

---

This is the **published version** of the bachelor thesis:

Hatero Comino, Goran; Lladós Canet, Josep, dir. Aplicación de Gestión de Socios para la asociación AmicsUAB. 2024. (Grau en Enginyeria Informàtica)

---

This version is available at <https://ddd.uab.cat/record/298940>

under the terms of the  license

# Aplicación de Gestión de Socios para la asociación AmicsUAB

Göran Hatero Comino

**Resumen**—AmicsUAB es una asociación situada en la Universidad Autónoma de Barcelona (UAB) que promueve una relación activa entre la Universidad y la Sociedad, con el objetivo de crear un puente de conocimiento, promoviendo así la transferencia de este y ofreciendo Servicios universitarios a sus miembros y empresas. Actualmente, la asociación enfrenta desafíos debido a la existencia de dos bases de datos separadas y un proceso de gestión de socios lento y laborioso. Con este proyecto de final de grado, **AmicsConnected**, se pretende desarrollar una aplicación web, personalizada para la asociación, que agilice los procesos de gestión de socios. Además, se pretende unificar las dos bases de datos actuales en una única base de datos eficiente, capaz de albergar toda la información necesaria sobre los socios. Para lograrlo, se aplica a lo largo del proyecto una metodología de Ingeniería del Software de manera eficaz, garantizando así un producto de alta calidad.

**Palabras clave**—Gestión de socios, web, ingeniería del software, AmicsUAB, Stakeholder, Metodología Agile, MVC, frontend, backend, Sprints, base de datos, interfaz, Vue, Node.

**Abstract**— AmicsUAB is an association based at the Autonomous University of Barcelona (UAB) that promotes an active relationship between the University and Society. Its goal is to create a knowledge bridge, thereby fostering the transfer of knowledge and providing university services to its members and affiliated companies. Currently, the association faces challenges due to the existence of two separate databases and a slow, tedious member management process. This final degree Project, **AmicsConnected**, aims to develop a customised web application for the association to streamline the membership management processes. In addition, the aim is to unify the two current databases into a single efficient database, capable of housing all the necessary information about the members. To achieve this, a software engineering methodology is applied throughout the Project in an efficient manner, thus ensuring a high quality product.

**Index Terms**—Partner management, web, software engineering, AmicsUAB, Stakeholder, Agile method, MVC, frontend, backend, Sprints, database, interface, Vue, Node.



## 1 INTRODUCCIÓN – CONTEXTO DEL TRABAJO

AmicsUAB<sup>[1][2]</sup> se trata de una entidad que fomenta una relación activa entre la Universidad y la sociedad. Su objetivo es ser un puente de conocimiento, promoviendo así la transferencia de este y ofreciendo servicios universitarios a sus miembros y empresas asociadas. Actualmente, cuenta con aproximadamente 5479 miembros, y hasta la fecha actual existen problemas de incongruencias con las dos bases de datos y procesos muy lentos y tediosos a la hora de gestionar sus socios, debido a que se tratan de procesos manuales, entre otros problemas existentes.

Todos estos problemas han llevado a la asociación a replantear sus diversos procesos. Como resultado, desean implementar una aplicación de gestión de socios que aborde todos estos desafíos y, en última instancia, les facilite sus tareas.

## 2 OBJETIVOS DEL PROYECTO

El objetivo general de este proyecto de final de grado, tal y como se ha mencionado en la sección anterior, es diseñar e implementar una aplicación que permita gestionar todos los socios (o miembros) de la asociación AmicsUAB, ubicada en la propia universidad (Universidad Autónoma de Barcelona).

A continuación, se detallan los objetivos específicos, que los podemos categorizar en:

- **Objetivos técnicos**
  - Simplificar y agilizar la mayoría de las tareas que se realizan en base a la gestión de socios.
  - Desarrollar una aplicación capaz de trabajar en un contexto de explotación real.
  - Realizar una aplicación escalable en el tiempo, es decir, que pueda ser mejorada en un futuro.
  - Desarrollar una Base de Datos eficiente que pueda reemplazar las dos bases de datos actuales y que contenga todos los datos necesarios.

---

- E-mail de contacto: G.Hatero@gmail.com
- Mencin realizada: Ingeniería del Software
- Trabajo tutorizado por: Josep Lladós Canet (Ciencias de la Computación)
- Curso 2023/24

- **Objetivos de aprendizaje**
  - Aprender a desarrollar una aplicación web en su conjunto.
  - Comprender y utilizar los diferentes procesos existentes del desarrollo SW aplicando los conceptos aprendidos en la mención de Ingeniería de Software.
  - Desarrollar habilidades de documentación.
  - Obtener una base de conocimiento del framework para frontend, Vue.

### 3 ESTADO DEL ARTE

Una aplicación de gestión de miembros para una organización o entidad no se trata de una idea innovadora, sino que en el mercado actual existen muchas soluciones semejantes [\[3\]](#).

A continuación, se presentan algunos de los productos similares:

#### 3.1 Playoff

Aplicación diseñada para ser utilizada por diversas organizaciones, independientemente de su especialidad. Tiene un carácter más general.

Características destacables:

- **Cobros y Pagos:** Automatiza cobros y gestiona fácilmente los impagos. Además, permite gestionar cuotas recurrentes y utilizar sistemas de pago online, remesas SEPA y descuentos personalizados.
- **Actividades y Eventos:** Ofrece inscripciones y venta de entradas online con tu propia identidad corporativa a través de web y aplicación móvil.
- **Gestión de reservas:** Ofrece a los socios la reserva de horas, citas o clases sin cambiar de programa. Facilita un calendario online para que cada usuario gestione sus reservas.
- **Comunicación y Encuestas:** Permite enviar mensajes de forma masiva y multicanal, y realizar enuestas y votaciones con análisis de resultados e informes.

#### 3.2 Coacha [\[4\]](#)

Coacha es una aplicación de gestión de miembros diseñada específicamente para clubes deportivos. Sus características más destacables son las siguientes:

- **Historial de pagos:** Permite a los miembros conocer en todo momento cuánto y cuándo han pagado
- **Organización por grupos:** Es posible organizar a los socios en tantos grupos como se desee (edad, nivel de habilidad, tipo de miembro, etc.).
- **Recordatorios y alertas:** Se pueden configurar recordatorios personales para cuando se necesite recordar algo.

#### 3.3 Chorus Connection [\[5\]\[6\]](#)

Utilizada específicamente para gestionar los miembros de un coro. Cabe destacar las siguientes funcionalidades:

- **Importaciones y Exportaciones:** Es posible importar y exportar los datos introducidos en la aplicación.
- **Permisos:** La organización puede otorgar a los usuarios acceso solamente a funciones que necesiten.

#### 3.4 Springly [\[7\]](#)

Al igual que Playoff, se trata de una aplicación de carácter general, para diferentes tipos de organización. Lo más destacable de Playoff es el **seguimiento de los pagos** mediante una función de pago para cobrar las cuotas de los miembros.

