



Escuela
Politécnica
Superior

FamSync:

Aplicación para la organización familiar

Grado en Ingeniería Multimedia

Trabajo Fin de Grado

Autor:

Mario Martínez López

Tutor/es:

Alberto Real Fernández



Universitat d'Alacant
Universidad de Alicante

Resumen

La organización de la vida familiar puede parecer algo sencillo desde fuera. Cada persona tiene sus rutinas, sus responsabilidades y sus hábitos. Pero cuando estas se entrelazan con las del resto del hogar, lo que parecía simple se convierte en una tarea compleja, donde coordinar horarios, recordar tareas, hacer la compra o gestionar medicaciones puede convertirse en una fuente constante de estrés.

En muchas ocasiones, estas responsabilidades no están repartidas de forma equitativa, y recaen de forma más intensa sobre ciertos miembros de la familia. Lo más común es que no exista una herramienta clara que centralice toda esta información. Cada familia encuentra su sistema: notas en la nevera, grupos de WhatsApp, agendas personales... pero rara vez existe una solución única y compartida por todos.

A raíz de esta necesidad surge FamSync, una aplicación móvil pensada para ayudar a las familias a organizar su día a día de forma más sencilla, práctica y colaborativa. FamSync permite crear perfiles individuales, gestionar calendarios compartidos, asignar tareas, controlar productos del hogar y planificar listas de la compra. Todo ello desde un entorno visual accesible, diseñado para que cualquier miembro de la familia pueda usarlo, independientemente de su edad o experiencia con la tecnología.

Con esta herramienta, el objetivo es claro: reducir la carga mental, mejorar la coordinación y ofrecer a las familias una forma moderna de organizarse. Porque vivir en familia no tiene por qué ser caótico. Y si lo es, al menos que exista una app que ayude a poner orden.

Resum

L'organització de la vida familiar pot semblar una cosa senzilla des de fora. Cada persona té les seues rutines, responsabilitats i hàbits. Però quan totes aquestes es barregen amb les de la resta de membres de la llar, allò que semblava simple es transforma en una tasca complexa, on coordinar horaris, recordar tasques, fer la compra o gestionar medicacions pot convertir-se en una font constant d'estrés.

En moltes ocasions, aquestes responsabilitats no estan repartides de manera equitativa i recauen, de manera més intensa, sobre alguns membres concrets de la família. El més habitual és que no existisca una eina clara que centralitze tota aquesta informació. Cada família troba el seu propi sistema: notes a la nevera, grups de WhatsApp, agendes personals... però poques vegades hi ha una solució única i compartida per tots.

Davant d'aquesta necessitat naix FamSync, una aplicació mòbil pensada per a ajudar les famílies a organitzar el seu dia a dia de manera més senzilla, pràctica i col·laborativa. FamSync permet crear perfils individuals, gestionar calendaris compartits, assignar tasques, controlar productes de la llar i planificar llistes de la compra. Tot açò des d'un entorn visual accessible, dissenyat perquè qualsevol membre de la família el puga utilitzar, siga quina siga la seua edat o experiència amb la tecnologia.

Amb aquesta eina, l'objectiu és clar: reduir la càrrega mental, millorar la coordinació i oferir a les famílies una forma moderna d'organitzar-se. Perquè viure en família no hauria de ser caòtic. I si ho és, almenys que existisca una aplicació que ajude a posar ordre.

Summary

From the outside, managing family life might seem simple. Each person has their own routines, responsibilities, and habits. But when these intertwine with those of the rest of the household, what once appeared to be a straightforward task quickly becomes complex. Coordinating schedules, remembering chores, doing the shopping, or managing medication can become a constant source of stress.

In many cases, these responsibilities are not distributed equally, and often fall more heavily on certain family members. Most families do not have a single tool to centralize all this information. Each household finds its own way—notes on the fridge, WhatsApp groups, personal planners—but rarely is there a unified and shared solution.

FamSync was born out of this need: a mobile application designed to help families organize their daily lives in a more practical, collaborative, and accessible way. FamSync allows users to create individual profiles, manage shared calendars, assign tasks, track household items, and prepare shopping lists. All of this is done through a user-friendly interface designed to be accessible to any family member, regardless of age or technological experience.

