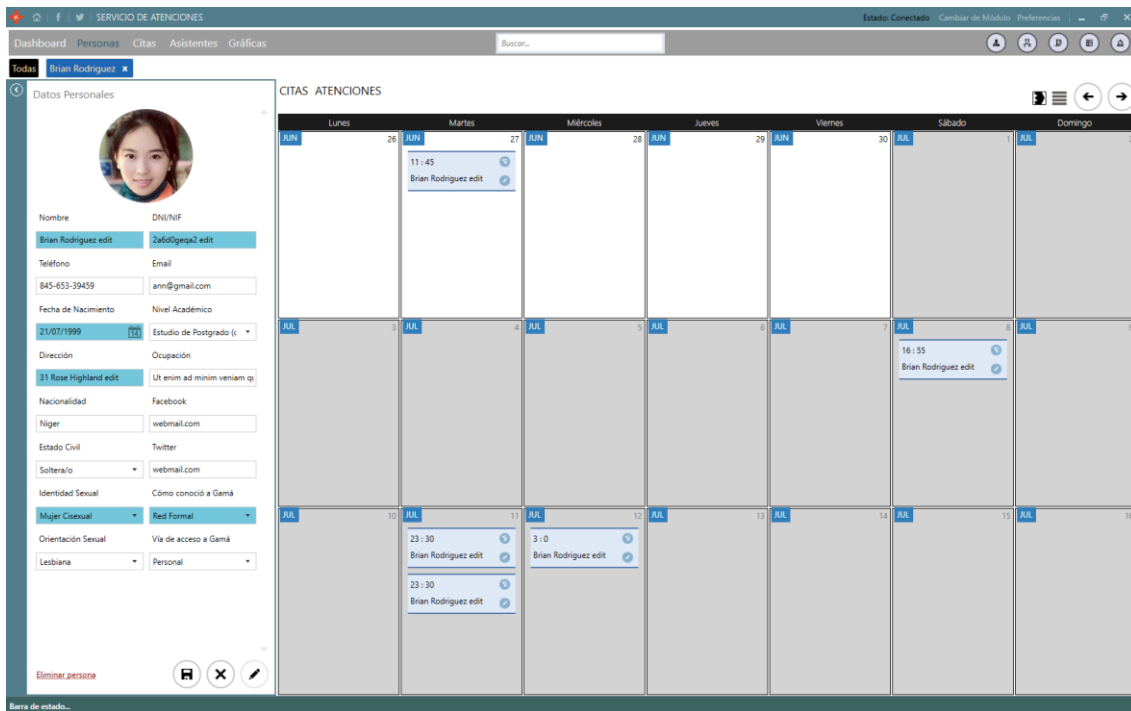


Ilustración 32. Seguimiento de cambios

Si se trata de cerrar una pestaña cuando hay cambios sin guardar, se notifica al usuario de si desea continuar con la operación:

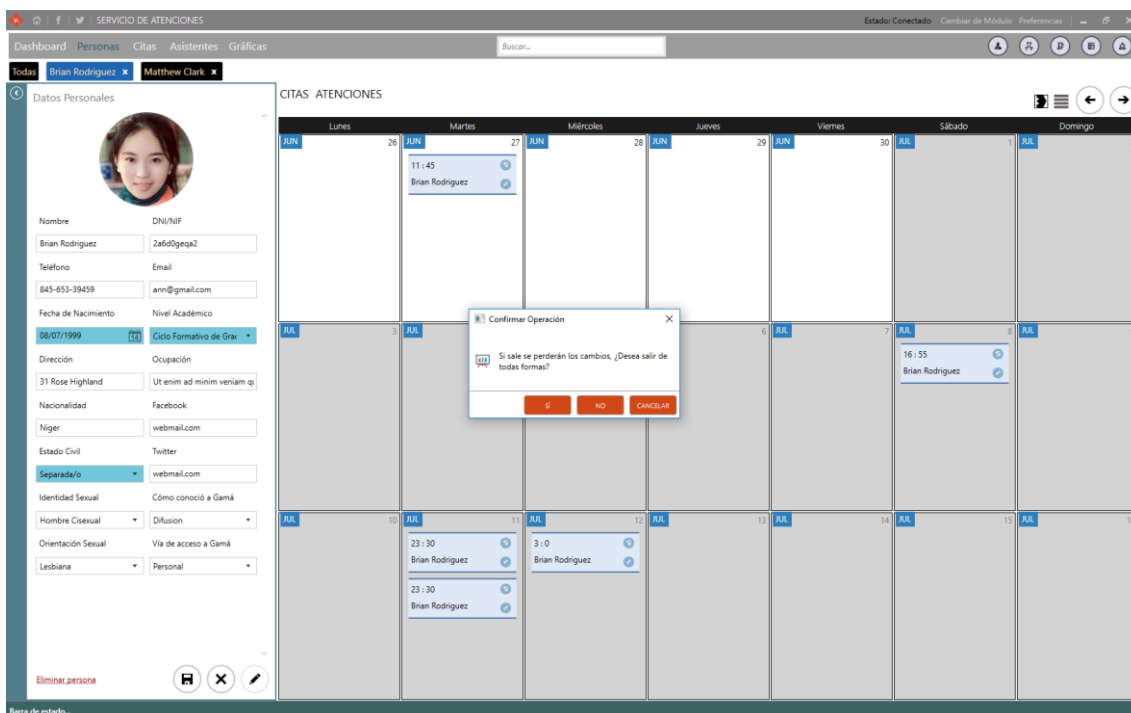


Ilustración 33. Confirmar operación

### 10.3.2. Validación

La validación tiene en cuenta errores como un NIF repetido, un campo obligatorio vacío, una fecha en formato incorrecto o un email inválido. Se muestra a continuación un caso a modo ilustrativo.

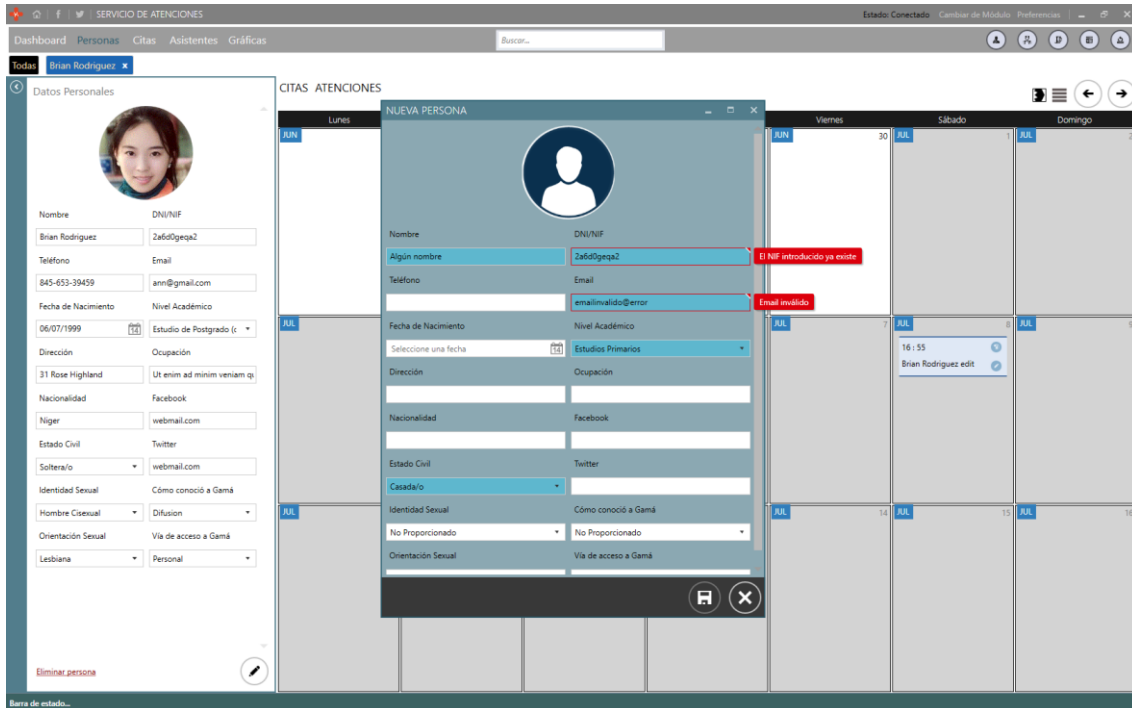


Ilustración 34. Validación de datos

### 10.3.3. Habilitación de botones

Las acciones que requiere de algún tipo de validación no se podrán llevar a cabo si el estado actual no es válido. Esto es llevado a la interfaz en forma de inhabilitación de los botones involucrados, mostrándolos con una opacidad menor. A continuación se muestra tal transición.



NUEVO ASISTENTE



⏏
✕

Ilustración 35. Botón de guardar inhabilitado

NUEVO ASISTENTE



⏏
✕

Ilustración 36. Botón de guardar habilitado

### 10.3.4. Notificar a través del Status Bar

La barra de estado se ilumina durante unos segundos cuando una acción que ha persistido la base de datos tiene lugar.

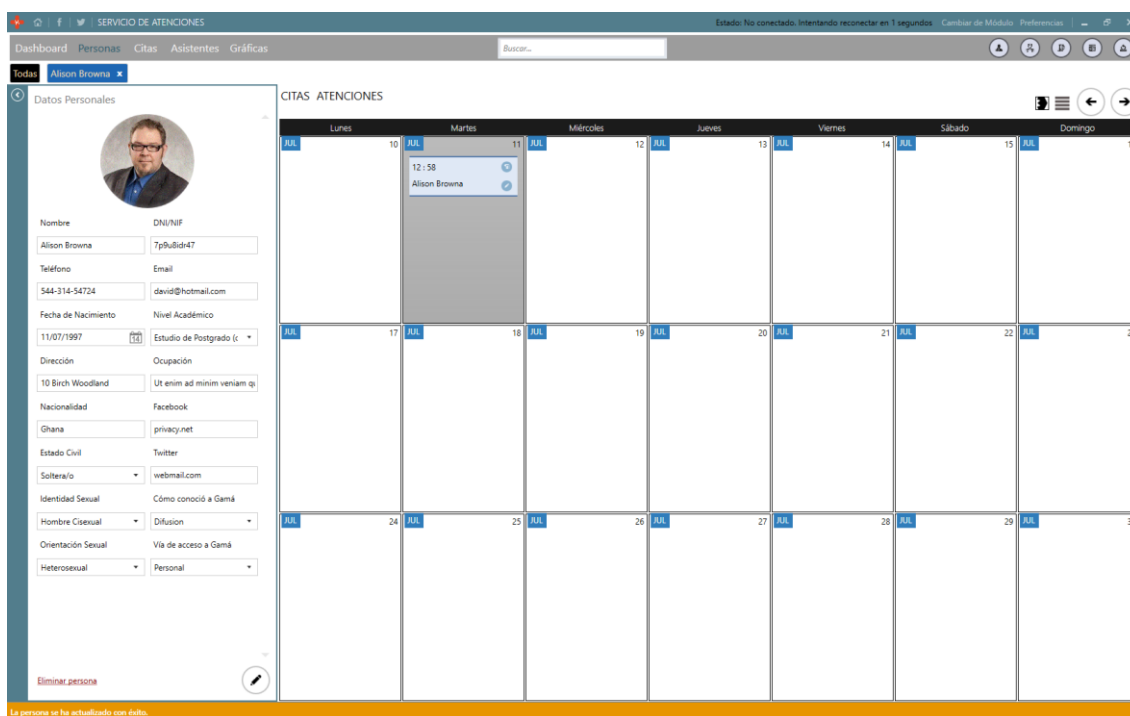


Ilustración 37. Notificación mediante la barra de estado

## 10.4. DISEÑO DE LA INTERFAZ DEL MÓDULO DE ATENCIONES

En esta sección se incluye una relación entre todas las vistas del módulo de atenciones y los requisitos de usuario que satisfacen. Se adjunta una captura de pantalla cada vista a continuación.

VISTA	REQUISITOS QUE CUBRE
Borde superior de la ventana	Acciones para acceder a los sitios Web de Gamá y para volver al selector de módulo. Ver estado de conexión con el servidor de sincronización.
Panel de navegación	Navegar a los distintos paneles
Cuadro de búsqueda	Buscar persona, navegar a persona
Barra de herramientas	Añadir persona, añadir asistente, exportar persona, exportar listado, hacer copia de seguridad, restaurar copia de seguridad
Barra de estado	Notificar acciones completadas exitosamente
Dashboard	Visualizar personas, visualizar citas, visualizar atenciones, filtrar listados por fecha, filtrar por persona seleccionada, navegar a persona,

	navegar a cita, navegar a atención
<b>Panel de personas</b>	
<b>Listado</b>	Visualizar personas, navegar a persona
<b>Personas individuales</b>	Editar persona, editar cita, editar atención, visualizar citas de una persona, visualizar atenciones de una persona , eliminar persona (con sus citas y atenciones)
<b>Panel de citas</b>	Visualizar citas, crear cita, editar cita, navegar a persona, navegar a cita, navegar a atención
<b>Panel de asistentes</b>	Visualizar asistente, visualizar asistentes, editar asistente, editar cita, navegar a cita, navegar a persona
<b>Gráficas</b>	Visualizar gráficas
<b>Preferencias</b>	Modificar preferencias de usuario

### 10.4.1. Dashboard

El dashboard sirve como vista global donde poder acceder de forma rápida a cualquier elemento, sea buscando por nombre a la persona, buscando el elemento en alguna de las listas o filtrando por fecha todos los datos mostrados.

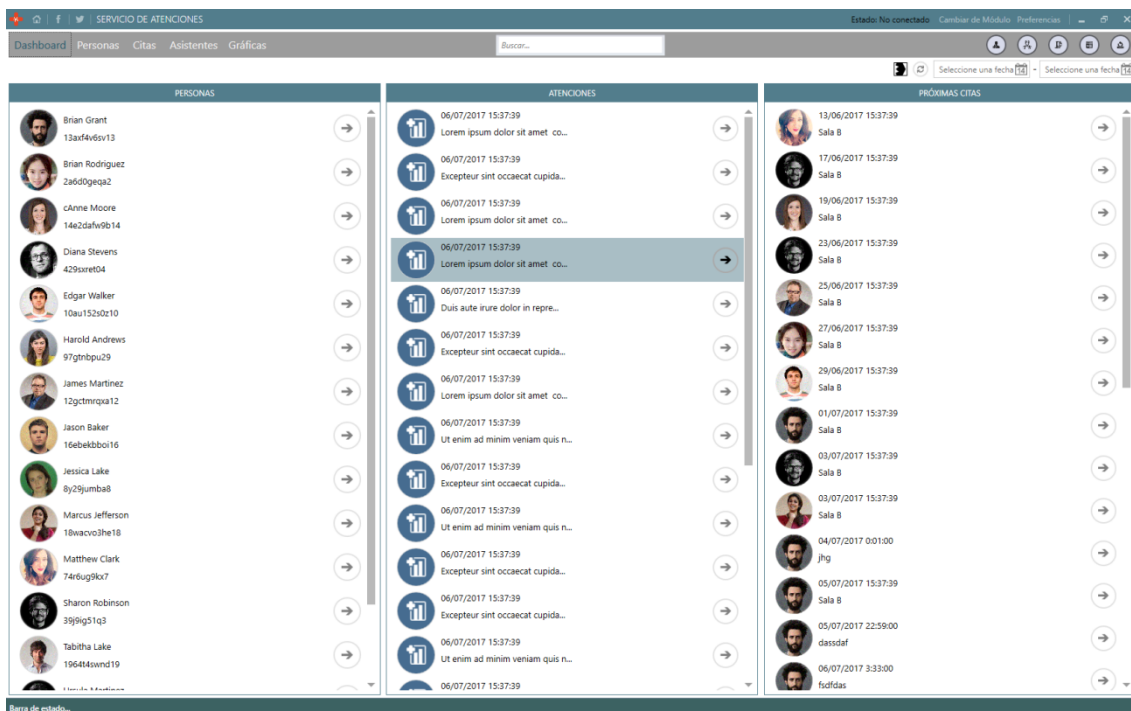


Ilustración 38. Dashboard

SERVICIO DE ATENCIONES

Estado: No conectado. Intentando reconectar en 2 segundos. Cambiar de Módulo. Preferencias.

Dashboard Personas Citas Asistentes Gráficas

Buscar...

PERSONAS

Brian Grant  
13ad460v13

ATENCIÓNES

06/07/2017 15:37:39  
Lorem ipsum dolor sit amet co...

06/07/2017 15:37:39  
Duis aute irure dolor in repre...

PRÓXIMAS CITAS

01/07/2017 15:37:39  
Sala B

04/07/2017 00:1:00  
j'jg

05/07/2017 15:37:39  
Sala B

05/07/2017 22:59:00  
classdaf

06/07/2017 3:33:00  
foolfdas

Barra de estado...

Ilustración 39. Dashboard aplica filtro al seleccionar una persona de la lista

## 10.4.2. Cuadro de búsqueda

El cuadro de búsqueda permite buscar a una persona por nombre. Aunque se muestran resultados limitados en el cuadro de resultados, el conjunto de resultados se muestran en la lista de personas. Durante la búsqueda sólo se ven las personas del conjunto filtrado.

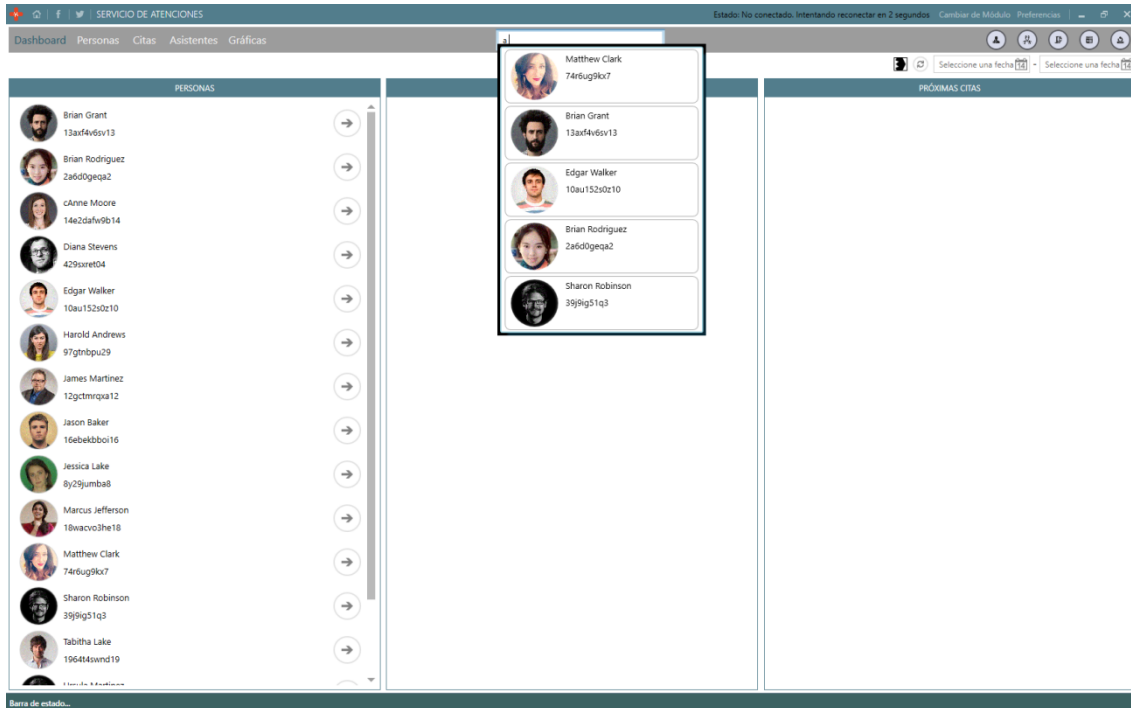


Ilustración 40. Cuadro de búsqueda afecta a los datos mostrados

Cuando sólo hay un resultado, se autoselecciona la persona, propagando el filtro hacia las citas y atenciones también.