Además de contar con un sistema de recordatorios automáticos por correo y renovaciones de membresía.

### 4 METODOLOGÍA

El presente proyecto se desarrolla siguiendo una metodología Agile inspirada en Scrum. Cabe destacar que esto no implica la adopción plena de esta metodología, dado que Scrum está concebida para equipos de tamaño considerable, y no resultaría adecuado para un proyecto de una sola persona, como el presente. Sin embargo, se han considerado y aplicado conceptos inherentes a Scrum, como la división del trabajo en Sprints, el Sprint backlog y el backlog list, adaptándolos a las necesidades específicas del proyecto.

Durante el inicio del proyecto se elaboró inicialmente un backlog list<sup>1</sup>, que ha experimentado cambios a lo largo del desarrollo del proyecto, algo prácticamente inevitable en un proyecto de software.

El proyecto se ha desarrollado a través de una serie de Sprints con una duración promedio de 2 a 3 semanas. Al comienzo de cada Sprint, se definía el Sprint backlog<sup>2</sup>, detallando las tareas específicas a realizar durante ese período. Al concluir cada Sprint, se llevaba a cabo un análisis exhaustivo de las tareas completadas y las pendientes. Basándose en este análisis, se tomaban decisiones estratégicas para asegurar la finalización adecuada de las tareas pendientes, minimizando lo máximo posible el impacto en la realización de las demás tareas.

#### Herramienta utilizada:

Para poder aplicar la metodología Agile se decidió utilizar la aplicación **Jira Software**. Esta aplicación ha demostrado ser de una gran utilidad para el seguimiento de todas las actividades realizadas, aquellas pendientes por completar y la gestión de fechas de conclusión de Sprints, entre otras funcionalidades relevantes.

La principal utilidad de Jira Software para este proyecto ha sido la de controlar qué tareas se tenían que completar

<sup>1</sup> Lista de trabajo ordenado por prioridades que se obtiene de la hoja de ruta y requisitos del proyecto [\[8\]](#).

<sup>2</sup> Describe las tareas y actividades específicas que se llevará a cabo durante un Sprint específico [\[9\]](#).

durante el Sprint (como se puede observar en la Ilustración 1) y, en general, tener un control del progreso.

Gracias a este control, se podía determinar antes de acabar el Sprint qué acciones realizar de cara al siguiente Sprint, tales como:

- Extender la fase<sup>3</sup>
- Priorizar unas tareas sobre otras
- Acabar la fase antes de tiempo

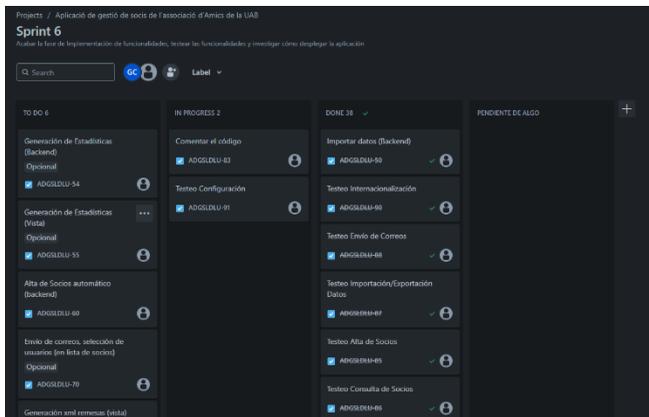


Ilustración 1 Jira Software Board Sprint 6

## 5 PLANIFICACIÓN

El proyecto se ha distribuido en un total de 5 fases:

1. Fase de Análisis de Requisitos
2. Fase de Diseño
3. Fase de Implementación de Funcionalidades
4. Fase de Pruebas y Despliegue
5. Fase de Cierre

En el desarrollo de este proyecto, cada fase se desglosó en diversas subtarefas, las cuales en conjunto constituyen una etapa esencial en el proceso de desarrollo de software. La organización del proyecto en estas fases, típicas de la ingeniería del software, ha sido un factor determinante para alcanzar un producto final satisfactorio y robusto.

Además, en determinadas fases, se concluía con la elaboración de la documentación necesaria para el caso:

- Fase de Análisis de Requisitos
  - Documento de Especificación de Requisitos
- Fase de Diseño
  - SDD\_AmicsConnected
- Fase de Pruebas y Despliegue
  - Informe de Retroalimentación del Cliente

Las diferentes fases se han distribuido en un conjunto de 7 Sprints, siguiendo el diagrama de Gantt de la siguiente Ilustración:

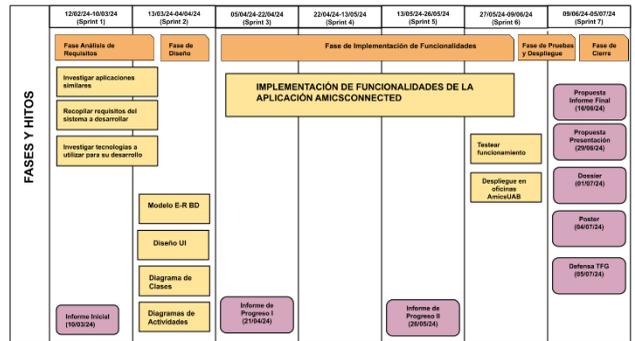


Ilustración 2 Diagrama de Gantt

En términos generales, la planificación se ha seguido sin problemas significativos. No obstante, en algunas ocasiones fue necesario extender la duración de la fase que estaba transcurriendo en ese período de tiempo. Por ejemplo, la Fase de Análisis de Requisitos, que según la planificación inicial debía concluir el 13 de marzo, se prolongó hasta el 23 de marzo debido a la falta de información y al tiempo invertido en la coordinación entre la Stakeholder<sup>4</sup> principal del proyecto y yo.

## 6 ANÁLISIS Y DISEÑO

### 6.1 Análisis de Requisitos

En el marco de cualquier proyecto de Ingeniería del Software, la creación de una aplicación que satisfaga las expectativas del cliente requiere de un estudio y análisis exhaustivo de los requisitos que debe cumplir la aplicación. Adicionalmente, una comprensión total de las funciones que debe realizar la aplicación a realizar simplifica enormemente su posterior programación.

Para recolectar la mayor información posible y definir los requisitos de la aplicación, se llevaron a cabo las siguientes acciones:

- 1) **Investigación de productos similares:** Se realizó un análisis de productos existentes similares, con el objetivo de entender el funcionamiento de este tipo de aplicaciones y recopilar posibles funcionalidades.
- 2) **Cuestionario:** Se diseñó y entregó un cuestionario al Stakeholder. Con el propósito de obtener información sobre la asociación AmicsUAB, la gestión actual de los socios y las expectativas con la nueva aplicación.
- 3) **Entrevista con el cliente:** Finalmente, se llevó a cabo una llamada telefónica con el cliente para definir las funcionalidades de la aplicación deseada. Esta interacción directa permitió una comprensión más profunda de las necesidades del cliente.