The goal is clear: to reduce mental load, improve coordination, and provide families with a modern way to stay organized. Because living in a family doesn't have to be chaotic—and if it is, at least there should be an app to bring some order.

Motivación, justificación y objetivo general

Cuando llegó el momento de decidir el tema para el Trabajo de Fin de Grado, tenía claro que quería desarrollar algo que tuviera utilidad en el día a día y que resolviera problemas reales. Desde el principio me motivaba la idea de crear una aplicación que pudiera ser útil para familias, no solo por cuestiones personales, sino porque observaba cómo, en muchas casas, la gestión de las tareas, horarios y responsabilidades puede convertirse en un auténtico caos si no se cuenta con una buena organización.

En el contexto actual, donde los miembros de una misma familia pueden tener rutinas muy distintas y vivir con agendas cada vez más apretadas, es fácil que se generen olvidos, malentendidos o duplicidades en las tareas cotidianas. Coordinar quién recoge a los niños, quién se encarga de las compras, cuándo hay que acudir a una cita médica o cuándo toca renovar una receta, son solo algunos ejemplos de situaciones comunes en las que una mejor organización puede marcar la diferencia.

Además, muchas aplicaciones móviles actuales de organización personal no están pensadas específicamente para el ámbito familiar, y suelen centrarse en el uso individual. Esto hace que sean inservibles en el ámbito familiar .

Por todo esto, decidí crear FamSync, una app móvil pensada para hacer más fácil la organización familiar, reuniendo en un solo sitio todo lo necesario y con un manejo sencillo. Mi intención con esta aplicación es no solo mejorar la coordinación entre los miembros de la familia, sino también animar a que todos participen activamente en las tareas compartidas.

El objetivo principal del proyecto es diseñar y desarrollar una aplicación móvil que permita gestionar de forma colaborativa tareas, eventos, recordatorios y otra información importante del día a día, ofreciendo así una solución práctica a una necesidad real en la vida familiar.

Agradecimientos

Quiero agradecer en primer lugar a mi padre y a mi hermana, por su apoyo incondicional durante todo este proceso y, en especial, durante este último año lleno de baches. Gracias por los ánimos en los momentos difíciles y por estar siempre ahí.

En segundo lugar, me gustaría dedicar este trabajo a mí mismo, por enseñarme a no rendirme, por las noches llenas de dudas y los retos superados. Este proyecto ha sido una prueba tanto técnica como personal, y mirar atrás con orgullo por el camino recorrido también forma parte del aprendizaje.

También me gustaría agradecer a mis antiguos compañeros de trabajo, que fueron quienes me metieron de lleno en el mundo del desarrollo de apps. Gracias a ellos descubrí lo mucho que me gustaba este campo y aprendí un montón de cosas que, sin duda, me han ayudado a lo largo de este proyecto.

No puedo olvidarme de dar las gracias también a mi tutor, Alberto, por su orientación y disponibilidad durante el desarrollo del TFG, así como por sus consejos, que me ayudaron a mejorar tanto el enfoque como la estructura del trabajo.

Por último, quiero dar las gracias a mis amigos por los cafés, las energéticas, las charlas absurdas, las ideas locas y por hacer este viaje mucho más llevadero.

Citas

Ten miedo, pero hazlo de todos modos. Lo importante es la acción. No tienes que esperar a tener confianza. Simplemente hazlo y, con el tiempo, la confianza te seguirá.

Carrie Fisher

Niños rebeldes, os insto a luchar contra la pomposa mancha de la conformidad con la que buscan sofocar vuestra gloria.

Russell Brand

Solo cuando perdemos todo, somos libres de hacer lo que queramos.