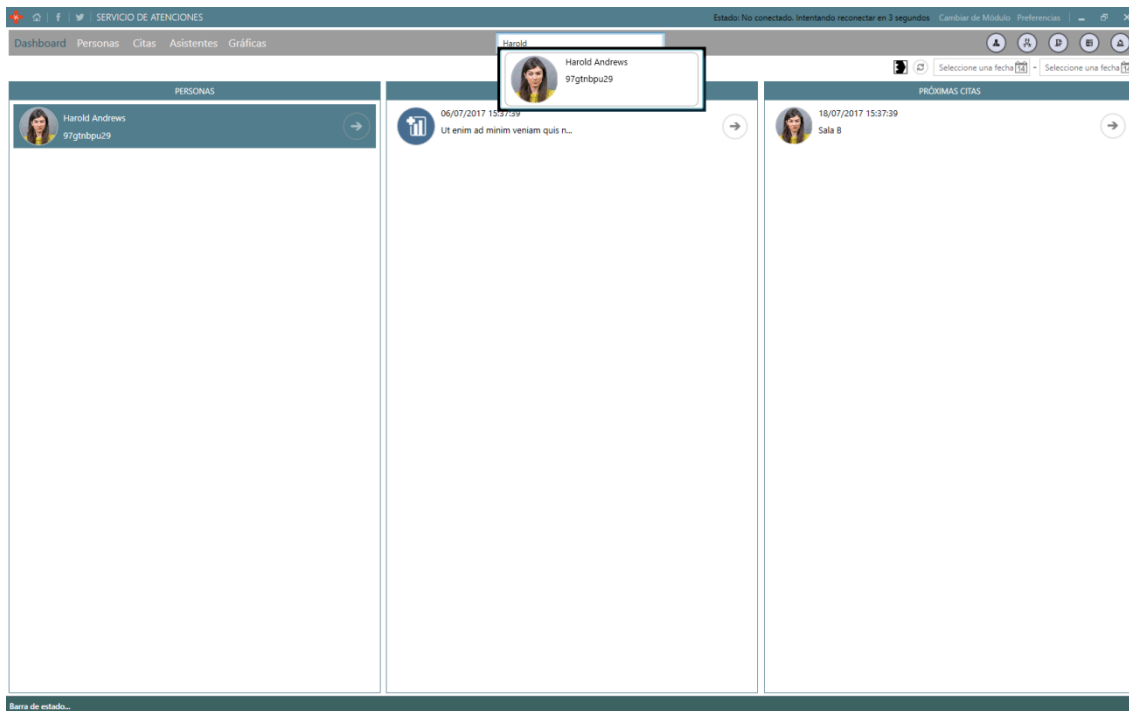


Ilustración 41. Cuadro de búsqueda con un sólo resultado

### 10.4.3. Acciones de la barra de herramientas

La barra de herramientas permite el acceso rápido a algunas funciones generales como añadir personas y atenciones, exportar, o hacer y recuperar copias de seguridad. Sólo se muestran las pantallas de las dos primeras pues las otras son acciones sin vista asociada.

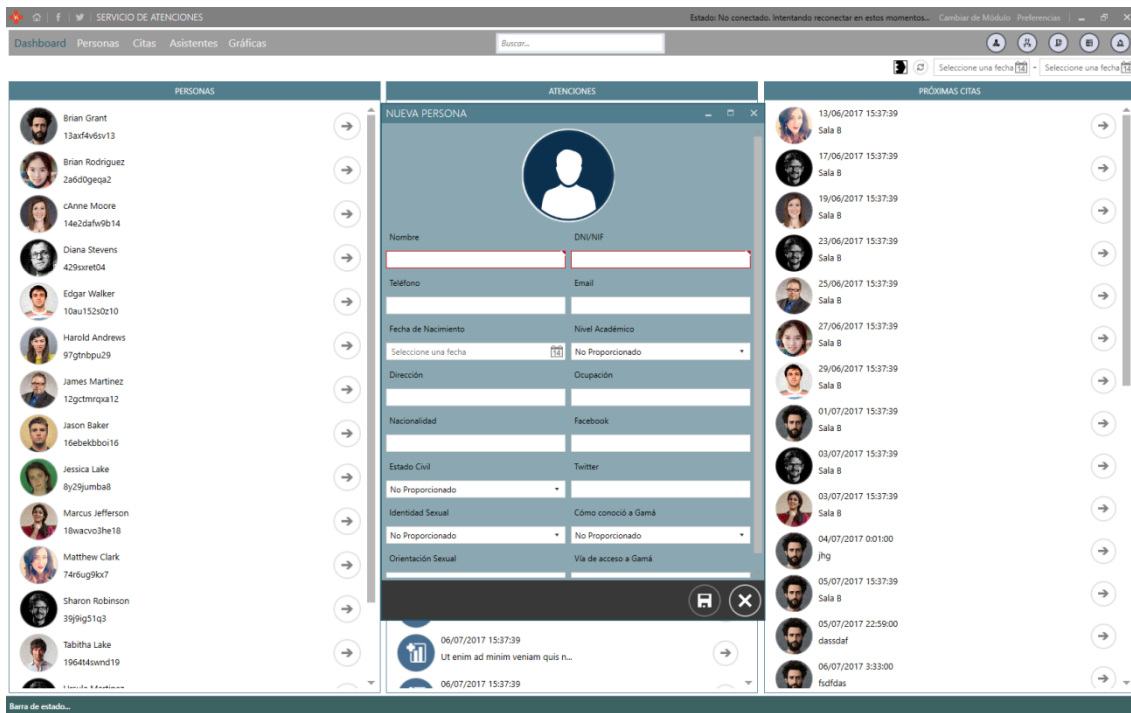


Ilustración 42. Añadir una nueva persona

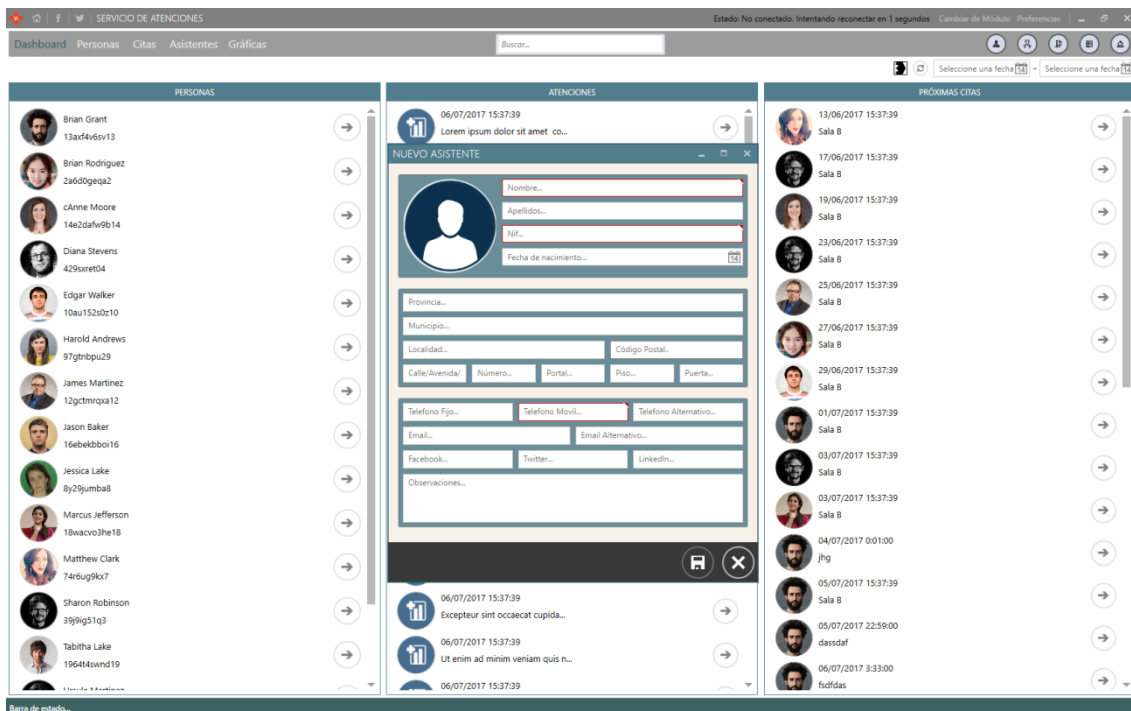


Ilustración 43. Añadir un nuevo asistente

#### 10.4.4. Panel de personas

El panel de persona se trata de un control de pestañas en el que siempre está incluida una primera pestaña con un listado paginado de todas las personas, junto a una pestaña más por cada persona que se haya abierto. La vista de la edición de la persona

cuenta con dos sub-vistas: la edición personal de citas (tanto en formato de calendario como tabular) y la edición de atenciones.

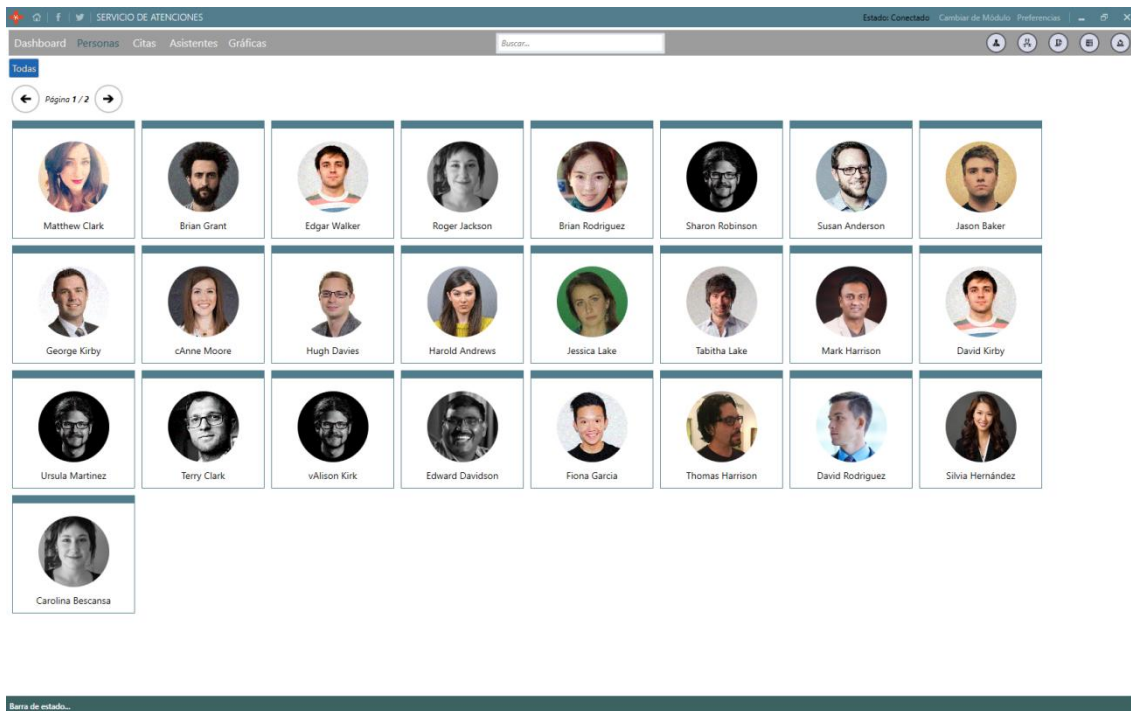


Ilustración 44. Listado paginado de personas

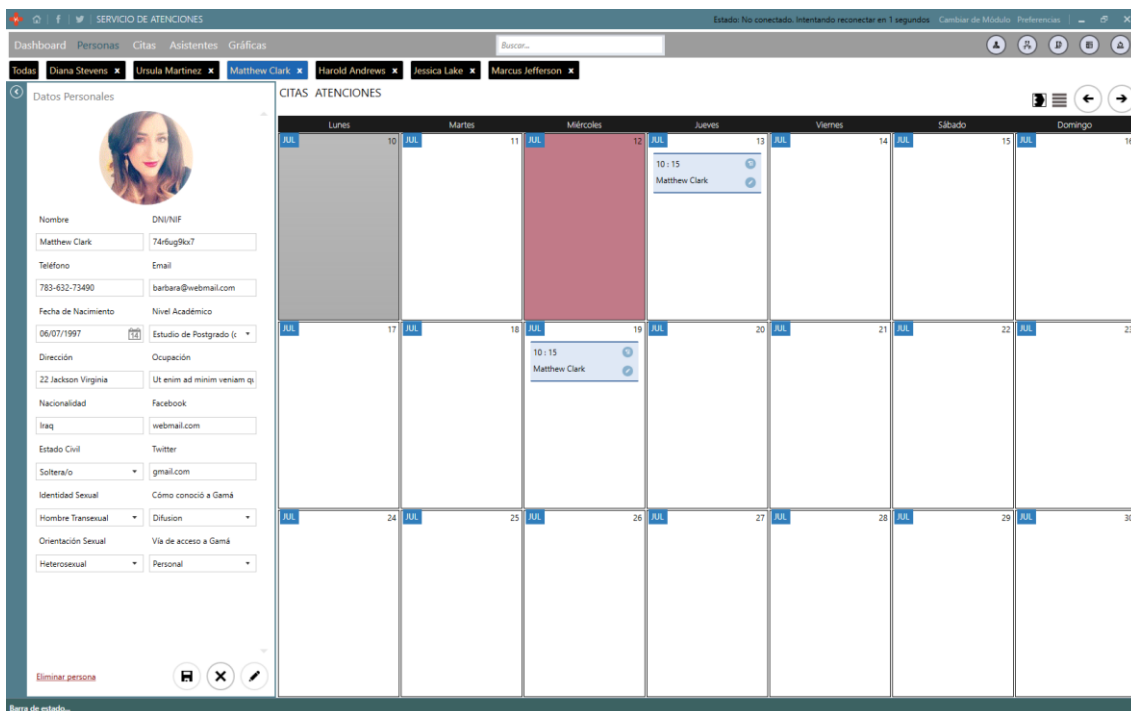


Ilustración 45. Edición de citas de una persona



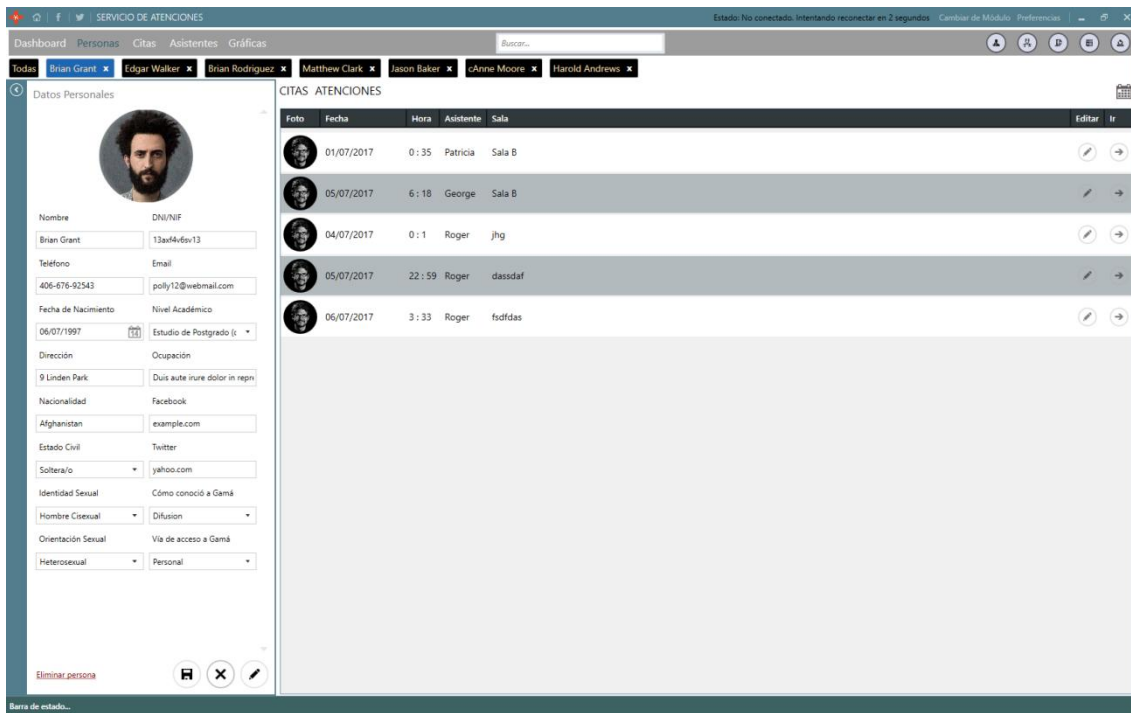


Ilustración 46. Edición de citas en formato tabular

En ocasiones conviene disponer de más espacio para ver el calendario. Por ello, la información personal se puede ocultar, como se muestra a continuación:

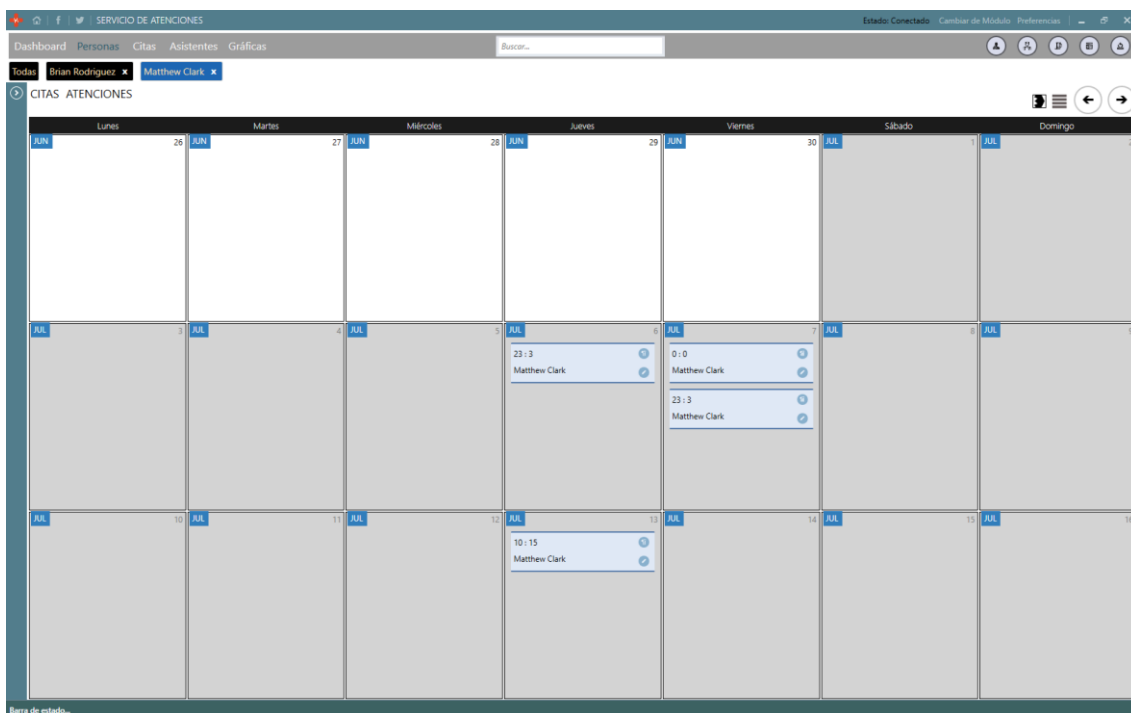


Ilustración 47. Información de persona oculta

A continuación se muestra uno de los múltiples sitios desde donde se puede editar una cita:

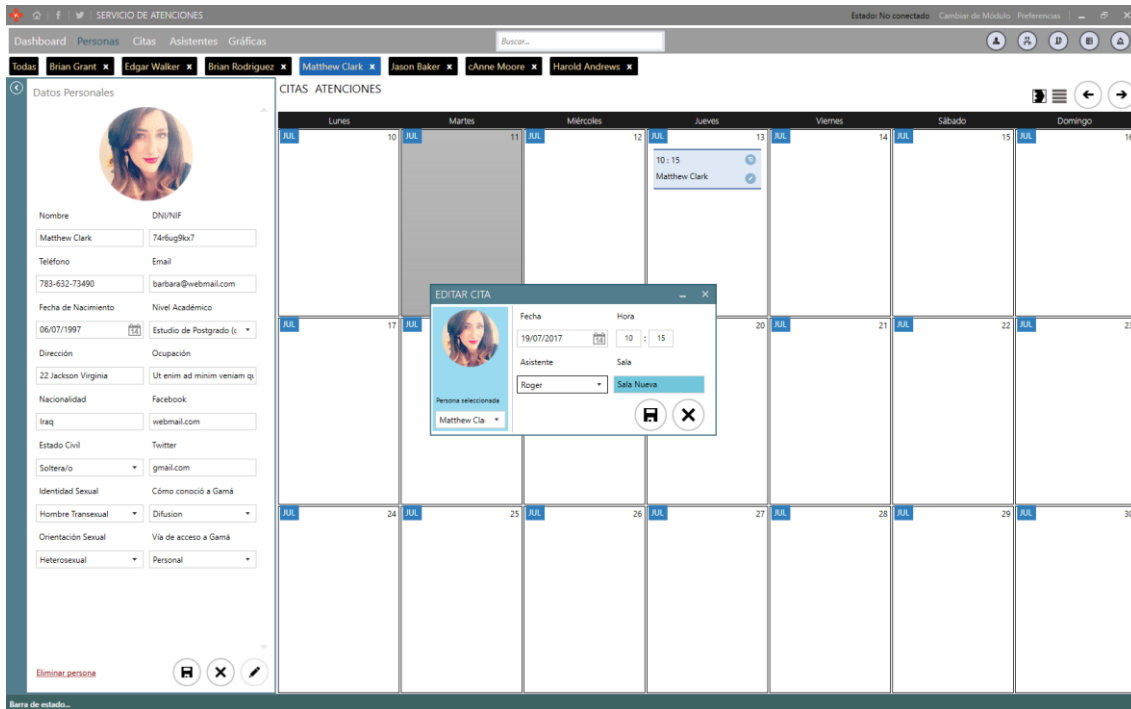


Ilustración 48. Editar una cita

Una atención es un pequeño informe asociado a una cita, de ahí que desde la atención podamos también ver y editar la información de la cita:

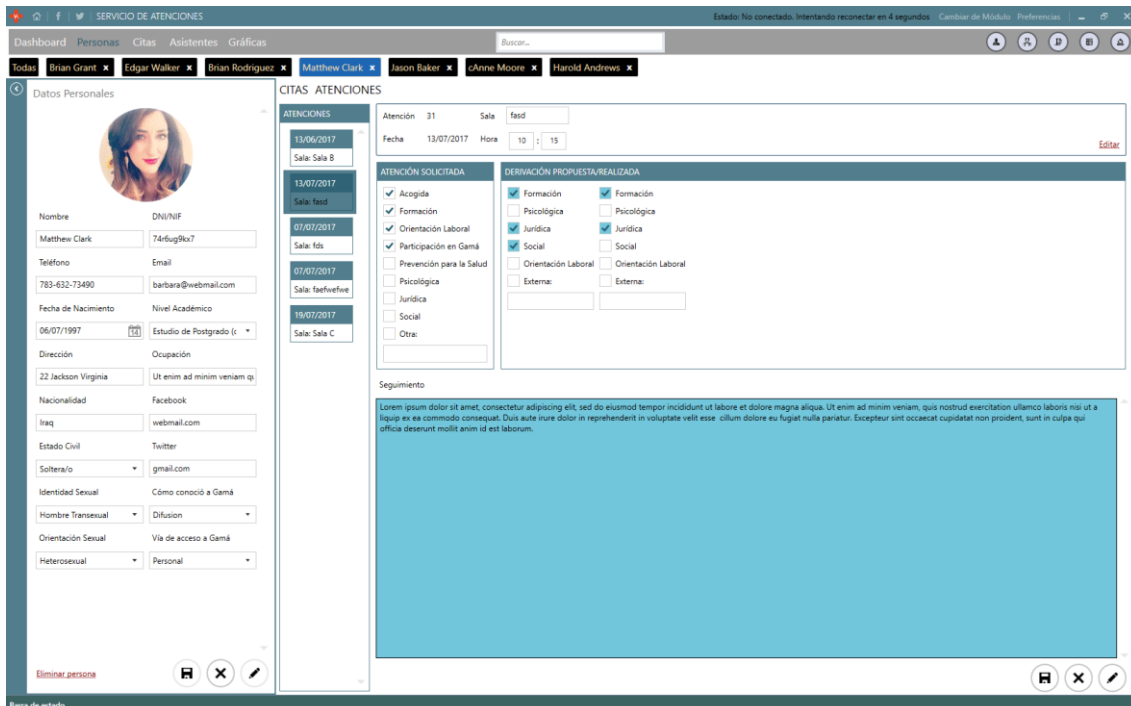


Ilustración 49. Edición de atenciones

### 10.4.5. Panel de citas

El calendario muestra todas las citas contenidas en el sistema. Como todos los demás calendarios, es navegable y permite cambiar entre formato de calendario y formato tabular, así como filtrar por fecha en este último formato.

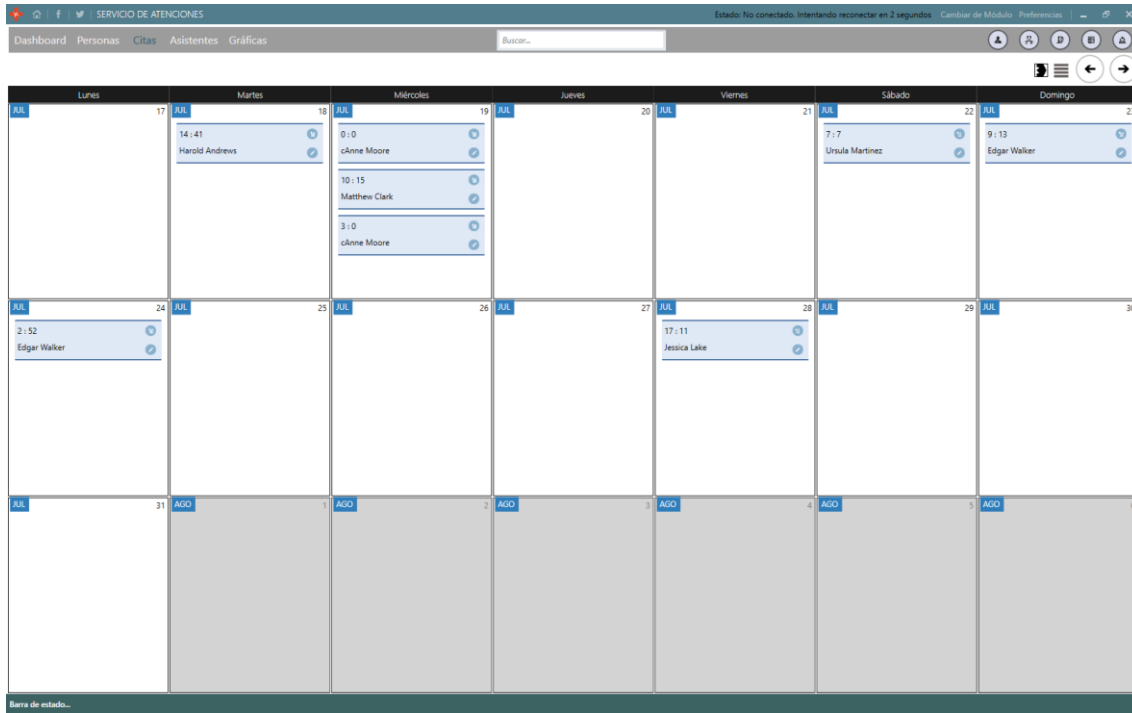


Ilustración 50. Panel de citas (vista de calendario)

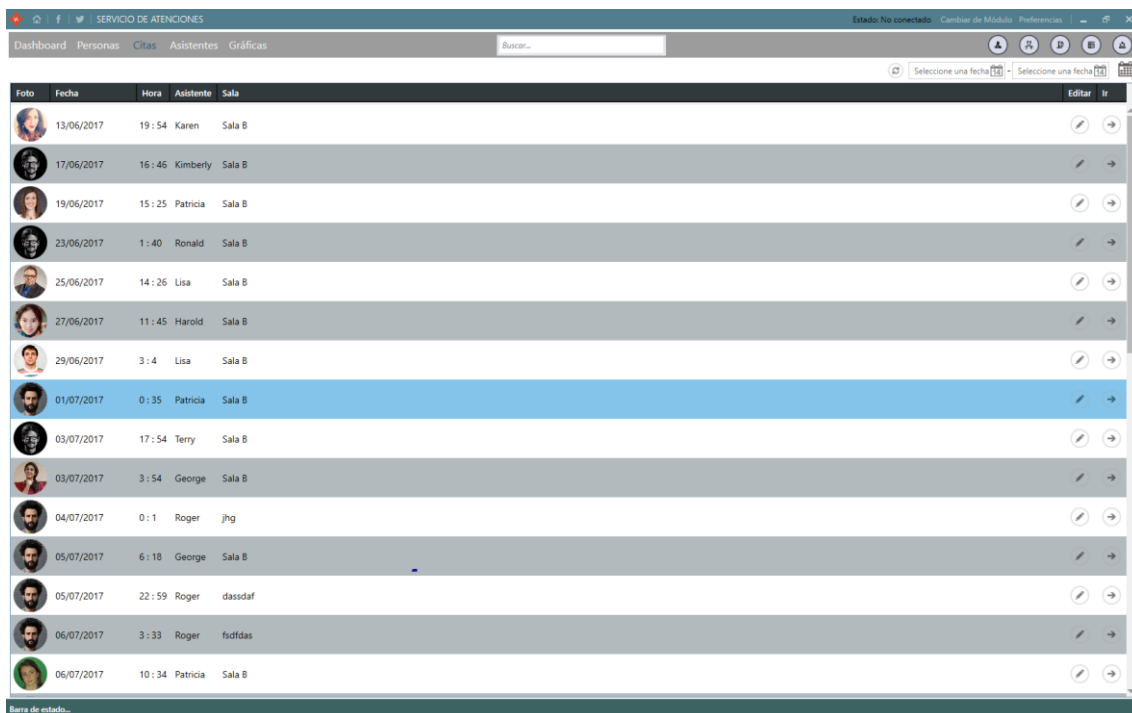


Ilustración 51. Panel de citas (vista tabular)

Foto	Fecha	Hora	Asistente	Sala
	04/07/2017	0:1	Roger	jhg
	05/07/2017	6:18	George	Sala B
	05/07/2017	22:59	Roger	dassdaf
	06/07/2017	3:33	Roger	fsdfdas
	06/07/2017	10:34	Patricia	Sala B
	06/07/2017	23:3	Roger	fasdf
	07/07/2017	0:0	Roger	fds

Ilustración 52. Panel de citas (vista tabular filtrada)

### 10.4.6. Panel de asistentes

Los asistentes no suelen ser muy numerosos. Por ello, en el panel de asistentes se muestra un listado navegable de asistentes junto a toda la información del asistente seleccionado en un momento dado. Esta información incluye todas las citas pasadas y futuras asociadas a él.

**Listado de Asistentes**

- Roger Davidson
- Roger Clarkson
- Terry Anthony
- Ronald Lewis
- Michael Grant
- Harold Walker
- Kimberly Andrews
- Lisa Peters
- Karen Major
- Marcus Andrews
- Karen Major
- Anthony Moore
- Patricia Anthony
- George Smythe
- Paul Thompson

**Datos Personales**

Roger Davidson

Oybušw4560

06/07/1989

24 Adams Highland

12 Washington Park

20 Wilson Park

Código Postal...

Calle/Avenida/... Número... Portal... Piso... Puerta...

683-732-55624 098-214-00089 Telefono Alternativo...

james@yahoo.com Email Alternativo...

msn.com webmail.com gmail.com

Nuevas observaciones no guardadas aún...

**Próximas Citas**

FOTO	NOMBRE	FECHA	HORA	SALA	EDITAR	IR
	Matthew Clark	13/07/2017	10:15	fasd		
	Matthew Clark	19/07/2017	10:15	Sala C		
	cAnne Moore	11/07/2017	0:0	fasd		
	Brian Rodriguez	11/07/2017	23:30	fsdf		
	Brian Rodriguez	11/07/2017	23:30	fsdf		
	cAnne Moore	13/07/2017	0:0	fasdf		
	cAnne Moore	19/07/2017	0:0	fasd		
	Diana Stevens	13/07/2017	23:0	f		
	Brian Rodriguez	12/07/2017	3:0	fasd		
	Matthew Clark	13/07/2017	10:15	fasd		
	Matthew Clark	19/07/2017	10:15	Sala C		
	cAnne Moore	11/07/2017	0:0	fasd		
	Brian Rodriguez	11/07/2017	23:30	fsdf		

**Citas Pasadas**

FOTO	NOMBRE	FECHA	HORA	SALA	EDITAR	IR
	Brian Grant	04/07/2017	0:1	jhg		
	Brian Grant	05/07/2017	22:59	dassdaf		
	Matthew Clark	07/07/2017	0:0	fds		
	Matthew Clark	06/07/2017	23:3	fasdf		
	Matthew Clark	07/07/2017	23:3	faefwefwe		
	Brian Grant	06/07/2017	3:33	fsdfdas		
	Brian Grant	04/07/2017	0:1	jhg		
	Brian Grant	05/07/2017	22:59	dassdaf		
	Matthew Clark	07/07/2017	0:0	fds		
	Matthew Clark	06/07/2017	23:3	fasdf		
	Matthew Clark	07/07/2017	23:3	faefwefwe		
	Brian Grant	06/07/2017	3:33	fsdfdas		

Ilustración 53. Listado de asistentes

### 10.4.7. Panel de gráficas

Al igual que con el panel de asistentes, se ha optado por dotar de un listado navegable (con vistas a extenderse en el futuro) para mostrar los dos conjuntos de gráficas siguientes:

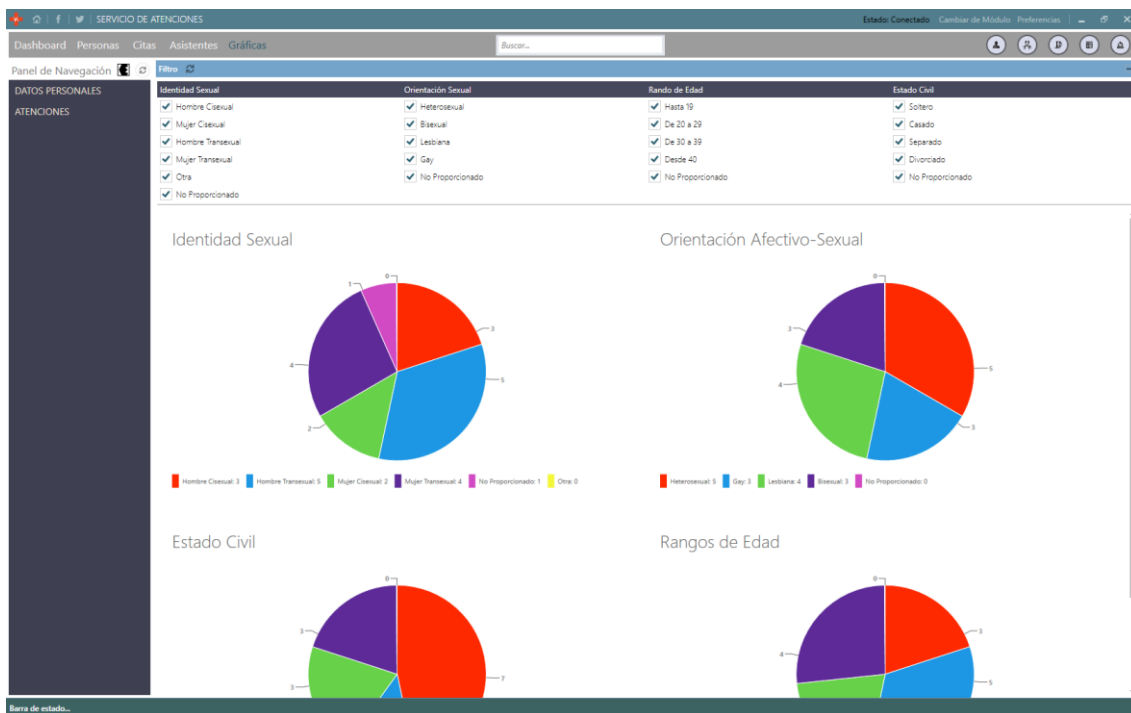


Ilustración 54. Gráficas de datos personales con el filtro visible

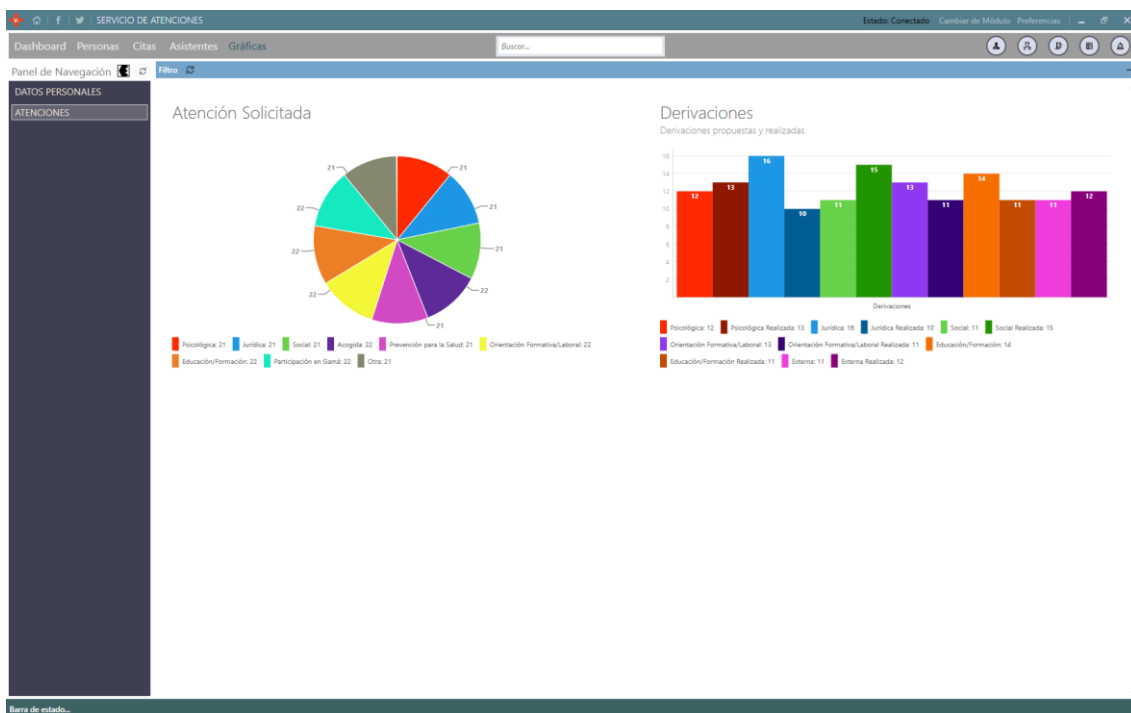


Ilustración 55. Gráficas de atenciones con el filtro oculto

## 10.4.8. Preferencias

En las preferencias se configuran algunos aspectos de la interfaz de usuario. También es aquí donde se configuran las opciones de la gestión automática de copias de seguridad.