<sup>3</sup> El proyecto se ha llevado a cabo en diferentes fases tales como (Fase Análisis de Requisitos, Fase de Diseño, etc.).

<sup>4</sup> **Stakeholder:** Se refiere a cualquier individuo o grupo con interés en una organización y que puede influir o ser influenciado por los resultados de las acciones de dicha organización. En este contexto particular (TFG), el término "organización" se restringe a un solo individuo (el autor de este TFG).

Estas acciones, junto con algunas reuniones iniciales en las instalaciones de AmicsUAB, facilitaron la creación de un documento de especificación de requisitos. Este documento abarca tanto los requisitos funcionales como los no funcionales. A continuación, se destacan los requisitos (funcionales y no funcionales) más relevantes:

- **RF5:** El sistema debe permitir al usuario generar un carné de identificación siguiendo el estilo visual de AmicsUAB en formato PDF.
- **RF9:** Al recibir una petición de baja externa, el sistema debe informar mediante correo electrónico al socio de la petición y programar una fecha de baja. Este correo debe contener un enlace para que el socio pueda cancelarla.
- **RF15:** El sistema debe permitir al usuario importar datos de socios desde un archivo en formato Excel. Esto incluye la selección del archivo, la carga del archivo, la validación del archivo y la actualización de la base de datos.
- **RF19:** El sistema debe permitir al usuario redactar un mensaje que se enviará por correo, adjuntar archivos y aplicar filtros sobre los destinatarios.
- **RF39:** El sistema debe permitir al usuario editar el contenido de las diferentes plantillas existentes (asunto y texto).
- **RNF2:** Los módulos externos que se implementen deben ser open-source.
- **RNF4:** La Interfaz debe integrarse con la imagen corporativa de AmicsUAB.

El documento completo de requisitos se puede consultar en el dossier.

## 6.2 Diseño

Una vez realizada la fase de análisis de requisitos, se procedió a diseñar el sistema. Para ello, se procedió a realizar diversos diagramas, cada uno para un objetivo específico:

### Diagramas de Actividades

Estos diagramas se utilizaron para visualizar y comprender los flujos de trabajo dentro de las funcionalidades del sistema. Se identificaron todas las tareas que deben completarse y las decisiones que se deben tomar durante cada proceso. Cabe destacar que no se crearon para todas las funcionalidades sino para las más relevantes y aquellas cuyo flujo de trabajo no era trivial o complejo: **Baja de socios**, **Consulta y visualización de socios** y **generación de estadísticas**.

### Diagrama de Entidad-Relación

Este diagrama fue crucial para el diseño de la Base de datos de AmicsConnected. Proporcionó una representación visual de las entidades y relaciones, facilitando tanto el diseño como la implementación posterior.

### Diagrama de Clases

Finalmente, el diagrama de clases del sistema permitió modelar la estructura y las relaciones entre los objetos del sistema, proporcionando una representación de las clases y sus relaciones.

Sin embargo, es importante mencionar que el diagrama de clases diseñado refleja los componentes principales, siguiendo el patrón MVC. Por ejemplo, encontramos clases como **SocioController**, **Socio** (Modelo) y **ExportacionDatosView**. De esta forma, se visualiza las interacciones entre los diferentes componentes del sistema.

En la siguiente Ilustración podemos ver un ejemplo:

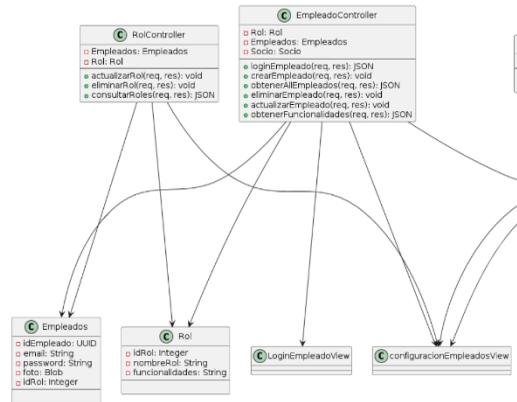


Ilustración 3 Diagrama de Clases MVC

Adicionalmente, se procedió al diseño de las interfaces de usuario, las cuales se pueden consultar en el siguiente [enlace](#), ofreciendo una primera aproximación al diseño visual de las pantallas de AmicsConnected. Este diseño facilitó considerablemente la codificación del frontend, ya que proporcionó una visión clara de la disposición de cada elemento gráfico.

El conjunto de estos elementos se concentra en un documento SDD<sup>5</sup>.

Durante el proceso de creación de los diferentes diagramas, también se llevó a cabo una investigación de herramientas open-source que resultaran útiles para el desarrollo del sistema.

Esta etapa ha representado un pilar fundamental en el desarrollo de este proyecto de fin de grado. La elaboración de los diversos diagramas, junto con el diseño de las interfaces de usuario (UI), ha facilitado un desarrollo más ágil de las funcionalidades de la aplicación. Además, este proceso proporcionó una comprensión más profunda y detallada de las operaciones que debe ejecutar y su alcance.

## 7 PROCESO DE DESARROLLO

### 7.1 Herramientas utilizadas para el desarrollo

#### Visual Studio Code<sup>[10][11]:</sup>

Visual Studio Code se trata de un editor de código fuente desarrollado por Microsoft, y se trata de un software libre y multiplataforma.

Las principales razones por la que se decidió escoger este entorno para desarrollar el código fuente de

<sup>5</sup> **Software Design Description:** Documento que detalla la arquitectura, componentes, interfaces y otras características de un sistema de software.

AmicsConnected son las siguientes:

- 1) **Herramienta familiar:** Este entorno de desarrollo es una herramienta que se ha empleado en diversos proyectos previos a lo largo del grado.
- 2) **Lenguajes de programación:** VS Code cuenta con una compatibilidad integrada con JavaScript, TypeScript y NodeJS y además cuenta con un amplio ecosistema de extensiones para otros lenguajes y entornos de ejecución, como C++, Java, etc.
- 3) **Extensiones:** Se trata de uno de los puntos fuertes del editor. Las extensiones que ofrece nos permiten personalizar y agregar funcionalidad adicional de forma modular y asilada. En definitiva, nos permiten tener una mejor experiencia.

**Gitlab/Git Extensions:**

Gitlab junto a Git Extensions<sup>[12]</sup> han permitido tener un control total y preciso de las diferentes versiones del código fuente en un repositorio alojado en Gitlab.

En el repositorio se alojan principalmente 2 ramas:

- La rama **Main:** Rama en la que se suben cambios y funcionalidades ya implementadas y testeadas. Esta rama aloja la versión más nueva y estable del producto.
- La rama **develop:** Rama para el desarrollo de las diferentes funcionalidades del proyecto, y, donde principalmente se realizan las diferentes pruebas y modificaciones.

Por otro lado, para gestionar los commits<sup>6</sup> realizados y llevar a cabo las operaciones de gestión de ramas y registro de cambios, se ha empleado **Git Extensions**. Esta herramienta consiste en una interfaz gráfica de usuario (GUI) para Git, lo que nos permite controlar Git sin necesidad de emplear una línea de comandos, véase la Ilustración 3.