Tyler Durden

Índice de contenidos

Resumen	2
Resum	3
Summary	4
Motivación, justificación y objetivo general	5
Agradecimientos	6
Citas	7
Índice de contenidos	8
Índice de figuras	10
Índice de tablas	11
1. Introducción	12
1.1 Objetivos	13
1.2 Metodologías	14
1.2.1 Fases del desarrollo bajo Scrum	14
1.2.2. Iteración y adaptación	15
1.2.3. Herramientas de gestión	15
2. Marco contextual y antecedentes tecnológicos	17
2.1 Análisis de soluciones existentes	17
2.2 Tecnologías y metodologías existentes	22
2.2.1. Tecnologías para el Frontend	22
2.2.2 Frameworks más populares para el desarrollo frontend	22
2.2.3. Tecnologías para el Backend	25
2.2.4. Frameworks más utilizados para el desarrollo backend	25
2.2.5. Tecnologías para APIs	28
2.2.6. Herramientas y estándares para el desarrollo y prueba de APIs	28
3. Diseño y Programación de FamSync	31
3.1 Arquitectura	32
3.1.1. Esquema de comunicación	32
3.1.2 FrontEnd (Cliente - Aplicación Móvil)	33
3.1.3 Backend (Servidor - API REST)	33
3.1.4 Base de datos (Persistencia de datos)	34
3.2 Gestión de Perfiles	34
3.2.1 Implementación técnica	38
3.3 Gestión de Inventario	39
3.3.1 Asignación de productos a listas de la compra	40
3.3.2 Visualización en formato Almacén	42
3.3.3 Implementación técnica	45
3.4 Gestión de Tareas	46
3.4.1 Asignación de tareas a perfiles	47
3.4.2 Visualización en formato Agenda	49
3.4.3 Implementación técnica	53

3.5 Gestión de Eventos	54
3.5.1 Asignación de tareas a eventos	55
3.5.2 Visualización en formato Calendario	57
3.5.3 Implementación técnica	60
3.6 Pruebas y despliegue	60
3.6.1 Pruebas realizadas	61
3.6.2 Herramientas utilizadas durante las pruebas	62
3.6.3 Despliegue	62
4. Casos de Uso	63
4.1 Asignación de una tarea a varios perfiles	63
4.2 Planificación de un evento a partir de una tarea	66
4.3 Creación de una lista de la compra semanal	67
4.4 Consulta del calendario familiar en vista mensual	68
5. Conclusiones y trabajo futuro	70
Referencias	72
Anexo I. Enlaces	76

Índice de figuras

Figura 1. Cozi Family Organizer. (Fuente: Google Play).....	19
Figura 2. FamilyWall (Fuente: Google Play).....	20
Figura 3. OurHome (Fuente: Google Play).....	21
Figura 4. TimeTree (Fuente: Google Play).....	22
Figura 5. Arquitectura cliente-servidor.....	33
Figura 6. Formulario Login.....	36
Figura 7. Vista de selección de perfil.....	38
Figura 8. Vista de listas de productos.....	43
Figura 9. Vista General.....	44
Figura 10. Vista Individual.....	45
Figura 11. Creación de tarea.....	49
Figura 12. Asignación de perfiles.....	50
Figura 13. Vista general tareas.....	52
Figura 14. Vista filtrada tareas.....	53
Figura 15. Creación de evento mediante tarea.....	57
Figura 16. Página de eventos diarios.....	58
Figura 17. Página de todos los eventos.....	59
Figura 18. Página de detalles de evento.....	60
Figura 19. Formulario creación de tarea.....	65
Figura 20. Visualización nueva tarea.....	66
Figura 21. Evento creado mediante Tarea.....	67
Figura 22. Creación de lista.....	68
Figura 23. Visualización de lista.....	69
Figura 24. Visualización de eventos.....	70

Índice de tablas

Tabla 1. Estructura de usuario.....	37
Tabla 2. Estructura de perfil.....	37
Tabla 3. Estructura de producto.....	41
Tabla 4. Estructura de lista.....	42
Tabla 5. Estructura de tarea.....	47
Tabla 6. Estructura de Evento.....	55

1. Introducción

En la actualidad, la gestión del entorno familiar se ha vuelto cada vez más compleja debido a la diversidad de actividades, tareas y responsabilidades que deben coordinarse entre los miembros del hogar. Diversos estudios señalan que la digitalización puede mejorar la organización familiar y fomentar la colaboración entre sus miembros [1]. En este sentido, las aplicaciones móviles se han consolidado como herramientas clave para facilitar la planificación diaria y la comunicación en tiempo real [2].