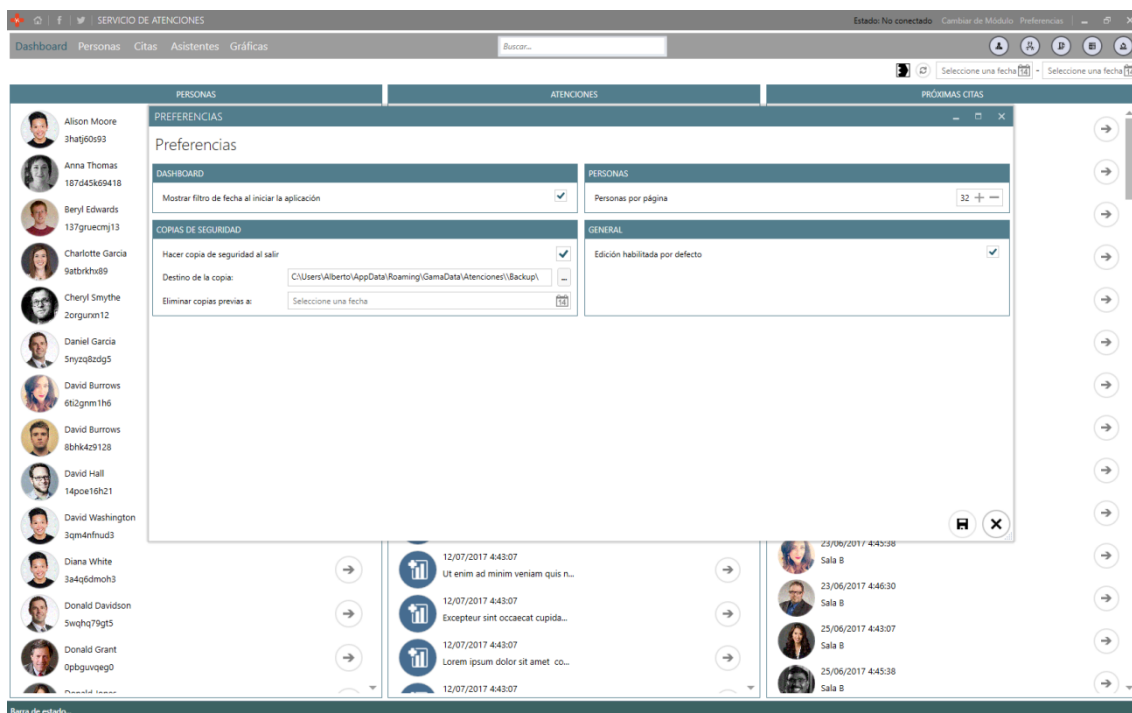


Ilustración 56. Ventana de preferencias

## 10.5. DISEÑO DE LA INTERFAZ DEL MÓDULO DE SOCIOS

En esta sección se incluye una relación entre todas las vistas del módulo de socios y los requisitos de usuario que satisfacen. Se adjunta una captura de pantalla cada vista a continuación.

VISTA	REQUISITOS QUE CUBRE
Borde superior de la ventana	<ul style="list-style-type: none"> <li>Acciones para acceder a los sitios Web de Gamá y para volver al selector de módulo.</li> <li>Ver estado de conexión con el servidor de sincronización.</li> <li>Cambiar de módulo.</li> <li>Mostrar el cuadro de preferencias.</li> </ul>
Panel de navegación	<ul style="list-style-type: none"> <li>Navegar a los distintos paneles</li> </ul>
Cuadro de búsqueda	<ul style="list-style-type: none"> <li>Buscar socio.</li> <li>Navegar al socio seleccionado</li> </ul>
Barra de herramientas	<ul style="list-style-type: none"> <li>Añadir socio,.</li> <li>Exportar socio o lista de socios.</li> <li>Hacer copia de seguridad.</li> <li>Restaurar copia de seguridad</li> </ul>
Barra de estado	<ul style="list-style-type: none"> <li>Notificar acciones completadas exitosamente</li> </ul>
Dashboard	<ul style="list-style-type: none"> <li>Visualizar lista de socios y editar la información de cada socio.</li> <li>Navegar a cada socio.</li> <li>Filtrar listados de socios.</li> <li>Mostrar estado general contable.</li> <li>Mostrar lista de socios con impagos.</li> </ul>
Panel de Socios	
Listado socios	<ul style="list-style-type: none"> <li>Visualizar lista completa de socios y navegar a un socio seleccionado</li> </ul>
Socio seleccionado	<ul style="list-style-type: none"> <li>Mostrar datos personales del socio y permitir modificarlos.</li> <li>Crear los periodos de alta con sus cuotas.</li> <li>Mostrar cada una de las cuotas y permitir modificarlas.</li> <li>Mostrar número de cuotas y cantidad por pagar.</li> <li>Mostrar número de cuotas y cantidad impagada.</li> </ul>
Gráficas	<ul style="list-style-type: none"> <li>Visualizar gráficas.</li> </ul>

### 10.5.1. Dashboard.

El Dashboard sirve como vista global donde poder acceder de forma rápida a cualquier elemento, sea buscando por nombre a cualquier socio, buscando a los socios en la lista principal. Se puede filtrar a los socios por tres categorías, edad, nacionalidad o estado de alta.

Podemos realizar las tareas básicas desde esta vista, podemos modificar la información personal de cada socio, como navegar hacia la vista que nos muestra todos los datos personales del socio seleccionado así como el estado de sus cuentas.

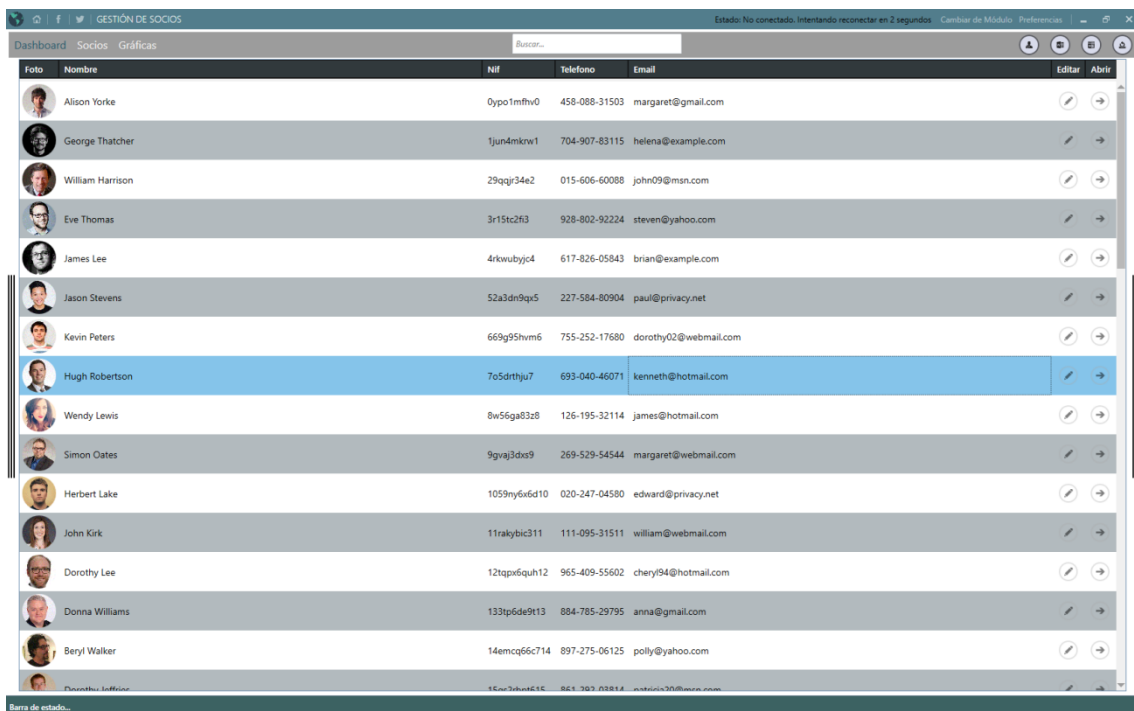


Foto	Nombre	Nif	Telefono	Email	Editar	Abrir
	Alison Yorke	0ypo1mfhv0	458-088-31503	margaret@gmail.com		
	George Thatcher	1jun4mkrw1	704-907-83115	helena@example.com		
	William Harrison	29qqj34e2	015-606-60088	john09@msn.com		
	Eve Thomas	3r15tc2f3	928-802-92224	steven@yahoo.com		
	James Lee	4rkwubjy4	617-826-05843	brian@example.com		
	Jason Stevens	52a3dn9qs5	227-584-80904	paul@privacy.net		
	Kevin Peters	669g95hvm6	755-252-17680	dorothy02@webmail.com		
	Hugh Robertson	7o5drthju7	693-040-46071	kenneth@hotmail.com		
	Wendy Lewis	8w56ga83z8	126-195-32114	james@hotmail.com		
	Simon Oates	9qvaj3dks9	269-529-54544	margaret@webmail.com		
	Herbert Lake	1059ny6vd10	020-247-04580	edward@privacy.net		
	John Kirk	11rakybic311	111-095-31511	william@webmail.com		
	Dorothy Lee	12tqp6quh12	965-409-55602	chery94@hotmail.com		
	Donna Williams	133tp6de9t13	884-785-29795	anna@gmail.com		
	Beryl Walker	14emcq66c714	897-275-06125	polly@yahoo.com		

Ilustración 22. Dashboard general.

La lista completa de socios que se muestra en el Dashboard se puede filtrar según tres categorías:

- Edad: Con rangos de filtrados son menos de 25 años, entre 26 y 40 años, entre 41 y 55 años, entre 56 y 65 años y mayores de 65 años.
- Nacionalidad: Generar una lista con los socios de una concreta nacionalidad.
- Estado de alta como socio.



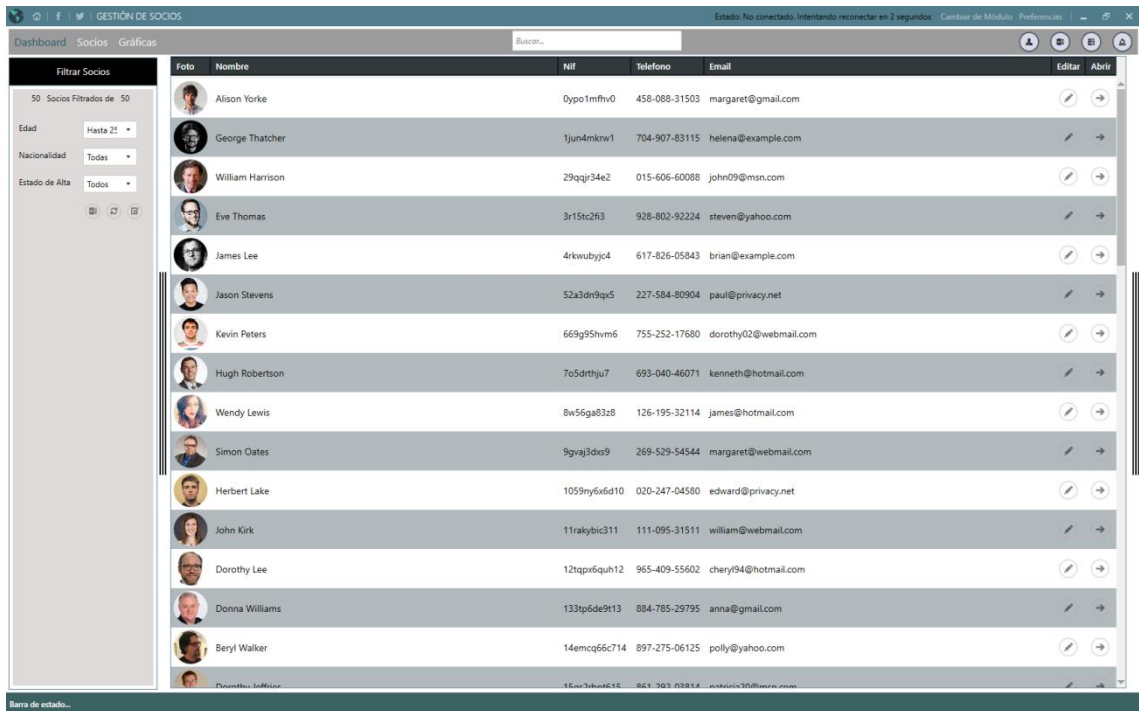


Ilustración 23. Dashboard con filtrado de socios.

En el Dashboard también se muestra dos apartados más de gran importancia. Se generan unos datos contables generales que permiten con un vistazo rápido conocer el estado de las cuentas. Muestra el número total de cuotas con la cantidad total de las cuotas pagadas, de las que están por pagar según los periodos de alta de los socios y las cuotas que están impagadas por fuera de plazo.

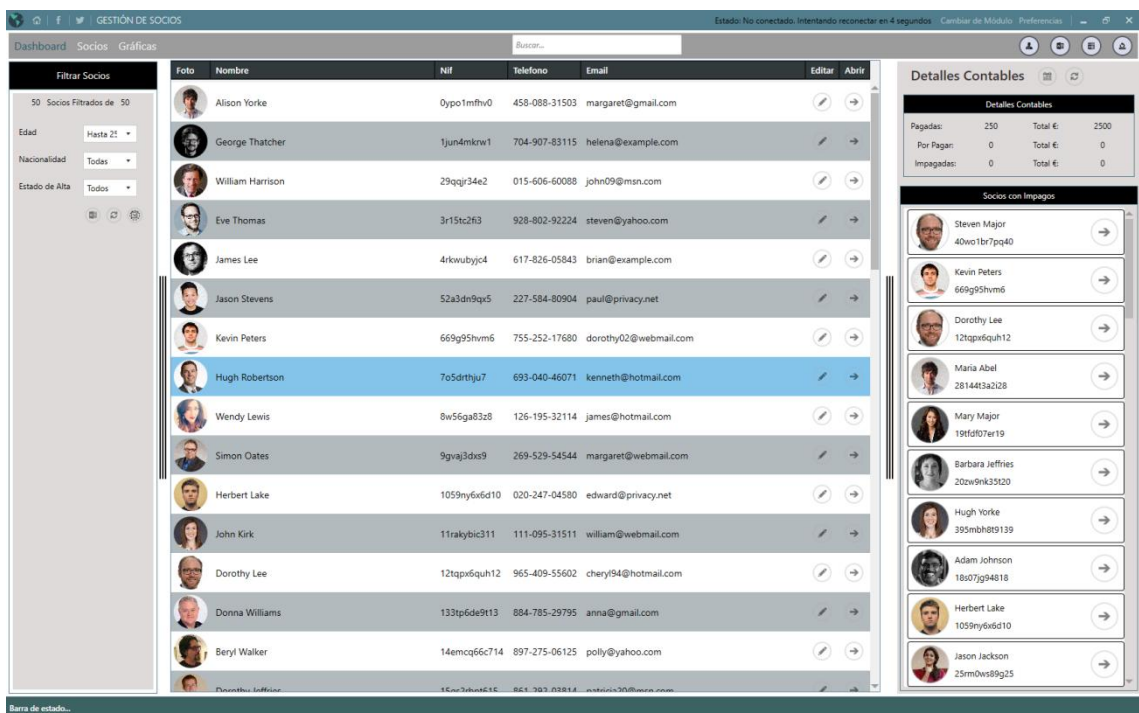


Ilustración 24. Dashboard con listado de impagos y contabilidad general

## 10.5.2. Panel de Socios

Se muestran todos los socios mediante una lista paginada, permitiendo navegar a la información personal de cada socio haciendo doble click en cada elemento de la lista sin tener que volver al Dashboard para navegar al socio escogido.

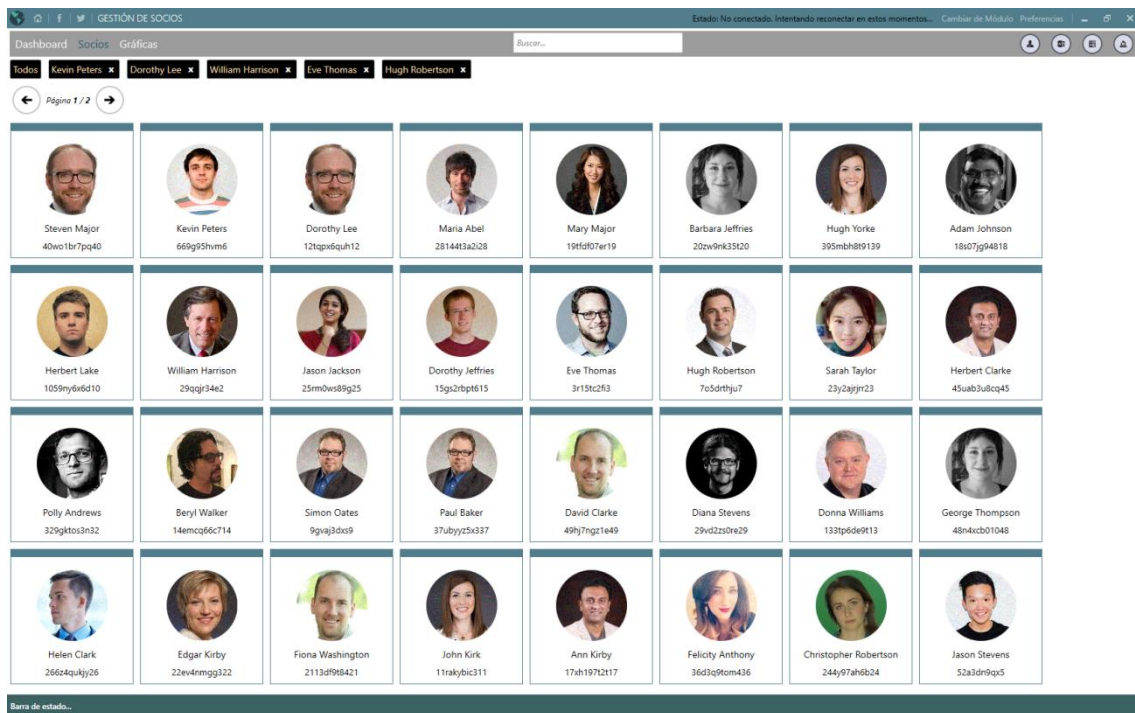


Ilustración 25. Socios con listado completo paginado.