La capacidad de realizar todas estas operaciones de control de versiones de forma visual, sin necesidad de utilizar la línea de comandos, simplifica significativamente el proceso. La interfaz visual proporciona una experiencia más intuitiva y facilita la comprensión de las acciones realizadas en el repositorio de Git.

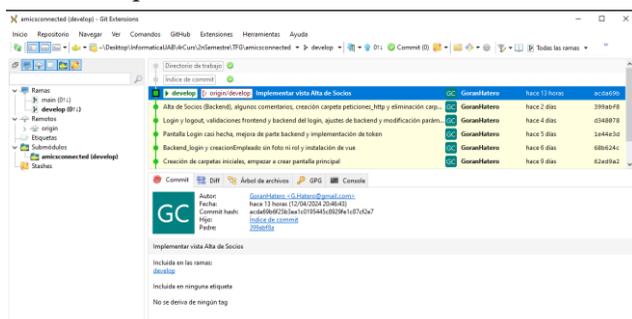


Ilustración 4 Git Extensions GUI

**MySQL Workbench<sup>[13]:</sup>**

Para la gestión de la base de datos de AmicsConnected, se seleccionó MySQL como el sistema de gestión de base de datos. Adicionalmente, se utilizó MySQL Workbench para facilitar la administración, creación y edición de la BD.

El uso principal de MySQL Workbench fue la creación de toda la estructura de la BD, lo que significa, la creación de las distintas tablas con sus respectivos campos, así como la definición de las claves primarias (PK) y claves foráneas (FK). En la siguiente Ilustración se puede observar el entorno de MySQL Workbench y algunas sentencias SQL:

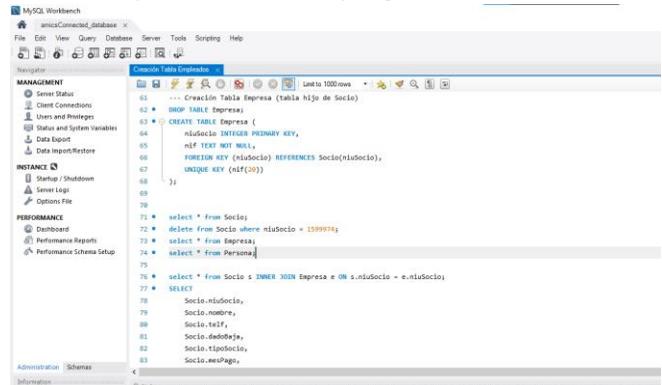


Ilustración 5 Entorno de MySQL Workbench

**7.2 Proceso de desarrollo**

Para el desarrollo del código fuente de esta aplicación web se ha adaptado y seguido el patrón de arquitectura MVC<sup>[14][15]</sup>

Este patrón es muy utilizado en el desarrollo de aplicaciones y páginas web que separa el código según las responsabilidades y manteniendo distintas capas encargadas de unas tareas específicas.

Sus siglas MVC representan las 3 capas en las que está dividida (Modelo-Vista-Controlador):

- **Modelo:** Representa la lógica del negocio, la estructura de datos.
- **Vista:** La interfaz gráfica de usuario (GUI), lo que el usuario ve.
- **Controlador:** Es el cerebro de la aplicación, actúa como un intermediario entre el Modelo y Vista respondiendo a las entradas del usuario.

En la siguiente Ilustración se presenta el esquema general del patrón arquitectónico MVC:

Patrones de Arquitectura MVC

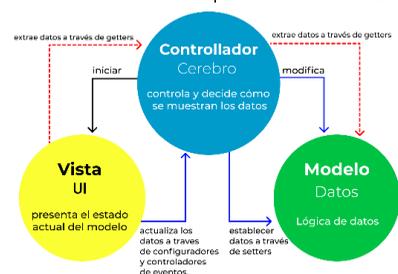


Ilustración 6 Patrón de Arquitectura MVC

<sup>6</sup> **Commit:** Guardar una versión específica de un archivo o conjunto de archivos, posibilitando la recuperación de esa versión en el futuro.

Por otro lado, para llevar a cabo la implementación de las diferentes funcionalidades durante la fase dedicada al desarrollo (Fase de Implementación de funcionalidades) se ha seguido, en términos generales, la estrategia de la siguiente Ilustración:

POR CADA FUNCIONALIDAD:

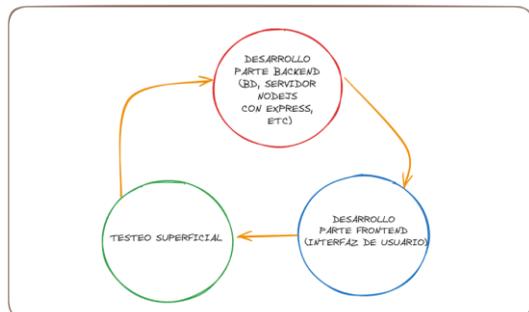


Ilustración 7 Estrategia para el desarrollo de las funcionalidades

### 1) Desarrollo Backend

- Implementación de la lógica del servidor para la funcionalidad requerida. Empleando **NodeJS**.
- Verificar que en el backend se obtiene el resultado esperado con un mínimo de errores.

### 2) Desarrollo Frontend

- Implementación de la parte visual de la funcionalidad, empleando **Vue**.

### 3) Testeo Superficial

- Testear la funcionalidad completa con una colaboración entre frontend y backend para garantizar el funcionamiento de este.

Cabe destacar que la parte de **Testeo Superficial** no se trata de pruebas muy exhaustivas, sino que su objetivo principal es comprobar el funcionamiento básico de la funcionalidad.

### Lenguajes, frameworks y entornos utilizados:

El desarrollo del código fuente de AmicsConnected se ha centrado principalmente en el uso de **JavaScript** como lenguaje principal. Y **SQL** en lo que respecta a la creación de la Base de Datos.

Adicionalmente, para poder facilitar el desarrollo y acelerar su proceso se ha empleado el uso de **NodeJS** y **VueJS**.

### NodeJS - Entorno de ejecución:

NodeJS, a diferencia de VueJS **NO** se trata de un framework<sup>7</sup>, sino que se trata de un **entorno de ejecución**<sup>8</sup> basado en JavaScript.

Una aplicación NodeJS<sup>[18]</sup> se ejecuta en un único proceso y además ofrece un conjunto de operaciones de E/S asíncronas, evitando así el bloqueo del código JavaScript.

Esto quiere decir, que cuando se realiza una operación como leer desde la red, acceder a una BD o al sistema de archivos, no se bloquea el subproceso ni se despiden ciclos del procesador esperando. En su lugar, Node reanuda las operaciones cuando llega la respuesta. Esto es una pieza fundamental para gestionar miles de conexiones simultáneas con un único servidor.

En la documentación oficial de NodeJS nos muestran como ejemplo el típico primer programa de **Hello World**, para tener un primer contacto con el entorno y sus bases. Este primer programa se puede visualizar en la siguiente Ilustración:

```
1 const { createServer } = require('node:http');
2
3 const hostname = '127.0.0.1';
4 const port = 3000;
5
6 const server = createServer((req, res) => {
7   res.statusCode = 200;
8   res.setHeader('Content-Type', 'text/plain');
9   res.end('Hello World');
10 });
11
12 server.listen(port, hostname, () => {
13   console.log(`Server running at http://${hostname}:${port}/`);
14 });
```

Ilustración 8 Hello World en NodeJS

En esta simple aplicación, se incluye el módulo **http** de Node (una de las muchas que ofrece el entorno).

Mediante este módulo creamos un servidor con **createServer()** configurado para escuchar en el puerto y nombre de host especificados. Cuando el servidor está listo, simplemente se llama a una función de devolución de llamada (en este caso, se realiza una **console.log** informando de que el servidor está escuchando).

Cada vez que el servidor recibe una petición, se trata dos objetos **http**, **req** y **res**:

- **Req** → Proporciona los detalles de la solicitud, los encabezados y datos.
- **Res** → Se utiliza para devolver al cliente una respuesta del servidor.

Por lo que cada vez que hagamos desde un cliente una petición a este servidor, nos devolverá una respuesta con un **statusCode** de 200 (lo cuál significa que se trata de una respuesta exitosa) y un texto plano (indicado por el encabezado **Content-Type**) con el siguiente contenido "Hello World".

Lógicamente esto se trata un de ejemplo muy simple de lo que nos puede ofrecer este entorno de ejecución. NodeJS nos ofrece una gran cantidad de módulos para la creación de aplicaciones web y no es de extrañar que NodeJS sea de las herramientas más utilizadas hoy en día en la programación de aplicaciones web.

<sup>7</sup> **Framework**<sup>[16][17]</sup>: Estructura predefinida que ofrece un conjunto de herramientas, librerías y buenas prácticas para el desarrollo de aplicaciones.

<sup>8</sup> **Entorno de ejecución**: Software que permite la ejecución de un código escrito en un lenguaje de programación fuera de su entorno nativo, proporcionando las herramientas y bibliotecas necesarias.

### **VueJS - Frontend Framework:**

Por otro lado, en lo que respecta a la creación de la parte visual de la aplicación, se decidió utilizar **VueJS**<sup>[19]</sup>, framework utilizado para crear interfaces de usuario basado en HTML, CSS y JavaScript estándar.

Proporciona un modelo basado en componentes permitiendo un desarrollo eficiente de cualquier nivel de dificultad.

Las características más destacables de este framework son las siguientes:

- 1) **Reactivo** → Reacción al cambio, es decir, dependiendo de algún evento el sistema reacciona de x manera, permitiendo modificar el comportamiento o vista de nuestra aplicación.
- 2) **Basado en componentes** → Un componente Vue, es un elemento que encapsula código reutilizable. Se asemejan a una pieza de lego, lo cuál tiene sus ventajas ya que aportan al programa:
  - a. **Modularidad**
  - b. **Escalabilidad**
- 3) **Curva de aprendizaje bajo** → Se trata de un framework muy fácil de aprender.

Adicionalmente, los componentes de Vue tienen un ciclo de vida<sup>[20]</sup>, siendo los más importantes:

- 1) **BeforeCreate:** En este punto Vue todavía no ha cargado el componente, por lo tanto, no es posible acceder a ninguna opción, método o datos del componente.
- 2) **Created:** De los más utilizados y es en este punto cuando Vue ya ha cargado completamente el componente.
- 3) **Mounted:** A diferencia del created, aquí tenemos acceso al DOM<sup>9</sup>. El código en este punto se ejecuta cuando se termina de visualizar la vista.
- 4) **Destroyed:** Se ejecuta cuando eliminamos un componente. Esto es cuando cambiamos de vista o cuando se elimina un componente del DOM ocultando mediante **v-if**.

## **8 TESTEO Y DESPLIEGUE**

Por último, al concluir la codificación de la aplicación web, se procedió a realizar pruebas exhaustivas de todas las funcionalidades desarrolladas. Para ello se diseñó un conjunto de pruebas para cada una de las funcionalidades implementadas.

El objetivo principal de estas pruebas fue garantizar la calidad del producto mediante la verificación de diferentes aspectos clave. En primer lugar, se llevaron a cabo **pruebas funcionales** con el objetivo de asegurar que la aplicación

cumpliera con todos los requisitos funcionales especificados. Posteriormente, se realizaron **pruebas de Interfaz de Usuario (UI)** para evaluar el correcto funcionamiento de los elementos gráficos, así como la adecuada presentación de mensajes y otros componentes visuales. Finalmente, se llevaron a cabo **pruebas de tolerancia a fallos** para evaluar la capacidad del sistema de manejar y recuperarse de situaciones de error de manera efectiva.

Ejemplos de la aplicación de pruebas de tolerancia fallos son:

- En la funcionalidad de alta de socios, en los campos del formulario se aplicaron distintos valores erróneos. En la subida de una foto de socio, añadir un archivo que no sea formato de imagen (png o jpeg), etc.
- En la funcionalidad de importación de datos, se comprobó la reacción del programa subiendo un archivo Excel vacío y, posteriormente uno con columnas incompletas.

Una vez testeada la aplicación y corregidos los errores encontrados, se inició una de las tareas más importantes y cruciales del proyecto, su **despliegue** en las oficinas de AmicsUAB. Adicionalmente, se preparó un formulario de Google Forms con el fin de recolectar feedback del cliente sobre el producto entregado y se prepararon las herramientas necesarias.

Las herramientas que se escogieron para hacer posible el despliegue temporal y para que el cliente pudiera testear la aplicación fueron:

- **Aiven** → Para desplegar la BD
- **Render** → Para desplegar la parte del servidor con Node (backend, sin BD) y el frontend.

Aiven y Render son servicios externos de hosting que ofrecen una modalidad gratuita, suficiente para que el cliente pruebe el producto y pueda aportar su opinión al respecto.

Sin embargo, de cara a un contexto de producción real, lo ideal sería no depender de servicios de hosting externos<sup>10</sup> y, en su lugar, configurar un entorno de servidor local en las oficinas que sea robusto, fácil de gestionar y que minimice la necesidad de intervención del cliente.

Para ello se debería de disponer de un servidor con sistema operativo Ubuntu y configurar como servidor web **nginx**<sup>11</sup>. Además de instalar la BD Mysql, las dependencias y recursos necesarios para NodeJS y VueJS y el gestor de procesos **pm2**<sup>12</sup>.

<sup>9</sup> Document Object Model<sup>[21]</sup>, visualiza el documento HTML como un árbol de etiquetas html relacionadas entre sí.

<sup>10</sup> Aunque esto también tiene sus ventajas dependiendo de los intereses de la entidad u organización.