Una vez tenemos un socio seleccionado nos presentamos en la vista con los datos del mismo. Aquí podremos también modificarlos datos personales además de crear los periodos de alta en la organización del socio y las cuotas que conlleva dicho periodo. A medida que se vayan abonando en el control de periodos de alta podemos modificar el estado de las cuotas correspondientes a los meses de pago.

Dashboard Socio Gráficas

Estado: No conectado. Intentando reconectar en 2 segundos. Cambiar de Módulo. Preferencias

Todos Kevin Peters x Dorothy Lee x William Harrison x Eve Thomas x Hugh Robertson x

Nombre: William Harrison  
DNI/NIF: 29qgj34e2  
Fecha de Nacimiento: 12/07/1992  
Email: johr09@msn.com  
Teléfono: 015-606-60088  
Nacionalidad: Channel Islands  
Dirección: 41 Wilson Hill  
Facebook: gmail.com  
Twitter: hotmail.com

12/12/2016 - 12/07/2017					
PAGADO	NA/NC	FECHA	TOTAL A PAGAR	PAGADO	COMENTARIOS
<input checked="" type="checkbox"/>	<input type="checkbox"/>	DIC 2016	10,00 €	10,00 €	Excepteur sint occaecat cupidatat non proident sunt in culpa qui officia d
<input checked="" type="checkbox"/>	<input type="checkbox"/>	ENE 2017	10,00 €	10,00 €	Ut enim ad minim veniam quis nostrud exercitation ullamco laboris nisi u
<input checked="" type="checkbox"/>	<input type="checkbox"/>	FEB 2017	10,00 €	10,00 €	Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore
<input checked="" type="checkbox"/>	<input type="checkbox"/>	MAR 2017	10,00 €	10,00 €	Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore
<input checked="" type="checkbox"/>	<input type="checkbox"/>	ABR 2017	10,00 €	10,00 €	Excepteur sint occaecat cupidatat non proident sunt in culpa qui officia d
<input checked="" type="checkbox"/>	<input type="checkbox"/>	MAY 2017	10,00 €	10,00 €	Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore
<input type="checkbox"/>	<input type="checkbox"/>	JUN 2017	10,00 €	0,00 €	
<input type="checkbox"/>	<input type="checkbox"/>	JUL 2017	10,00 €	0,00 €	

29/04/2016 - 30/09/2016					
PAGADO	NA/NC	FECHA	TOTAL A PAGAR	PAGADO	COMENTARIOS
<input type="checkbox"/>	<input type="checkbox"/>	ABR 2016	10,00 €	0,00 €	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MAY 2016	10,00 €	0,00 €	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	JUN 2016	10,00 €	0,00 €	
<input type="checkbox"/>	<input type="checkbox"/>	JUL 2016	10,00 €	0,00 €	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	AGO 2016	10,00 €	0,00 €	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SEP 2016	10,00 €	0,00 €	

12/07/2017 -

Barra de estado...

Ilustración 26. Socios con socio seleccionado y sus periodos de alta.

Además en esta vista está disponible información contable relacionada con el socio seleccionado. Se generan listas de cuotas que están por pagar así como listas con las cuotas impagadas si tuviera alguna, indicando la fecha correspondiente a dichas cuotas.

Dashboard Socio Gráficas

Estado: No conectado. Intentando reconectar en estos momentos... Cambiar de Módulo. Preferencias

Todos Alison Yorke x

Nombre: Alison Yorke  
DNI/NIF: 0ypo1mfnv0  
Fecha de Nacimiento: 12/07/1992  
Email: margaret@gmail.com  
Teléfono: 458-088-31503  
Nacionalidad: Costa Rica  
Dirección: 41 Chestnut Hill  
Facebook: webmail.com  
Twitter: hotmail.com

12/12/2016 - 12/07/2017					
PAGADO	NA/NC	FECHA	TOTAL A PAGAR	PAGADO	COMENTARIOS
<input checked="" type="checkbox"/>	<input type="checkbox"/>	DIC 2016	10,00 €	10,00 €	Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore
<input checked="" type="checkbox"/>	<input type="checkbox"/>	ENE 2017	10,00 €	10,00 €	Excepteur sint occaecat cupidatat non proident sunt in culpa qui officia d
<input checked="" type="checkbox"/>	<input type="checkbox"/>	FEB 2017	10,00 €	5,00 €	Excepteur sint occaecat cupidatat non proident sunt in culpa qui officia d
<input checked="" type="checkbox"/>	<input type="checkbox"/>	MAR 2017	10,00 €	3,00 €	Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore
<input type="checkbox"/>	<input type="checkbox"/>	ABR 2017	10,00 €	10,00 €	Ut enim ad minim veniam quis nostrud exercitation ullamco laboris nisi u
<input type="checkbox"/>	<input type="checkbox"/>	MAY 2017	10,00 €	10,00 €	Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore
<input type="checkbox"/>	<input type="checkbox"/>	JUN 2017	10,00 €	0,00 €	
<input type="checkbox"/>	<input type="checkbox"/>	JUL 2017	10,00 €	0,00 €	

Cuotas Por Pagar 4 20 €		
Fecha Pago	Total	Pagado
12/04/2017	10	10
12/05/2017	10	10
01/06/2017	10	0
01/07/2017	10	0

Cuotas Impagadas 0 0		
Fecha Pago	Total	Pagado

Barra de estado...

Ilustración 27. Socios con socio seleccionado y sus periodos de alta.

Para añadir un socio bastará con hacer click en la acción correspondiente en la barra de herramientas.

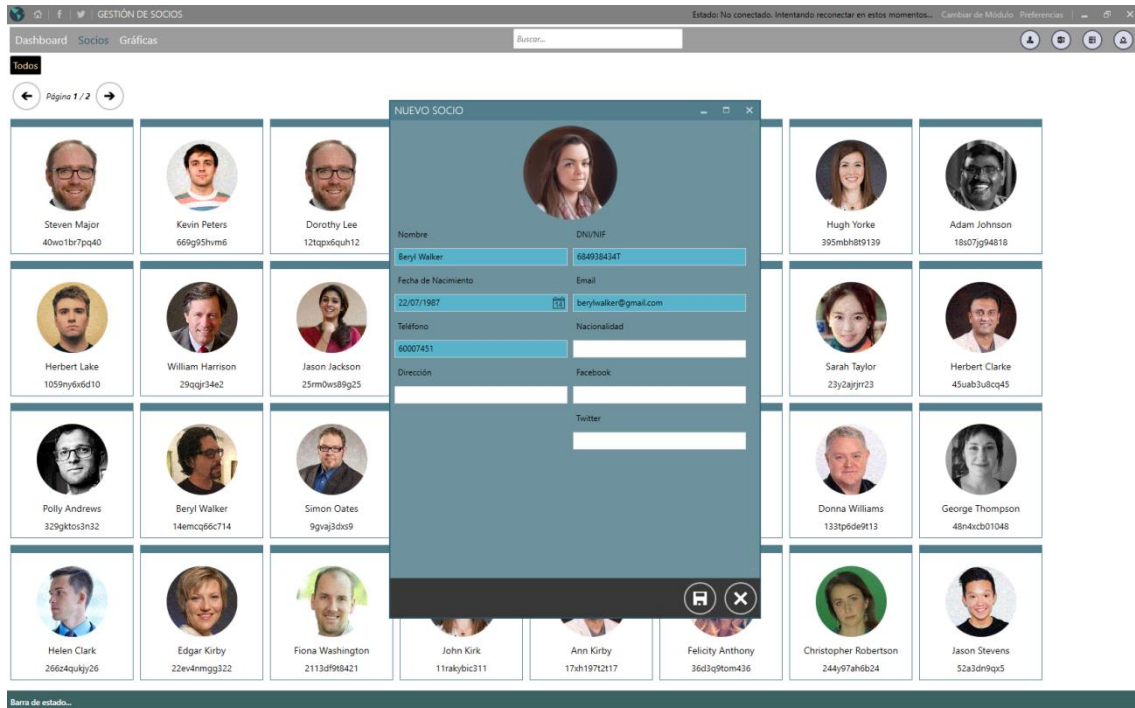


Ilustración 27. Añadir socio.

En las preferencias se configuran algunos aspectos de la interfaz de usuario. También es aquí donde se configuran las opciones de la gestión automática de copias de seguridad.

Dashboard Socios Gráficas

Estado: No conectado Cambiar de Módulo Preferencias

50 Socios Filtrados de 50

Edad: Todas

Nacionalidad: Todas

Estado de Alta: Todos

Foto	Nombre	Nif	Telefono	Email	Editar	Abrir
	Alison Yorke	Oypo1mfhv0	458-088-31503	margaret@gmail.com		
	George Thatcher	1jun4mkrw1	704-907-83115	helena@example.com		
<div style="border: 1px solid #ccc; padding: 5px;"> <p><b>PREFERENCIAS</b></p> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p><b>SOCIOS</b></p> <p>Socios a mostrar en Listado de Socios: 30</p> </div> <div style="width: 45%;"> <p><b>CONTABILIDAD</b></p> <p>Meses impagos para considerar morosidad: 3</p> <p>Cuota mensual predeterminada: 1</p> </div> </div> <hr/> <p><b>COPIAS DE SEGURIDAD</b></p> <p>Hacer copia de seguridad al salir: <input checked="" type="checkbox"/></p> <p>Destino de la copia: C:\Users\Alberto\AppData\Roaming\GamaData</p> <p>Eliminar copias previas a: Seleccione una fecha</p> </div>						
	Simon Oates	9gvaj3dks9	269-529-54544	margaret@webmail.com		
	Herbert Lake	1059my6x6d10	020-247-04580	edward@privacy.net		
	John Kirk	11rakybic311	111-095-31511	william@webmail.com		
	Dorothy Lee	12tqpx6quh12	965-409-55602	chery94@hotmail.com		
	Donna Williams	133tp6de9t13	884-785-29795	anna@gmail.com		
	Beryl Walker	14emcq66c714	897-275-06125	polly@yahoo.com		

**Detalles Contables**

Detalles Contables	
Pagadas: 250	Total €: 2500
Por Pagar: 1	Total €: 10
Impagadas: 6	Total €: 60

**Socios con Impagos**

- Steven Major (40wo1br7pq40)
- Kevin Peters (669g55hvm6)
- Dorothy Lee (12tqpx6quh12)
- Maria Abel (28144t3a2t28)
- Mary Major (19efdt07er19)
- Barbara Jeffries (20zwn9nk35t20)
- Hugh Yorke (395mbh89t39)
- Adam Johnson (18s07g948t8)
- Herbert Lake (1059my6x6d10)
- William Harrison (29qqr34e2)

Barra de estado...

Ilustración 28. Preferencias

## 10.6. DISEÑO DE LA INTERFAZ DEL MÓDULO DE COOPERACIÓN

VISTA	REQUISITOS QUE CUBRE
Borde superior de la ventana	<p>Acciones para acceder a los sitios Web de Gamá y para volver al selector de módulo.</p> <p>Ver estado de conexión con el servidor de sincronización.</p> <p>Cambiar de módulo.</p> <p>Mostrar el cuadro de preferencias.</p>
Panel de navegación	Navegar a los distintos paneles.
Cuadro de búsqueda	<p>Buscar Actividad.</p> <p>Navegar a la actividad seleccionada</p>
Barra de herramientas	<p>Añadir Actividad.</p> <p>Añadir cooperante</p> <p>Exportar lista de actividades o una actividad seleccionada.</p> <p>Exportar la lista de cooperantes o un cooperante seleccionado.</p> <p>Hacer copia de seguridad.</p> <p>Restaurar copia de seguridad</p>
Barra de estado	Notificar acciones completadas exitosamente
Dashboard	<p>Visualizar lista de Actividades y editar la información de cada actividad.</p> <p>Navegar a cada actividad.</p> <p>Eliminar cada actividad.</p> <p>Filtrar listados de actividades.</p> <p>Mostrar la lista de eventos de las actividades.</p> <p>Navegar a la actividad desde el evento acaecido.</p> <p>Filtrar la lista de eventos por fechas.</p>
Actividades	
Listado actividades	Visualizar lista completa de actividades y navegar a una actividad seleccionado
Actividad seleccionado	<p>Mostar lista tareas que componen la actividad.</p> <p>Crear y borrar tareas.</p> <p>Borrar tareas</p> <p>Mostrar el historial de desarrollo de la tarea.</p>

Cooperantes	<p>Mostrar incidencia y complicaciones en el desarrollo de la tarea.</p> <p>Crear y borrar foros de discusión</p> <p>Mostrar lista de foros de discusión sobre la tarea o la actividad.</p> <p>Mostrar los mensajes enviados a los foros de discusión.</p> <p>Mostrar lista de eventos acaecidos en la actividad seleccionada.</p> <p>Filtrar la lista de eventos por fechas.</p>
	<p>Mostrar información del cooperante seleccionado.</p> <p>Mostrar lista de actividades en las que trabaja.</p> <p>Modificar datos del cooperante seleccionado.</p> <p>Exportar los datos del cooperante seleccionado.</p> <p>Mostrar lista completa de cooperantes.</p> <p>Seleccionar cooperantes que mostrar de la lista completa de cooperantes</p>
Calendario	<p>Mostrar las actividades en formato de calendario según la fecha de entrega.</p> <p>Crear actividades desde la fecha del calendario.</p>
Gráficas	<p>Visualizar gráficas.</p>
Desarrollo de la tarea	<p>Ver lista de trabajos realizados.</p> <p>Añadir nuevos trabajos realizados en la tarea.</p>
Incidencias	<p>Ver listado de complicaciones surgidas en el desarrollo de la tarea.</p> <p>Añadir nuevas incidencias.</p>
Foro seleccionado	<p>Ver todos los mensajes enviados al foro de discusión.</p> <p>Añadir nuevos mensajes.</p>

### 10.6.1. Dashboard.

El Dashboard sirve como vista global donde poder acceder de forma rápida a cualquier actividad creada en el módulo así como de todos los eventos ocurridos en todas las actividades. Desde la lista de actividades se podrá modificar la información de cada una, así como las acciones de borrado y navegación al detalle de cada actividad. Desde los eventos también se puede navegar al detalle de la actividad donde se originó dicho evento. En esta lista de eventos se indica la fecha y hora en que se generó el evento, que evento se generó y la actividad donde se originó.

The dashboard displays a table of activities and a sidebar of events. The activities table has columns for ID, Título, Estado, Fecha Final, Coordinador, Editar, Eliminar, and Abrir. The events sidebar shows a list of events with dates and titles, each with a right-pointing arrow icon.

ID	Título	Estado	Fecha Final	Coordinador	Editar	Eliminar	Abrir
90	Duis aute irure dolor in reprehenderit in voluptate velit esse cillum	Comenzado	26/08/2017	Kevin	✍	🗑	➔
89	Ut enim ad minim veniam quis nostrud exercitation ullamco labc	Comenzado	25/08/2017	Linda	✍	🗑	➔
88	Excepteur sint occaecat cupidatat non proident sunt in culpa qui	Comenzado	25/08/2017	Sharon	✍	🗑	➔
87	Duis aute irure dolor in reprehenderit in voluptate velit esse cillum	Comenzado	24/08/2017	Donald	✍	🗑	➔
86	Excepteur sint occaecat cupidatat non proident sunt in culpa qui	Comenzado	23/08/2017	Laura	✍	🗑	➔
85	Duis aute irure dolor in reprehenderit in voluptate velit esse cillum	Comenzado	22/08/2017	Marcus	✍	🗑	➔
84	Lorem ipsum dolor sit amet consectetur adipiscing elit sed do €	Comenzado	22/08/2017	Diana	✍	🗑	➔
83	Excepteur sint occaecat cupidatat non proident sunt in culpa qui	Comenzado	21/08/2017	Joseph	✍	🗑	➔
82	Lorem ipsum dolor sit amet consectetur adipiscing elit sed do €	Comenzado	20/08/2017	Edgar	✍	🗑	➔
81	Ut enim ad minim veniam quis nostrud exercitation ullamco labc	Comenzado	19/08/2017	Sharon	✍	🗑	➔
80	Duis aute irure dolor in reprehenderit in voluptate velit esse cillum	Comenzado	19/08/2017	Dorothy	✍	🗑	➔
79	Ut enim ad minim veniam quis nostrud exercitation ullamco labc	Comenzado	18/08/2017	Christopher	✍	🗑	➔
78	Duis aute irure dolor in reprehenderit in voluptate velit esse cillum	Comenzado	17/08/2017	Sandra	✍	🗑	➔
77	Duis aute irure dolor in reprehenderit in voluptate velit esse cillum	Comenzado	16/08/2017	Anne	✍	🗑	➔
76	Ut enim ad minim veniam quis nostrud exercitation ullamco labc	Comenzado	16/08/2017	Jessica	✍	🗑	➔
75	Lorem ipsum dolor sit amet consectetur adipiscing elit sed do €	Comenzado	15/08/2017	Robert	✍	🗑	➔
74	Lorem ipsum dolor sit amet consectetur adipiscing elit sed do €	Comenzado	14/08/2017	Donna	✍	🗑	➔
73	Excepteur sint occaecat cupidatat non proident sunt in culpa qui	Comenzado	13/08/2017	George	✍	🗑	➔
72	Duis aute irure dolor in reprehenderit in voluptate velit esse cillum	Comenzado	13/08/2017	Brian	✍	🗑	➔
71	Ut enim ad minim veniam quis nostrud exercitation ullamco labc	Comenzado	12/08/2017	Michael	✍	🗑	➔

Ilustración 29. Dashboard general.

En esta vista también se nos permite poder filtrar la lista de actividades y la lista de eventos. Para la lista de actividades se cuenta de filtrado por el estado de la actividad y filtrado por meses, mostrando las actividades cuyas fechas de finalización se sitúan en los meses y año indicados. La lista de eventos se puede filtrar por fechas, todos los generados desde una fecha concreta o todos los generados en un rango de fechas concreto.