<sup>11</sup> Servidor web de código abierto que ofrece un bajo uso de memoria y alta concurrencia<sup>[22]</sup>.

<sup>12</sup> Gestor de procesos que se utiliza para implementar, administrar y mantener aplicaciones NodeJS en entornos de producción.

Seguidamente, el proceso general a seguir para realizar el despliegue sería:

1. Configurar la Base de Datos
  - a. Crear las tablas, campos y relaciones entre tablas
  - b. Añadir los datos predefinidos que sean necesarios
2. Desplegar Backend
  - a. Clonar el repositorio desde Gitlab
  - b. Configurar las variables de entorno necesarias para la conexión con BD
  - c. Instalar las dependencias mediante **npm install**
  - d. Iniciar la aplicación utilizando PM2
3. Desplegar Frontend
  - a. Instalar las dependencias al igual que el backend.
  - b. Configurar las variables de entorno necesarias.
  - c. Compilar el frontend para producción ejecutando **npm run build**
  - d. Copiar los archivos del directorio **dist** (generado tras la compilación) a la carpeta pública de Nginx
  - e. Editar el archivo de configuración de Nginx para que pueda servir los archivos estáticos del frontend y manejar las solicitudes hacia el backend de NodeJS.

Adicionalmente, en lo que respecta a la seguridad del producto, sería recomendable restringir el acceso a la aplicación web a ciertas direcciones IP, protegiéndola así contra accesos no autorizados.

## 9 RESULTADOS

El desarrollo de la aplicación AmicsConnected ha proporcionado unos resultados significativos. Aunque aún hay mejoras y cambios por realizar, la experiencia que ofrece es satisfactoria y cumple con la mayoría de los requisitos acordados con el cliente.

### 9.1 Logros

Se ha podido desarrollar las siguientes funcionalidades:

- **Gestión de socios:** La aplicación permite realizar una gestión bastante completa de los socios de la organización. Incluyendo la posibilidad de añadir, modificar, eliminar socios, así como la posibilidad de aplicar filtros de búsqueda en la consulta de los socios.
- **Envío de Correos:** Es posible el envío de correos electrónicos (junto con documentos adjuntados) a los socios. Con la posibilidad de filtrar los socios a los que se desea enviar el mensaje.
- **Configuración:** Adicionalmente, es posible configurar diversos parámetros de la aplicación, entre las cuáles destacan la **creación de roles de usuario** y la **personalización de plantillas** de los correos automáticos.
- **Creación de listados:** AmicsConnected dispone de diversos apartados en las que es posible

generar un archivo Excel con información (ej: Exportación de datos o listado tras la consulta de socios).

Además, se ha podido desarrollar una Base de Datos totalmente funcional y capaz de reemplazar las dos bases de datos que actualmente utiliza la asociación.

En el [Anexo 1](#) del presente documento, se incluyen diversas capturas de pantalla de la aplicación, que ilustran las diferentes funcionalidades implementadas.

### 9.2 Retroalimentación del Usuario

Durante la fase de Testeo y Despliegue (véase apartado 8 del presente documento), se entregó un formulario de Google detallado al cliente para obtener retroalimentación sobre el producto final. Esto permitió recopilar información valiosa sobre el nivel de satisfacción del usuario, así como la identificación de errores y áreas de mejora detectadas durante el uso de la aplicación.

Los resultados de este feedback se han recopilado en un Informe de Retroalimentación, en el [Anexo 2](#) del presente documento se presenta un resumen de la retroalimentación recibida por el cliente.

En resumen, la retroalimentación recibida indica que las pruebas realizadas por el usuario han mostrado, en general, un alto nivel de satisfacción con las funcionalidades desarrolladas. Destacando positivamente la facilidad de uso, la usabilidad y el correcto funcionamiento de la mayoría de las características. Además, gracias a este feedback, se ha podido identificar diversos errores y áreas de mejora, lo cuál es fundamental para futuras actualizaciones de la aplicación.

## 10 DESAFÍOS

El desarrollo de la aplicación AmicsConnected ha implicado una serie de retos significativos que han requerido de una mayor atención y dedicación considerable. Estos desafíos, han resultado inevitablemente en una ampliación en el marco temporal previsto para la finalización de la aplicación.

A continuación, se detallan los aspectos del desarrollo de esta aplicación que han presentado un mayor desafío:

### 10.1 Importación masiva de datos

La implementación a nivel de código de la funcionalidad que facilita la incorporación de socios desde un archivo Excel representó un desafío considerable. Esto se debió a la gran cantidad de socios, aproximadamente 9000 filas, entre las cuáles existían socios repetidos.

Por lo tanto, el proceso no se limitó a simplemente leer e importar el contenido del fichero Excel, sino que fue necesario realizar un procesamiento adicional del archivo para eliminar duplicados.

La superación de este desafío se debió a la reestructuración

del código encargado de la importación en partes más pequeñas. Esto no solo mejoró la legibilidad, sino que también permitió realizar las siguientes operaciones de manera más eficiente:

- 1) Lectura de los datos del fichero Excel
- 2) Procesamiento de los datos duplicados (esto permitió reducir el número de filas a 5479)
- 3) Importar los datos del Excel, actualizando o insertando en BD.

## 10.2 Coordinación

La coordinación con la stakeholder, así como la gestión de los diversos problemas surgidos y las nuevas posibles incorporaciones en la aplicación, representaron un desafío debido a la limitación de recursos de tiempo. Por ello, fue crucial coordinarse y organizarse de la mejor forma posible.

Esto es un aspecto común y normal en proyectos de Software, por esta razón en los inicios del proyecto se optó por escoger una metodología de trabajo Agile.

## 10.3 CSS

Otro aspecto, concretamente en el proceso de codificación, que ha representado un desafío ha sido la creación de las interfaces de usuario y la aplicación de CSS o herramientas de estilo propias de la librería **Vuetify**.

Mi experiencia con estos aspectos no es reciente, sin embargo, el área del desarrollo visual de una página web me presenta una mayor dificultad en comparación al desarrollo del backend (parte del servidor).

La superación de este desafío no ha sido una tarea sencilla y, en la actualidad continúa representando una dificultad adicional en el proceso de desarrollo de aplicaciones/páginas web. No obstante, gracias a la práctica constante, recursos de aprendizaje por Internet y aplicando el método de ensayo y error, no ha representado un problema grave ni ha obstaculizado el proceso de desarrollo de AmicsConnected.

## 11 CONCLUSIONES

El proyecto AmicsConnected ha logrado de manera general simplificar y agilizar muchos de los procesos de gestión de socios en la asociación AmicsUAB, centralizando las dos bases de datos previamente existentes en una sola y mejorando algunos de los procesos.

Durante su desarrollo, se presentaron diversos desafíos técnicos y de coordinación. La adopción de una metodología ágil facilitó la adaptabilidad y la rápida respuesta a los cambios y necesidades emergentes a lo largo del proyecto.

Sin embargo, debido a factores como el tiempo limitado, la aparición de nuevas características y las dificultades encontradas, no se ha podido completar todas las funcionalidades deseadas, como la generación de remesas o estadísticas. Y, además, en la última etapa del proyecto, se decidió centrar los esfuerzos en verificar que las funcionalidades

implementadas funcionasen correctamente, asegurando así un producto viable.

Pero, a pesar de no haber logrado todas las funcionalidades, las ya implementadas lo convierten en un producto útil y viable en un contexto real. Cumpliendo además en mayor o menor medida todos los objetivos planteados en un principio.

Adicionalmente, este proyecto me ha permitido adquirir una base inicial en el uso del framework VueJS. Ha sido una experiencia enriquecedora en la que he podido aplicar mis conocimientos informáticos, aprender más sobre el desarrollo web y mejorar tanto mis habilidades técnicas como personales.

En resumen, AmicsConnected no solo ha resuelto diversos problemas actuales de la asociación, sino que también ha representado un reto personal y una oportunidad para mejorar mis habilidades técnicas propias de la Ingeniería del Software.

## 12 MEJORAS Y EVOLUCIÓN

El desarrollo de AmicsConnected no finaliza en este instante, sino que este proyecto sienta las bases para futuras mejoras y la incorporación de nuevas funcionalidades.

Se plantean dos nuevas funcionalidades, que fueron solicitadas en un principio por el cliente. Sin embargo, debido a diversos problemas, limitaciones de tiempo y decisiones estratégicas, no fue posible incorporarlas.

Se trata de:

- **Estadísticas:** Generar datos estadísticos mediante diagramas y la posibilidad de descargarlos aplicando diversos filtros.
- **Remesas:** Generar un archivo xml para su posterior entrega al banco, con las remesas de los socios. Existiendo, además, filtros temporales entre otros.

Adicionalmente, se puede considerar las siguientes mejoras y añadidos:

- **Remesas - Gestión de pagos:** Cargar un archivo Excel generado tras la entrega del XML de remesas al banco correspondiente. Esto permitiría gestionar y controlar los pagos de los socios a través de una tabla en la BD. Además, se sugiere implementar una interfaz interactiva para la visualización de estos datos.
- **Mejoras del código:** El código del backend y del frontend probablemente contiene imperfecciones y áreas de mejora, derivadas de la limitada experiencia con las herramientas utilizadas para su desarrollo (VueJS y NodeJS).
- **Escanéo de códigos de barras:** Los carnés de identificación de los socios cuentan con un código de barras único, que actualmente no tiene una funcionalidad definida. Sería recomendable considerar la posibilidad de escanear estos códigos para identificar rápidamente a los socios. No obstante,

esto podría implicar el desarrollo de una nueva aplicación enfocada en esta funcionalidad.

- **Estilo visual (CSS/Vuetify):** La aplicación del estilo visual mediante CSS y Vuetify no es perfecta, por lo que está abierta a mejoras que podrían hacerla más atractiva.

Las mejoras y nuevas funcionalidades mencionadas son solo algunas de las posibles consideraciones futuras para tener en cuenta. Sin embargo, el ingenio humano es sorprendente e “infinito”, por lo que es probable que existan aún más funcionalidades y mejoras por descubrir.

## AGRADECIMIENTOS

Quisiera dar mi más sincero agradecimiento a todas las personas y a la entidad AmicsUAB que han contribuido en la realización de este Trabajo de Fin de Grado.

En primer lugar, a mi tutor Josep Lladós por su apoyo y los consejos aportados. Sus conocimientos y su seguimiento han sido fundamentales para el desarrollo de este proyecto.

A Gemma Ruiz de la asociación AmicsUAB por su paciencia, colaboración y comprensión a lo largo del trabajo.

Finalmente, quisiera dar un agradecimiento especial a mi familia por su constante apoyo emocional y motivacional.

## BIBLIOGRAFIA

- [1] “AMICS UAB-Benvinguts a AmicsUAB”. AMICS UAB. Disponible: [https://www.amicsuab.cat/portall1/M\\_item-detail.asp?tipo\\_id=122&contentid=1062](https://www.amicsuab.cat/portall1/M_item-detail.asp?tipo_id=122&contentid=1062)
- [2] AmicsUAB. Vídeo presentació AmicsUAB. (15 de marzo de 2023). Disponible: [https://youtu.be/UQUbPbMN4-s?si=bQ\\_goKogmphy-lrT3](https://youtu.be/UQUbPbMN4-s?si=bQ_goKogmphy-lrT3)
- [3] Capterra Australia. 5 Free Membership Management Software. (17 de octubre de 2022). Disponible: <https://www.youtube.com/watch?v=xHLATkfp-zQ>
- [4] “The Best Membership Management Software in 2024 | Springly”. Springly. Disponible: <https://www.springly.org/en-us/features/nonprofit-membership-software/>
- [5] “Admin Tools - Member Management – Chorus Connection”. Chorus Connection. Disponible: <https://www.chorusconnection.com/mem-ber-management>
- [6] “What is Choir Management Software? – Chorus Connection”. Chorus Connection. Disponible: <https://www.chorusconnection.com/what-is-choir-management-software>
- [7] “#1 Sports club management membership software - FREE Trial | Coacha”. #1 Sports club management membership software - FREE Trial | Coacha. Disponible: <https://www.coacha.co.uk/>
- [8] D.Radigan. “Explicación del backlog del producto[+ ejemplos]”. Atlassian. Disponible: <https://www.atlassian.com/es/agile/scrum/backlogs>
- [9] “Qué es un backlog de sprint [y cómo crear uno] | Atlassian”. Atlassian. Disponible: <https://www.atlassian.com/es/agile/project-management/sprint-backlog>
- [10] Microsoft. “Visual Studio: IDE y Editor de código para desarrolladores de software y Teams”. Visual Studio. Disponible: <https://visualstudio.microsoft.com/es/>
- [11] F. Flores. “Qué es Visual Studio Code y qué ventajas ofrece”. OpenWebinars.net. Disponible: <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>
- [12] Visual Studio Marketplace. “Git Extensions - Visual Studio Marketplace”. Extensions for Visual Studio family of products | Visual Studio Marketplace. Disponible: <https://marketplace.visualstudio.com/items?itemName=GitExtensionsApp.VS2022>
- [13] KeepCoding Bootcamps. “¿Qué es MySQL Workbench? | KeepCoding Bootcamps”. KeepCoding Bootcamps. Disponible: <https://keepcoding.io/blog/que-es-mysql-workbench/>
- [14] R. D. Hernandez. “El patrón modelo-vista-controlador: Arquitectura y frameworks explicados”. freeCodeCamp.org.. Disponible: <https://www.freecodecamp.org/espanol/news/el-modelo-de-arquitectura-view-controller-pattern/>
- [15] M. Angel Alvarez. “Qué es MVC”. DesarrolloWeb.com. Disponible: <https://desarrolloweb.com/articulos/que-es-mvc.html>
- [16] V. UNIR. “Framework: qué es, para qué sirve y algunos ejemplos | UNIR FP”. UNIR FP. Disponible: <https://unirfp.unir.net/re-vista/ingenieria-y-tecnologia/framework/>
- [17] E. Bello. “Framework: Qué es, para qué sirve y por qué deberías usarlo”. Thinking for Innovation. Disponible: <https://www.iebschool.com/blog/framework-que-es-agile-scrum/>
- [18] “Node.js – Introduction to Node.js”. Node.js – Run JavaScript Everywhere. Disponible: <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>
- [19] “Introducción – Vue.js”. Vue.js. Disponible: <https://es.vuejs.org/v2/guide/>
- [20] “Vue.js”. Vue.js - The Progressive JavaScript Framework | Vue.js. Disponible: <https://vuejs.org/guide/essentials/lifecycle>
- [21] M. Dev. “¿Qué es el DOM? - Javascript en español”. Lenguaje Javascript | Documentación sobre programación web - Javascript en español - Lenguaje JS. Disponible: <https://lenguajejs.com/javascript/dom/que-es/>
- [22] “¿Qué Es Nginx y Cómo Funciona?” Kinsta®. Disponible: <https://kinsta.com/es/base-de-conocimiento/que-es-nginx/>