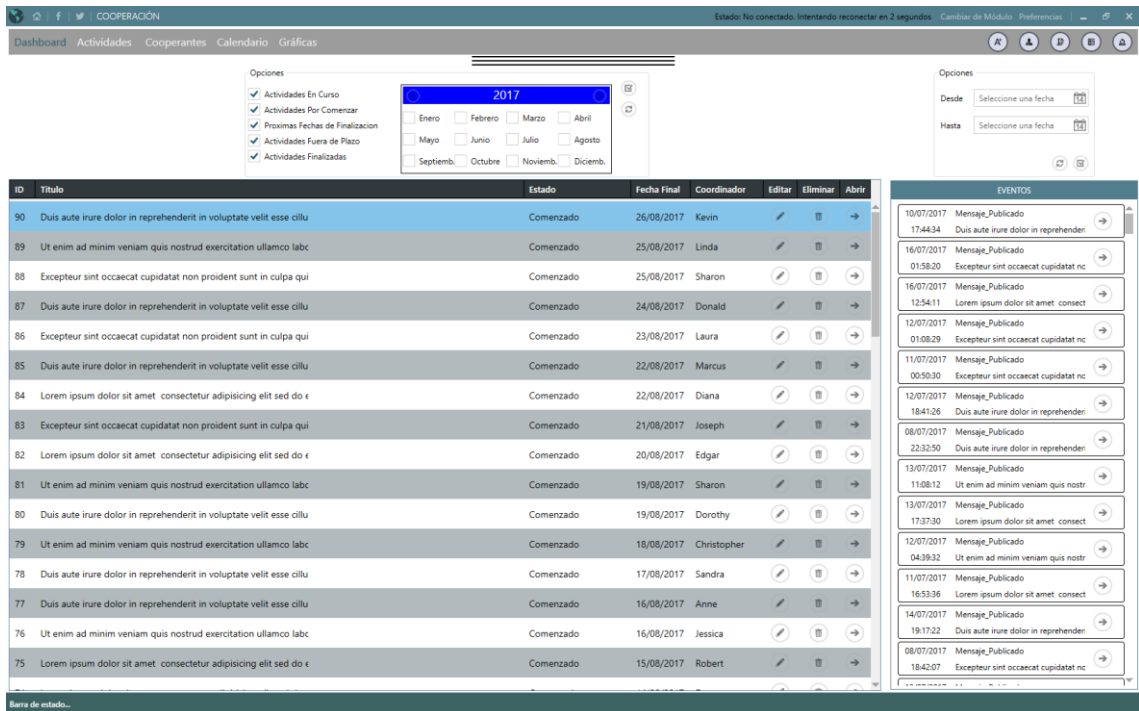


Ilustración 30. Dashboard con filtrado de actividades y eventos.

### 10.6.2. Actividades.

Se muestran todas las actividades mediante una lista paginada, permitiendo navegar a la información de cada actividad haciendo doble click en cada elemento de la lista sin tener que volver al Dashboard para navegar a la actividad escogida.

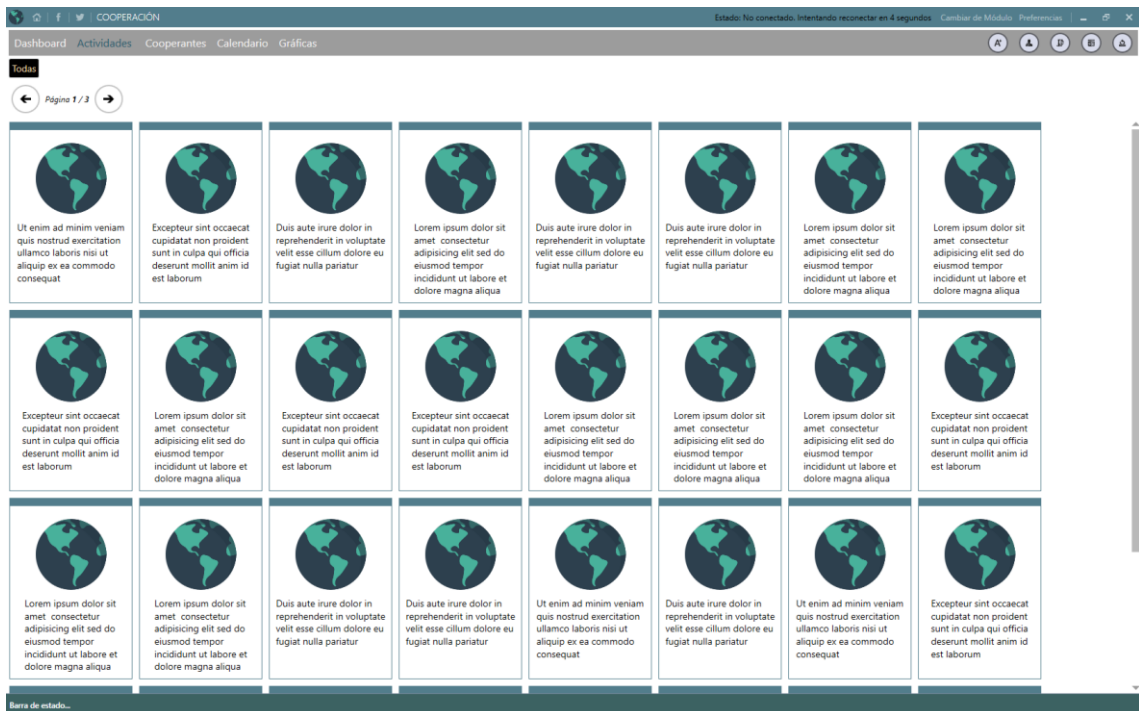


Ilustración 31. Actividades con listado completo paginado

Al navegar a una actividad concreta, en la vista de Actividades se muestra una lista con las tareas en las que se subdivide la actividad y una lista con los foros de discusión sobre las tareas o la actividad seleccionada. En esta vista se pueden crear, borrar, editar, añadir incidencias y añadir trabajos desarrollados sobre las tareas y en lo concerniente a los foros de discusión se pueden crear nuevos foros y acceder al historial de mensajes.

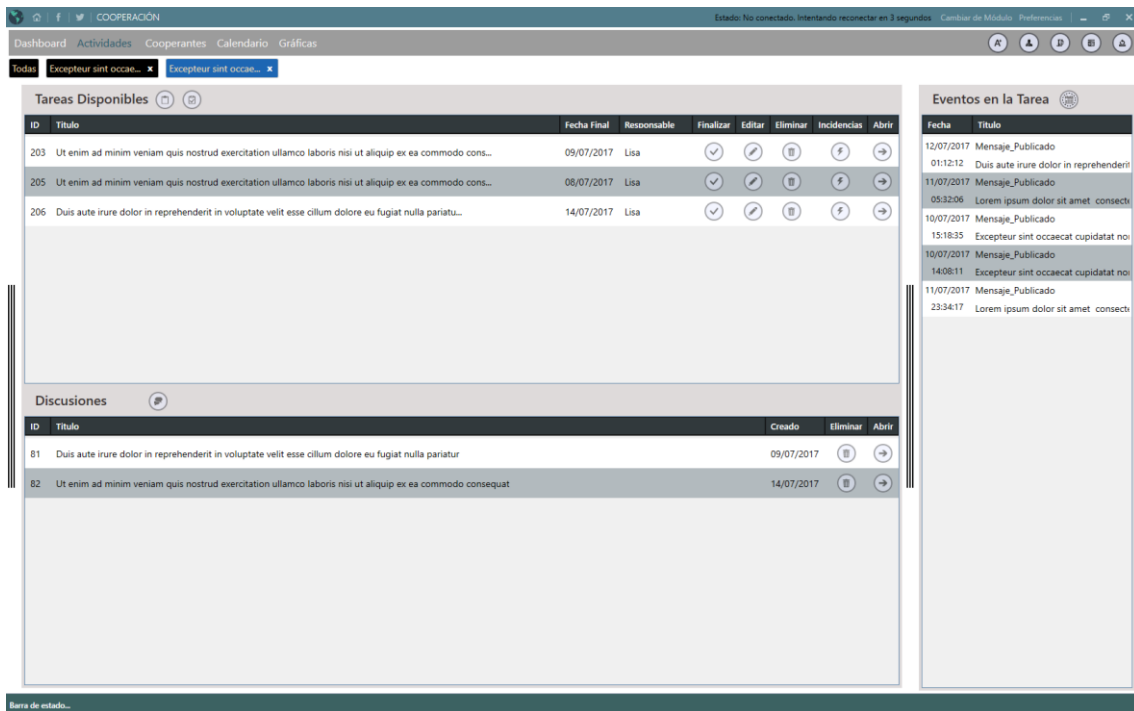


Ilustración 32. Actividades con la actividad mostrando tareas y foros de discusión

Acompañando a soporte para tareas y foros de discusión tenemos la lista de eventos que se han originado en la actividad seleccionada.

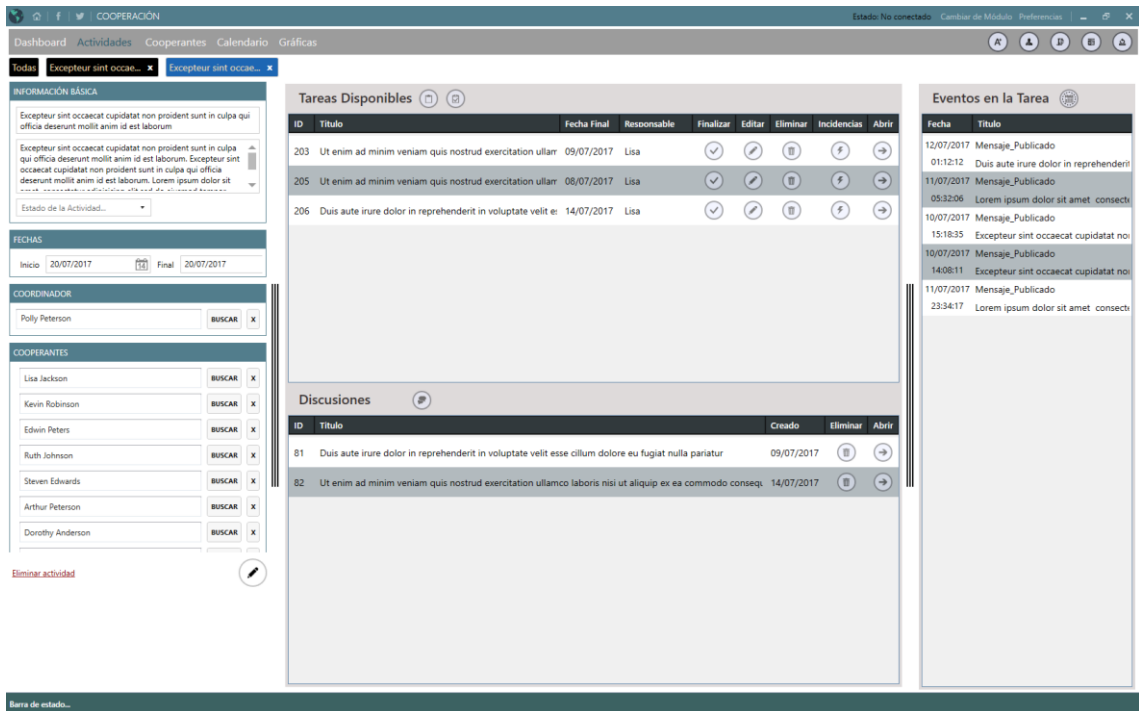


Ilustración 33. Actividades con información de la actividad seleccionada, tareas y foros.

Tenemos la posibilidad de filtrar los eventos originados en la actividad con un filtro por fechas. Muestra todos los eventos originados desde una fecha o muestra todos los eventos originados en un rango de fechas.

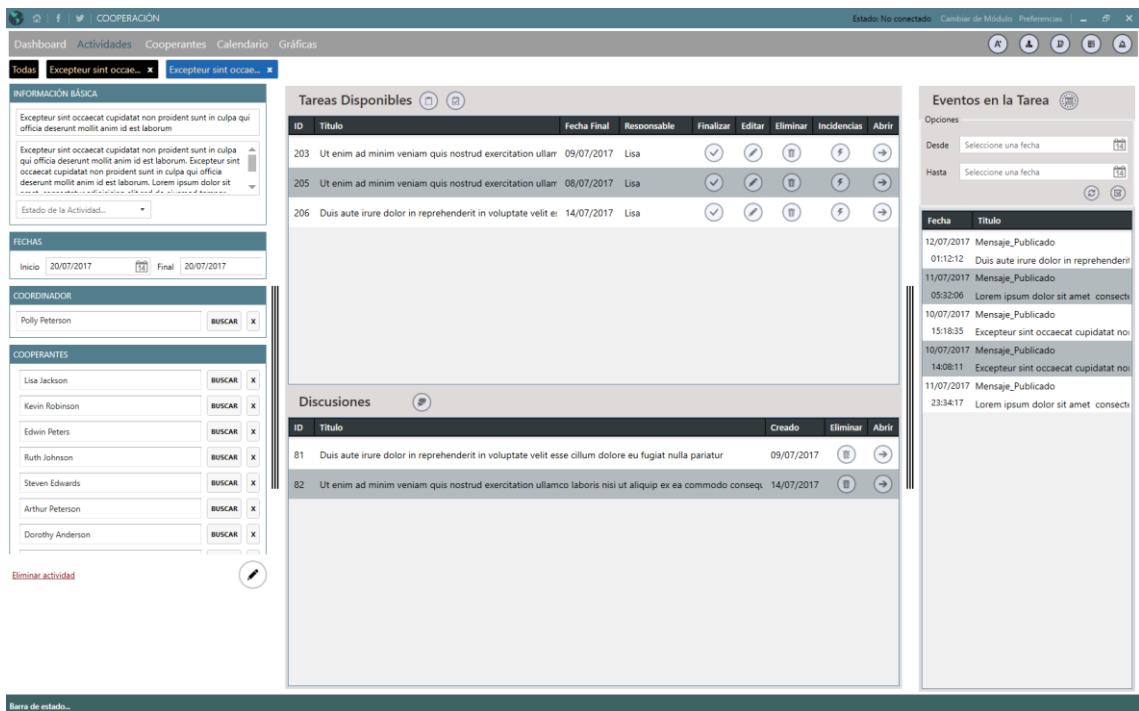


Ilustración 34. Actividades con el filtro de eventos de la actividad.

Tanto al añadir una actividad como cuando se edita, se pueden modificar sus cooperantes y el coordinador asignado con un pequeño cuadro auxiliar para seleccionar cooperantes.

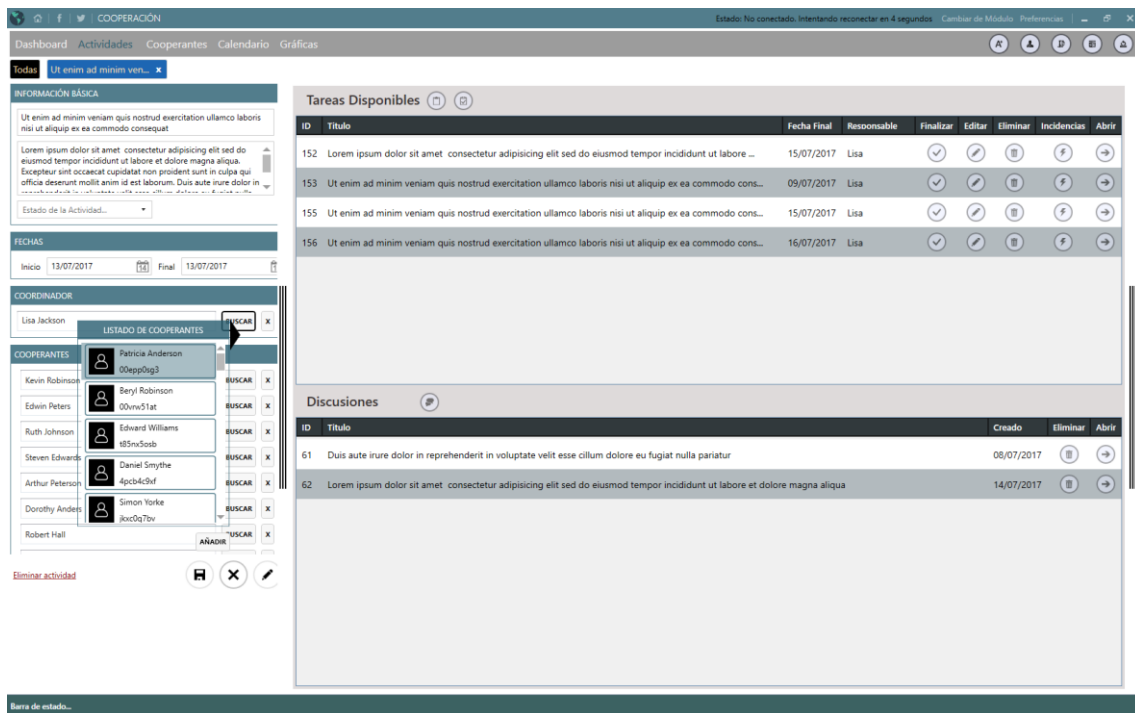


Ilustración 57. Editando el coordinador o cooperantes de una actividad

### 10.6.3. Cooperantes.

En el panel de cooperantes es donde accedemos a todos los datos de los cooperantes con los que cuenta la organización. Aquí en un primer vistazo tenemos la información personal de los cooperantes y la posibilidad de modificarlos y las lista de las actividades en las que participa como coordinador y la lista de las actividades en las que participa como cooperante. Como se muestran lista de actividades se brinda la posibilidad de poder editar, borrar o navegar a la actividad en la que trabaja el cooperante sin tener que volver a alguna de las vistas anteriores donde también se permiten estas acciones.

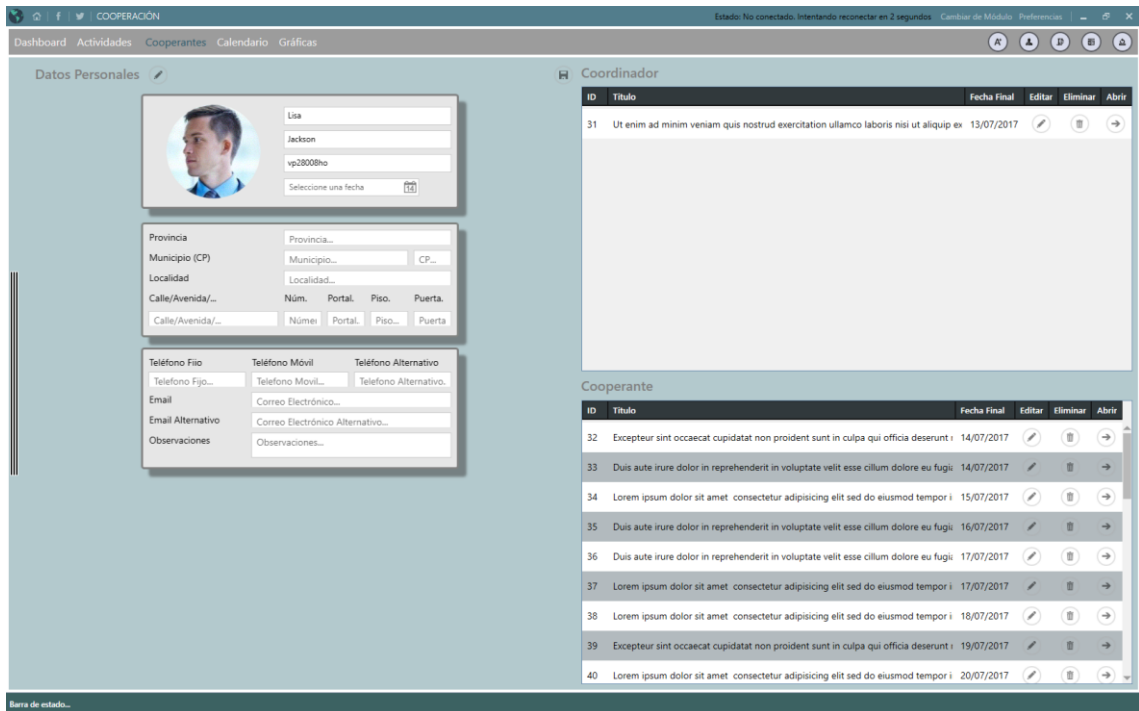


Ilustración 35. Cooperantes con información del cooperante seleccionado, tareas y foros.

Contamos con una lista completa de los cooperantes para tener acceso a la información de cada uno con un solo click de ratón.

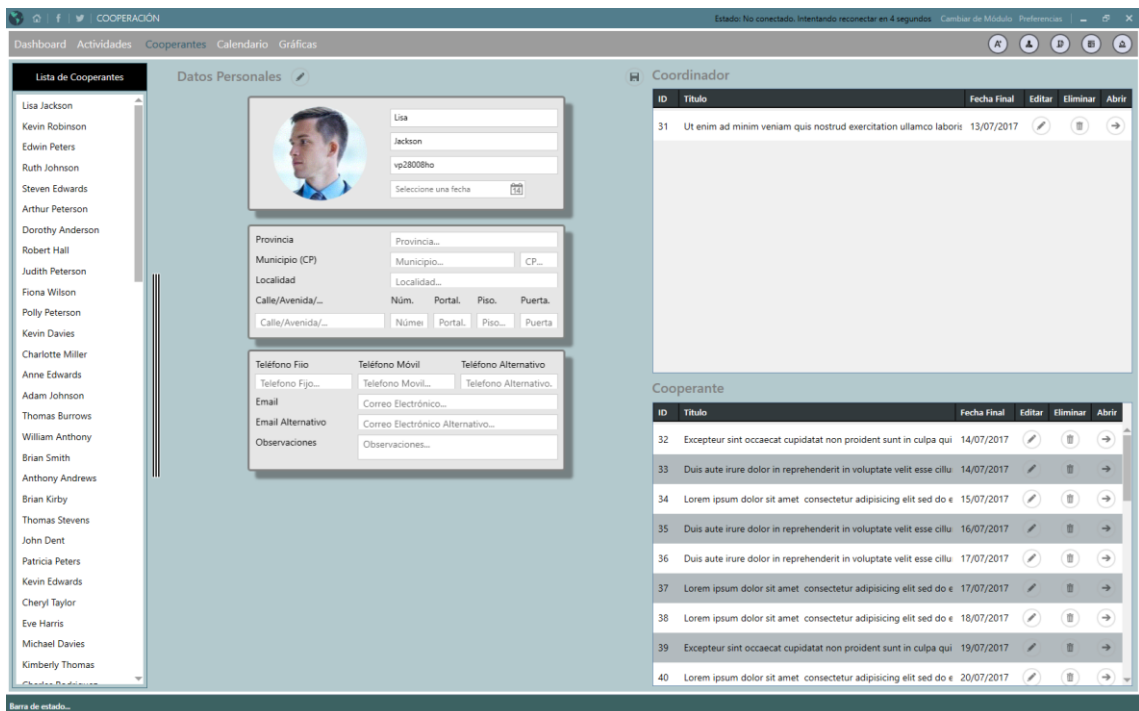


Ilustración 36. Actividades Cooperantes con información del cooperante seleccionado y lista completa de cooperantes.

### 10.6.4. Nueva actividad.

Esta es la vista para añadir una nueva actividad. Se ingresan todos los datos necesarios. Se indican cuales son los mínimos necesarios para poder crear la actividad. Si estos campos no están completados no se habilitará el botón para poder crear la actividad.

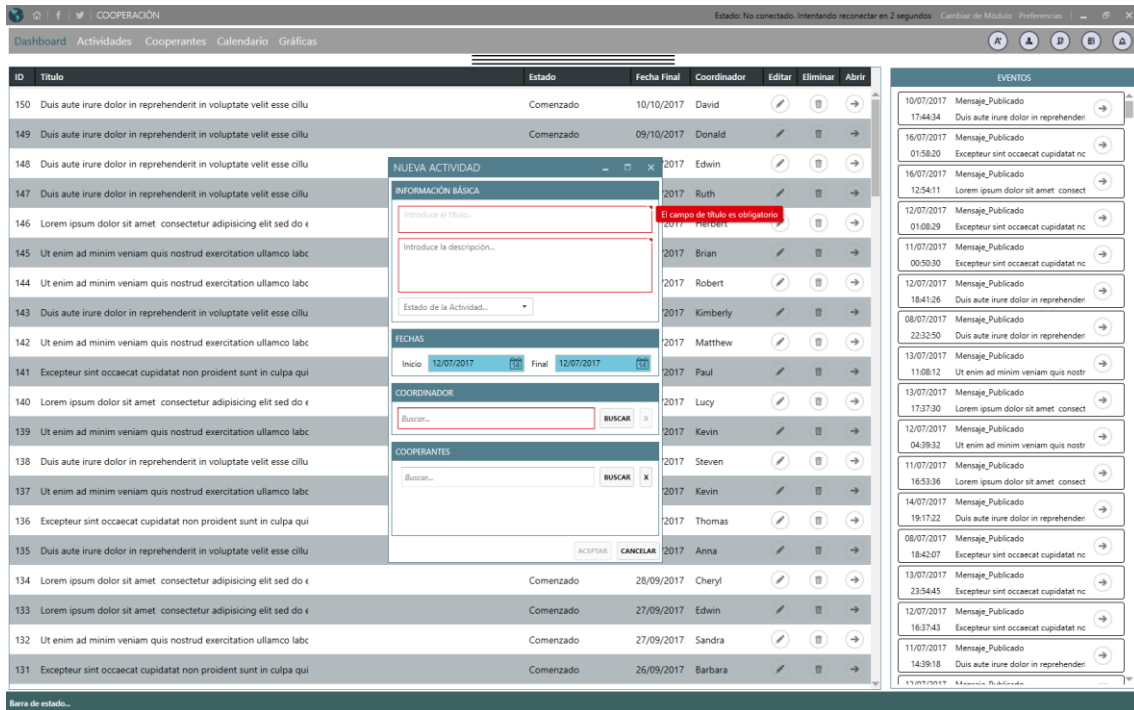


Ilustración 37. Nueva actividad.

### 10.6.5. Nuevo Cooperante.

Esta vista nos permite añadir nuevos cooperantes. Siguiendo con la misma filosofía usada para crear nuevas actividades, se indican los campos imprescindibles para habilitar el botón y añadir el nuevo cooperante.

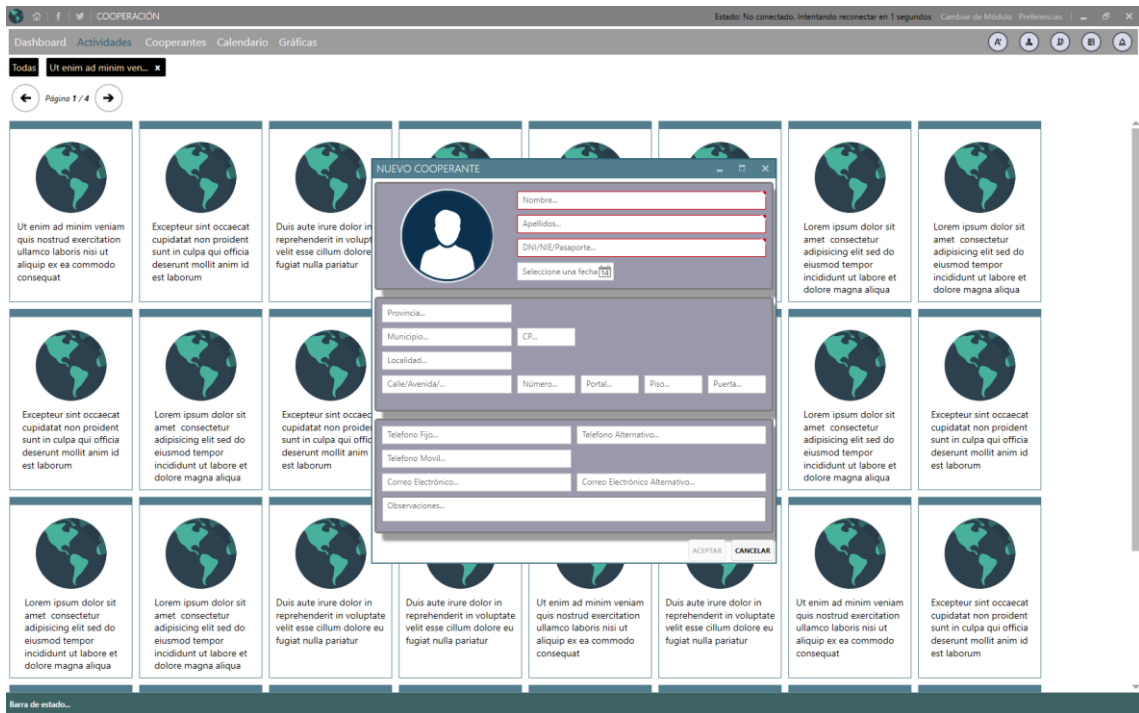


Ilustración 38. Nuevo cooperante

### 10.6.6. Nueva Tareas.

Es la vista que nos permite añadir tareas a una actividad. Una vez creada estarán disponibles las acciones que se pueden ejecutar sobre la actividad.

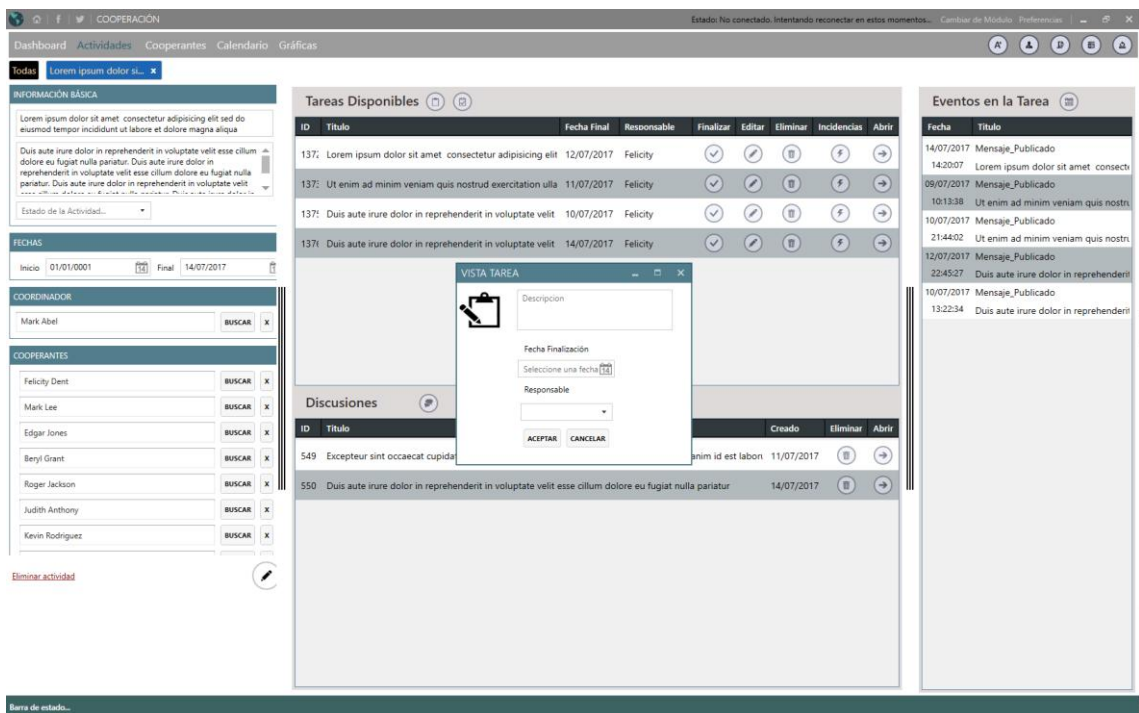


Ilustración 39. Nueva tarea.

### 10.6.7. Foros de discusión

Para tener un entorno donde poder intercambiar ideas u opiniones sobre las tareas a realizar o sobre la propia actividad se crean los foros de discusión. Con esta vista podemos añadir todos los foros necesarios.

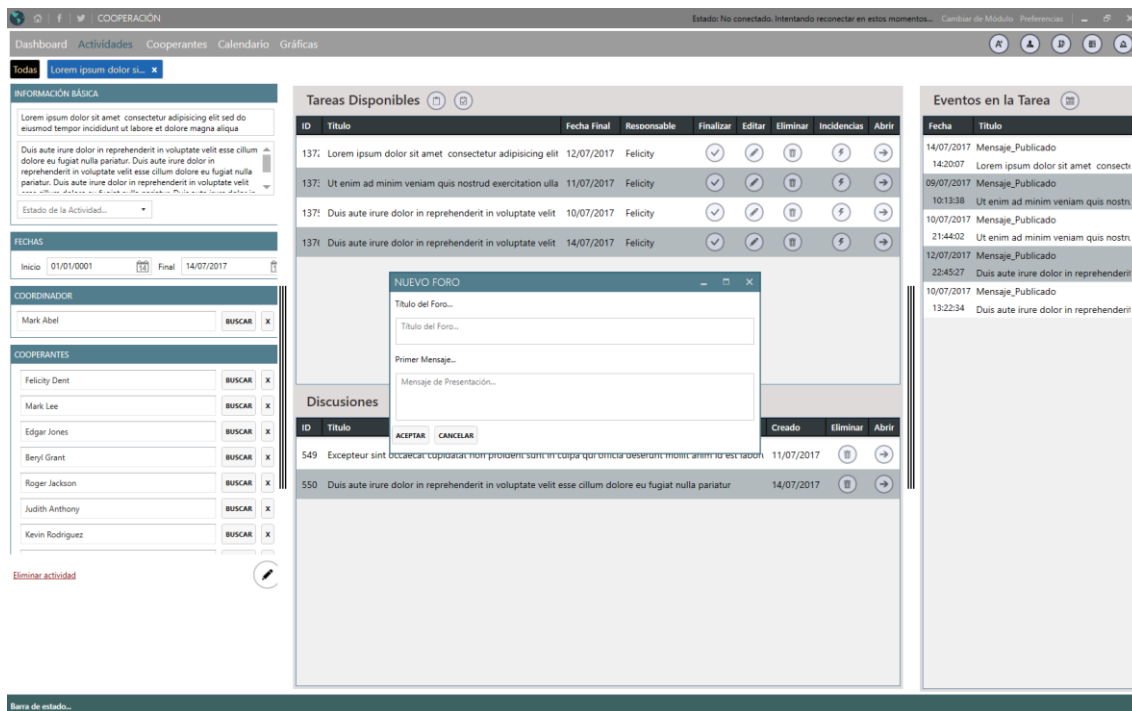


Ilustración 40. Nuevo foro de discusión

### 10.6.8. Seguimiento.

Para llevar a cabo una tarea concreta se puede necesitar llevar a cabo diferentes actuaciones o trabajos. Se dispone un historial de trabajos realizados en la tarea para tener una visión completa de lo que se ha realizado. Se podrán añadir a este seguimiento cualquier trabajo realizado.



The screenshot displays a web application interface for task management. The top navigation bar includes 'Dashboard', 'Actividades', 'Cooperantes', 'Calendario', and 'Gráficas'. The main content area is divided into several sections:

- INFORMACIÓN BÁSICA:** Contains a text area with placeholder text and a dropdown menu for 'Estado de la Actividad...'.
- FECHAS:** Shows 'Inicio' as 14/07/2017 and 'Final' as 14/07/2017.
- COORDINADOR:** Lists 'Mark Lee' with a search button.
- COOPERANTES:** Lists several names (Felicity Dent, Mark Abel, Edgar Jones, Beryl Grant, Roger Jackson, Judith Anthony, Kevin Rodriguez) with search buttons.
- Tareas Disponibles:** A table with columns 'ID', 'Titulo', 'Fecha Final', 'Responsable', 'Finalizar', 'Editar', 'Eliminar', 'Incidencias', and 'Abrir'. It shows a task with ID 138, title 'Ut enim ad minim veniam...', and responsible person 'Felicity'.
- DESARROLLO DE LA TAREA:** A detailed view of the task showing a timeline of activities with dates and times (e.g., 11/07/2017 14:10:36, 15/07/2017 14:15:48, 13/07/2017 13:38:01).
- Discusiones:** A section for discussions with a table with columns 'ID', 'Titulo', 'Estado', 'Eliminar', and 'Abrir'. It shows two entries with IDs 551 and 552, both titled 'Excepteur'.
- Eventos en la Tarea:** A list of events with columns 'Fecha' and 'Titulo', showing various timestamps and titles like 'Mensaje\_Publicado'.

Ilustración 41. Historial de trabajos realizados en una tarea.

### 10.6.9. Incidencias.

Cuando se lleva a cabo una tarea pueden surgir contratiempos en la ejecución de los trabajos para completarla. En incidencias se cree un entorno donde poder llevar un control de los problemas surgidos y que los demás cooperantes de la actividad y su coordinar puedan conocer los detalles de las complicaciones surgidas.

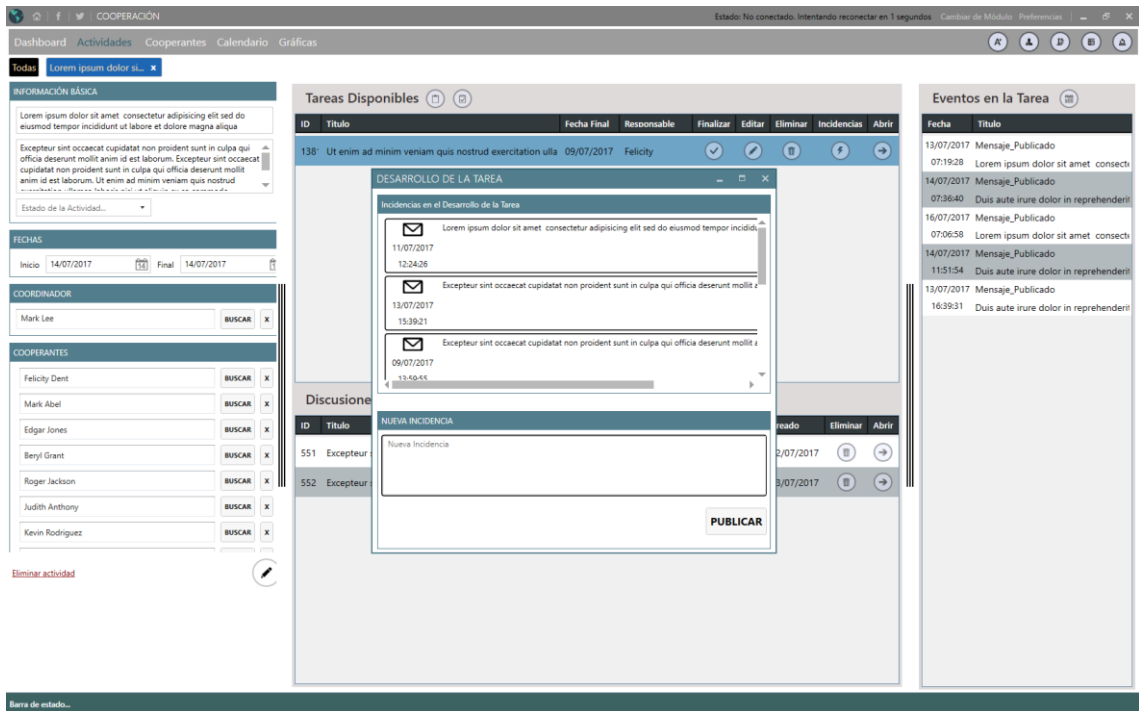


Ilustración 42. Incidencias en el desarrollo de una tarea.

### 10.6.10. Calendario.

Para tener una visión global de las actividades según su fecha de finalización se genera este calendario.