## APÉNDICE

### A1. Capturas amicsconnected

**Alta de Socios**  
Por favor, introduzca los datos del nuevo socio

**DATOS GENERALES DEL SOCIO:**



Niu del Socio	Código del Socio	Número del Socio	Email	Teléfono
<input type="text" value="Niu"/>	<input type="text" value="Código del Socio"/>	<input type="text" value="Número del Socio"/>	<input type="text" value="Email"/>	<input type="text" value="Teléfono"/>

NIF	Dirección	Código Postal	Población
<input type="text" value="NIF"/>	<input type="text" value="Dirección"/>	<input type="text" value="Código Postal"/>	<input type="text" value="Población"/>

**IBAN**

Ilustración 9 Pantalla Alta de Socios

**Consulte aquí todos los socios disponibles**

**Filtros Generales:**

Niu del Socio	Código del Socio	Dirección	Código Postal	Población	¿Año gratuito?
<input type="text" value="Niu del Socio"/>	<input type="text" value="Código del Socio"/>	<input type="text" value="Dirección"/>	<input type="text" value="Código Postal"/>	<input type="text" value="Población"/>	<input type="text" value="¿Año gratuito?"/>

NIF	Método de pago	Estado de membresía	Tipo de Socio	¿Aplicar filtros particulares?
<input type="text" value="NIF"/>	<input type="text" value="Método de pago"/>	<input type="text" value="Estado de membresía"/>	<input type="text" value="Tipo de Socio"/>	<input type="button" value="ABRIR FORMULARIO"/>

1	MARIA JOSÉ GARCIA PÉREZ	38	Persona Física	<input type="button" value="CONSULTAR SOCIO"/>	<input type="button" value="EDITAR SOCIO"/>
2	M. José Blaya Valverde	109	Persona Física	<input type="button" value="CONSULTAR SOCIO"/>	<input type="button" value="EDITAR SOCIO"/>
3	Francisco Bogonez Dominguez	110	Persona Física	<input type="button" value="CONSULTAR SOCIO"/>	<input type="button" value="EDITAR SOCIO"/>
4	Maria Montserrat Caro Silanes	175	Persona Física	<input type="button" value="CONSULTAR SOCIO"/>	<input type="button" value="EDITAR SOCIO"/>

Ilustración 10 Pantalla Consulta de Socios

**REDACCIÓN DE CORREOS**

Para:

Asunto:

[SUBIR UN ARCHIVO](#)

Archivos adjuntos:

[▶ ENVIAR CORREO](#)

*Ilustración 11 Pantalla Envío de Correos*

**ROLES DE USUARIO**

Nombre:

Funcionalidades:

<input type="checkbox"/> Alta de Socios	<input type="checkbox"/> Consulta de Socios	<input type="checkbox"/> Modificación de Socios
<input type="checkbox"/> Envío de Correos	<input type="checkbox"/> Importación	<input type="checkbox"/> Exportación
<input type="checkbox"/> Estadísticas	<input type="checkbox"/> Configuración	<input type="checkbox"/> Admin

[CREAR ROL](#)

Roles Existentes:

1 GESTOR		
2 CONSULTOR		

*Ilustración 12 Pantalla Configuración - Gestión Roles*

**CONFIGURACIÓN DE PLANTILLAS**

Seleccione el tipo de Plantilla:

Asunto:   
TEXTO POR DEFECTO

Contenido:   
TEXTO POR DEFECTO

[GUARDAR PLANTILLA](#)

*Ilustración 13 Pantalla Configuración - Plantillas*



Ilustración 14 Carnet de Identificación

	A	B	C	D	E	F
<b>Listado Consulta de Socios</b>						
	Código de Socio	DNI	Nombre	Email	Teléfono	Fecha Alta
12345		58975869F	Göran Hatero Comino	G.Hatero@gmail.com	644424465	23/06/2024

Ilustración 15 Listado Excel tras consulta

## A2. Resumen de la retroalimentación del cliente

Funcionalidad	Puntos fuertes	Problemas identificados	Recomendaciones/Mejoras
Login	Mensajes de errores adecuados. Acceso restringido correcto.	Al acceder a alguna funcionalidad sin iniciar sesión NO se redirige al usuario automáticamente a la pantalla de login	Ninguna.
Alta de Socios	Creación correcta de los 3 tipos de socios. Mensajes de errores adecuados.	Ninguno.	Campos "Género" y "Relación con UAB" deberían ser opciones seleccionables en un desplegable.
Importación de Datos	No se pudo realizar correctamente	Proceso lento, no se pudo verificar la importación correctamente.	Ninguna.
Exportación de Datos	Exportación realizada satisfactoriamente	Ninguno.	Ninguna.
Consulta de Socios	Filtros de búsqueda se aplican correctamente. El listado Excel generado tras consulta es adecuado.	El carné de identificación no sigue al 100% el estilo visual de la entidad. La fotografía del socio sale deformada, se visualiza solamente el primer apellido.	El código de barras no es necesario.
Envío de Correos	Filtros y envíos correctos. Los correos llegan al destinatario correctamente.	Ninguno.	Ninguna.
Configuración	Gestión de roles y empleados correcto, servicio de correo electrónico y personalización de plantillas funcionan adecuadamente.	Ninguno.	Ninguna.
Retroalimentación del producto	Calificación global: 8 Calificación estilo visual: 7 Aplicación amigable e intuitiva.	Los errores identificados en las diferentes funcionalidades.	Mejorar aspectos visuales según el estilo de la entidad.

La aplicación AmicsConnected ha recibido una aceptación positiva con una calificación global de 8 sobre 10. Sin embargo, se han identificado errores y áreas de mejora que deben abordarse para optimizar la experiencia del usuario y mejorar el producto.