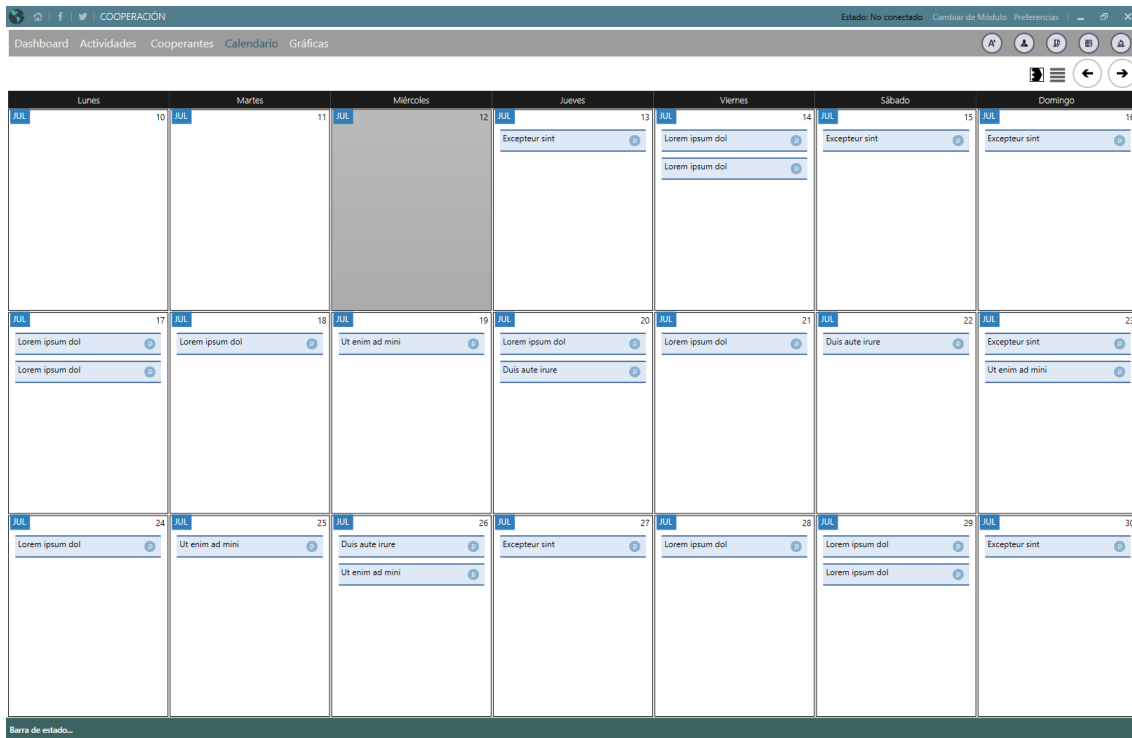


Ilustración 58. Calendario con las fechas de fin de las actividades

## 10.7. DISEÑO DE LA INTERFAZ DE LA EXTRANET

La extranet cuenta con un diseño sencillo y se limita a mostrados listados. Hay una sección por cada listado que se muestra. Las secciones son: personas, citas, asistentes, socios, cooperantes. A continuación de muestra una de estas listas, a modo representativo del resto.









Index - My ASP.NET App X

localhost:50076

Application name Home About Personas Citas Socios Asistentes Contact Register Log in

## Index

[Create New](#)

Id	Imagen	Nombre	Nif	Email	Nacionalidad	DireccionPostal	EstadoCivil	FechaDeNacimiento	Id
24		Alison Moore	3hatj60s93	linda@webmail.com	Dominica	30 Cedar Hill	Soltera	12/07/1997 4:45:38	Hc
39		Anna Thomas	187d45k69418	edgar@webmail.com	Guinea-Bissau	14 Larch Park	Soltera	12/07/1997 4:45:38	Mi
14		Beryl Edwards	137gruecmj13	arthur@yahoo.com	Uruguay	49 Jackson Park	Soltera	12/07/1997 4:43:06	Hc
10		Charlotte Garcia	9atbrkxh89	victoria@webmail.com	Colombia	27 Franklin Highland	Soltera	12/07/1997 4:43:06	Mi
3		Cheryl Smythe	2orgurxn12	ursula47@example.com	Czech Republic	13 Rose Hill	Soltera	12/07/1997 4:43:06	Hc
46		Daniel Garcia	5nyzq8zdg5	adam71@msn.com	Maldives	49 Maple Virginia	Soltera	12/07/1997 4:46:29	Nc
7		David Burrows	6ti2gnm1h6	betty87@example.com	Bangladesh	44 Madison Virginia	Soltera	12/07/1997 4:43:06	Hc
29		David Burrows	8bhk4z9128	adam@example.com	Austria	3 Madison Sunset	Soltera	12/07/1997 4:45:38	Hc

59. Uno de los listados mostrados en la Extranet. En este caso, el listado de personas.

---

# 11. TRABAJO FUTURO

---

Aunque se han cumplido los objetivos satisfactoriamente, la aplicación, concebida en sus inicios como base tecnológica a escalar en el futuro, cuenta a día de hoy con expectativas prometedoras. Algunas de las ideas que se barajan se listan a continuación.

## 11.1. TRABAJO FUTURO GENERAL

En relación a cuestiones generales, algunas líneas de trabajo futuro consideradas han sido las siguientes:

- Ofrecer más opciones de configuración de usuario
- Interacción e integración entre módulos (un socio podría ser un cooperante, o un atendido, y diferentes combinaciones)
- Enriquecer la interfaz con más *feedback* visual. Hay bastantes situaciones donde, por ejemplo, una iluminación transitoria ayudaría a llevar la vista del usuario a donde corresponde, mejorando la eficiencia así como la experiencia de uso. Poder trabajar offline y luego sincronizar. Ahora mismo se requiere conexión a Internet para poder acceder a cualquier de los módulos.
- Replantear la arquitectura del ensamblado *Gama.\*.Wpf* para tratar de organizarlo mejor en distintos ensamblados, de forma que, por ejemplo, el ensamblado *Gama.Extranet* no dependa de dicho ensamblado completamente sino de partes diferenciadas que hayan sido separadas.
- Abstraer el sistema personalizado de navegación que se implementó como alternativa a la navegación con Prism.
- Hacer más sofisticado (aumentando las prestaciones) el sistema de comunicación entre los clientes y el servidor de sincronización.
- Plantear el uso de librerías libres de inteligencia artificial (*machine learning* probablemente) al modo que se utilizan en centros médicos (al menos en EEUU) para predecir distintas necesidades asistenciales de las personas que pasan por Gamá.

## 11.2. TRABAJO FUTURO PARA EL MÓDULO DE ATENCIONES

En relación al servicio de atenciones, algunas líneas de trabajo futuro consideradas han sido las siguientes:

- Añadir vista de calendario en formato de planificador semanal (*scheduler*).
- Generar gráficas sobre más conjuntos de datos
- Exportar gráficas a formato CSV o XLSX.
- Establecer, en la barra superior de acciones, notificaciones al arrancar el programa de las citas programadas para hoy.
- Permitir la programación de avisos al email o teléfono de un asistente que tenga una cita próxima en el tiempo.

## 11.3. TRABAJO FUTURO PARA EL MÓDULO DE SOCIOS

En relación a la gestión de socios, algunas líneas de trabajo futuro consideradas han sido las siguientes:

- Sistema de aviso de socios próximos a cumplir años o que cumplen el día presente.
- Permitir el envío de correos electrónicos y mensajes de teléfono desde la aplicación para notificar de cuestiones como el impago de cuotas.

## 11.4. TRABAJO FUTURO PARA EL MÓDULO DE COOPERACIÓN

En relación al módulo de cooperación, algunas líneas de trabajo futuro consideradas han sido las siguientes:

- Dividir cooperantes en distintas categorías, como voluntarios, colaboradores parciales, entidades específicas, etc., de forma que se facilite la gestión de cooperación en relación a las distintas actividades. Estas nuevas categorías contarían con nueva información asociada, como preferencias de actuación, disponibilidad, seguimiento de relación con ellos, etc.

- Plantear un sistema de mensajería más sofisticado e integrado.

## 11.5. TRABAJO FUTURO PARA LA EXTRANET

Finalmente, en relación a la extranet, algunas líneas de trabajo futuro consideradas han sido las siguientes:

- Aumentar la cantidad de información accesible
- Diferenciar entre distintos usuarios y permisos de acceso a la información

---

## 12. CONCLUSIÓN

---

Si se hubieran de **resumir y abstraer los objetivos** de este proyecto, tendríamos, por un lado, ofrecer a Gamá una **aplicación moderna** (con una experiencia de usuario rica) **y eficiente** y, por otro lado, que ha tratado sobre el **desarrollo de aplicaciones altamente estructuradas y de bajo acoplamiento**, donde se enfatiza el diseño de una arquitectura que facilite la creación de pruebas y el mantenimiento, así como dar soporte a los desarrolladores para trabajar con mayor independencia y productividad. Todo esto supone retos técnicos y organizativos que no deben ser infravalorados. Los costes de tener un código acoplado se conocen, de ahí que surgieran y sigan creándose alternativas de diseño como las adoptadas en este proyecto. No obstante, hay que **ser conscientes de los costes de hacer un diseño desacoplado** [véase el artículo 1 de la *bibliografía*], para **analizar si se ajusta** a la idiosincrasia de un desarrollo determinado y si es por tanto apropiado adoptar tal diseño.

Dadas las características de este proyecto, en el que **los requisitos han sido cambiantes** tanto en detalle como en número (aparecieron nuevos requisitos durante del desarrollo), y en el que el diseño ha tenido por tanto que ir evolucionando, **el impacto de un cambio** (número y *peso* de los cambios que a su vez provoca) **se vuelve un criterio prioritario** a la hora de elegir la metodología de desarrollo y el enfoque arquitectónica.

**A la luz de los resultados**, que se muestran en esta memoria en forma de requisitos implementados –funcionales y de calidad–, así como de argumentaciones varias y múltiples esquemas y diagramas, **se concluye lo siguiente**:

**En primer lugar**, que los objetivos planteados para Gamá han sido alcanzados, ya que **los objetivos mínimos de funcionalidad y calidad se han alcanzado y superado ampliamente**, se ha implementado un **diseño visual moderno** y una **interacción user-friendly**, y se han mantenido **buenos niveles de rendimiento**.

**En segundo lugar**, que el desarrollo ha sido en efecto satisfactorio:

1. En **términos de impacto de los cambios** propios a la constante evolución del sistema, ya que en general se pudieron abordar de forma ordenada, además de



suponer un ejercicio periódico de re-entendimiento, de revisión informal de código, de la arquitectura tanto a nivel global como a niveles más locales.

2. En **términos organizativos**, ya que **se permitió el desarrollo independiente** no sólo de vistas y resto de componentes en general para un mismo desarrollador, sino entre los desarrolladores y los ensamblados, pudiendo, como se ha mostrado, trabajar en un módulo aun cuando los otros ni siquiera compilen, lo cual contribuye al **aumento de productividad**.
3. En **términos académico y profesionales**, se ha podido en efecto **poner en práctica y aprender ampliamente** sobre las tecnologías, técnicas, método organizativo y flujo de trabajo.

---

## 13. BIBLIOGRAFÍA

---

Aunque se han consultado más fuentes, las principales se listan a continuación. Se ha dividido la bibliografía entre: a) Libros y documentos; b) Sitios web con contenido textual y/o audiovisual; c) Artículos.

### 13.1. LIBROS Y DOCUMENTOS

Aunque se han consultado más libros y documentos, los principales son estos tres:

- [WPF Unleashed](#)
- [Prism 5 for WPF](#)
- [Design Patterns: Elements of Reusable Object-Oriented Software](#)

### 13.2. SITIOS WEB CON RECURSOS TEXTUALES Y AUDIOVISUALES

[Pluralsight](#), [YouTube](#), [CodeProject](#), [StackOverflow](#), [Microsoft](#), [CodePlex](#), [Agile Modelling](#).

Especial mención merece Pluralsight, sitio Web de pago de cursos audiovisuales online. Gracias a la cuenta universitaria ligada a Microsoft, pudimos obtener acceso a todos los cursos de Pluralsight durante tres meses, hecho que contribuyó (en algunos casos notablemente y en otros no tanto) a la profundización en C#, WPF, MVVM, TDD, Scrum, patrones de diseño y más. A título ilustrativo, listamos algunos de estos cursos:

- [XAML Layout in Depth](#)
- [WPF and MVVM: Advanced Model Treatment](#)
- [WPF and MVVM: Test Driven Development of ViewModels](#)
- [Introduction to WPF Custom Controls](#)
- [Creating Modern WPF Apps with MahApps.Metro](#)
- [Building End-to-End Multi-Client Service Oriented Applications](#)
- [C# Learning Path](#)

- [Introduction to Cryptography in .NET](#)
- [How Git Works](#)
- [Introduction to Prism](#)

Vídeos alojados en otras plataformas incluyen:

- [NHDay – Nhibernate Hidden Gems \(Ayende\)](#)
- [Intro To MVVM - Free Webinar From Miguel Castro](#)
- [Learn ASP NET MVC 5](#)

## 13.3. ARTÍCULOS

Incluir la lista total de artículos visitados sería imposible, además de que no se ha llevado un seguimiento de todos ellos. No obstante, cabe destacar los siguientes:

1. [Cost of coupling VS cost of de-coupling](#)
2. [Classic Mistakes Enumerated](#)
3. [SOLID principles](#)
4. [Fluent Interfaces](#)
5. [Inversion of Control Containers and the Dependency Injection pattern](#)
6. [Patterns - WPF Apps With The Model-View-ViewModel Design Pattern](#)
7. [How to use design patterns](#)
8. [NHibernate mistakes and anti-patterns](#)
9. [Customizing the WPF Calendar Controls](#)

---

## 14. ANEXO I: INSTRUCCIONES DE USO

---

### 14.1. CÓMO USAR LA APLICACIÓN

Para poder usar la aplicación hay que contar con un instalador convenientemente configurado para trabajar con un servidor que exista y esté en funcionamiento.

### 14.2. CÓMO USAR LA APLICACIÓN CON VISUAL STUDIO

Hay que modificar las cadenas de conexión a la base de datos en el archivo *App.config* del ensamblado *Gama.Bootstrapper* para que ataquen a una base de datos que exista. El sistema gestor de bases de datos utilizado es MySQL 5.6, por lo que el sistema indicado en las cadenas de conexión debe contar con un servicio MySQL activo, tres bases de datos con los nombres *gama\_atenciones*, *gama\_socios*, y *gama\_cooperacion* deben existir, además de crear el usuario indicado en la cadena de conexión y otorgarle los privilegios adecuados. Luego, hay que crear en cada base de datos una tabla llamada *usuarios* con los campos *nombre* y *password* e incluir, para cada módulo, los usuarios a los que se permita acceso. Estos datos de acceso son los que se introducen en el selector de módulo. Para finalizar, si se desea poder efectuar la acción de restaurar la base de datos desde una copia externa, hay que otorgar permisos especiales para ello.

De forma alternativa, bastará con *establecer como **proyecto de inicio*** alguno de los ensamblados *Gama.Atenciones.Wpf*, *Gama.Socios.Wpf*, *Gama.Cooperacion.Wpf* o *Gama.Extranet* para lanzar los módulos sin pasar por el control de acceso, por lo que no haría falta crear la tabla de usuarios.

En cuanto al servidor de sincronización, hay que asegurarse de habilitar los puertos que se escojan.

Si no se dispone de la carpeta con todos los archivos del proyecto, se puede descargar desde [el repositorio en GitHub del proyecto](#), aunque habrá entonces que añadir un archivo con el nombre *Keys.xaml* al la carpeta *Encryption* del ensamblado *Core*, ya que

por razones de seguridad, la clave que se utiliza para encriptar los datos se excluye del repositorio público. Además, tanto en un caso como en otro habrá que dejar que Visual Studio restaure todos los paquetes a través de Nuget. Este proceso puede tomar algunos minutos. Si se lanza la aplicación sin haberlos restaurado, deberá proceder a hacerlo automáticamente primero, después compilará, y si no tiene lugar ningún error de configuración, se lanzará.

Se muestra a continuación un ejemplo de este archivo con unas claves arbitrarias pero válidas para el algoritmo de encriptación en uso (Rijndael):

---

```
<ResourceDictionary
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:sys="clr-namespace:System;assembly=mscorlib"
    xmlns:local="clr-namespace:Core.Encryption">
    <sys:String x:Key="PassPhrase">560A18CD-GAMA-4CF0-2017-671F9B6B</sys:String>
    <sys:String x:Key="IV">741952hheey66#cs!9hJV887mxx7@8y</sys:String>
</ResourceDictionary>
```

---

Ilustración 60. Keys.xaml

Como se ve, se indica tanto la clave como el vector de inicialización.

### 14.2.1. Preparación de la base de datos

Como puede resultar algo engorroso realizar todas esas tareas, se pone a continuación el código necesario para hacerlo con comodidad. Bastará con cambiar los nombres de usuario y contraseñas para ajustarse a lo que se desee.

---

```
/* Crear los usuarios para localhost y acceso remoto */
CREATE USER 'gama'@'localhost' IDENTIFIED BY 'secret';
CREATE USER 'gama'@'%' IDENTIFIED BY 'secret';

/* Crear las bases de datos */
CREATE DATABASE gama_atenciones CHARACTER SET utf8 COLLATE utf8_spanish_ci;
CREATE DATABASE gama_socios CHARACTER SET utf8 COLLATE utf8_spanish_ci;
CREATE DATABASE gama_cooperacion CHARACTER SET utf8 COLLATE utf8_spanish_ci;

/* Crear esta tabla para cada base de datos */
CREATE TABLE usuarios
(
    nombre VARCHAR(255),
    password VARCHAR(255)
);
INSERT INTO usuarios values ('admin', 'clave');
```

---

---

```
/* Otorgar privilegios */
GRANT ALL ON gama_atenciones.* TO 'gama'@'localhost';
GRANT ALL ON gama_atenciones.* TO 'gama'@'%';
GRANT ALL ON gama_socios.* TO 'gama'@'localhost';
GRANT ALL ON gama_socios.* TO 'gama'@'%';
GRANT ALL ON gama_cooperacion.* TO 'gama'@'localhost';
GRANT ALL ON gama_cooperacion.* TO 'gama'@'%';

/* Otorgar privilegios especiales */
GRANT RELOAD ON *.* TO 'gama'@'localhost';
GRANT CREATE, INSERT, DROP, UPDATE ON mysql.backup_progress TO 'gama'@'localhost';
GRANT CREATE, INSERT, SELECT, DROP, UPDATE ON mysql.backup_history TO
'gama'@'localhost';
GRANT REPLICATION CLIENT ON *.* TO 'gama'@'localhost';
GRANT SUPER ON *.* TO 'gama'@'localhost';
GRANT PROCESS ON *.* TO 'gama'@'localhost';

FLUSH PRIVILEGES;
```

---

#### Ilustración 61. Configuración de la base de datos

Para autogenerar todas las tablas e introducir datos falsos, bastará con ir al archivo *Bootstrapper.cs* del ensamblado *Gama.[Nombre\_Modulo].Wpf* y activar, en el constructor, las banderas para hacer la base de datos y popularla. Se muestra a continuación un ejemplo:

---

```
public Bootstrapper(string title = "SERVICIO DE ATENCIONES") : base(title)
{
    NHibernateSessionFactory._EXECUTE_DDL = true;
    _CLEAR_DATABASE = false;
    _SEED_DATABASE = true;
}
```

---

#### Ilustración 62. Gama.Atenciones.Wpf/Bootstrapper.cs