La realidad, sin embargo, muestra que la comunicación y la coordinación dentro del entorno familiar no siempre son óptimas. Es habitual que se produzcan olvidos, solapamientos de horarios, malentendidos sobre quién debe hacer qué, o incluso la duplicación de tareas por falta de una planificación común. Esto puede derivar en tensiones dentro del hogar y en una carga mental desproporcionada para aquellos miembros que asumen más responsabilidades o que intentan mantener el orden general del día a día.

En los últimos años, el avance de la tecnología móvil y la proliferación de aplicaciones digitales han transformado profundamente la forma en que las personas gestionan su vida cotidiana. Existen apps para organizar tareas, programar recordatorios, coordinar citas, e incluso para automatizar rutinas domésticas, lo que ha facilitado en gran medida la planificación individual.

Sin embargo, aunque existen numerosas herramientas digitales enfocadas a la productividad y la gestión de tareas o calendarios, la mayoría de estas aplicaciones están diseñadas desde una perspectiva individual. Es decir, están pensadas para que una sola persona controle su tiempo o sus listas, sin contemplar la complejidad de una unidad familiar, donde es necesario compartir, coordinar y adaptarse continuamente a cambios en los planes comunes.

Además, las familias actuales son muy diversas y requieren soluciones flexibles, que se adapten tanto a núcleos numerosos como a familias pequeñas, con niños, adultos mayores o personas dependientes. Cada miembro puede tener un rol distinto en la organización del hogar, por lo que contar con una herramienta que les permita colaborar de manera sencilla, visual y estructurada puede marcar la diferencia en la calidad de vida de todos los integrantes.

Ante esta situación, surge la necesidad de contar con una solución digital pensada específicamente para familias: una herramienta que permita a todos los miembros acceder a la misma información, colaborar en tiempo real y adaptarse a los cambios del día a día de forma

flexible. Una aplicación que facilite la coordinación, reduzca la carga mental asociada a la organización y contribuya a un entorno familiar más equilibrado y funcional.

De esta necesidad nace FamSync, una aplicación móvil orientada a la gestión integral del entorno familiar. A través de una interfaz sencilla e intuitiva, FamSync permite organizar calendarios compartidos, asignar tareas, controlar productos del hogar y mantener una comunicación más fluida entre los miembros de la familia.

1.1 Objetivos

El objetivo principal de este TFG [3] es diseñar y desarrollar una aplicación móvil destinada a facilitar la gestión familiar, centralizando el acceso a información clave como horarios, tareas, productos y listas de compras, y fomentando así una mejor comunicación y coordinación entre los miembros del hogar. Con esta herramienta se pretende optimizar la organización del entorno familiar, reduciendo la carga mental asociada a la planificación del día a día y promoviendo una participación equitativa en las responsabilidades comunes.

Para alcanzar esta meta, se establecen una serie de objetivos específicos que abarcan distintos módulos funcionales, los cuales trabajarán de forma integrada dentro de la aplicación. En primer lugar, se plantea la implementación de un inventario compartido, intuitivo y visualmente accesible, que permita a los usuarios registrar, organizar y consultar los productos que forman parte del hogar. Este módulo permitirá, además, asociar productos a distintas tiendas, controlar cantidades y crear listas de la compra compartidas entre varios perfiles. Se priorizará su desarrollo en las primeras etapas del proyecto, ya que constituye uno de los pilares sobre los que se estructura el uso cotidiano de la aplicación.

Otro de los módulos clave será el sistema de gestión de tareas, orientado a la asignación de responsabilidades a distintos miembros de la familia. Este sistema permitirá crear tareas individuales o grupales, clasificarlas por categorías, establecer niveles de prioridad y hacer seguimiento de su estado (pendiente, en progreso o completada). Su objetivo principal es facilitar la organización y el reparto equitativo de las labores diarias, promoviendo la colaboración activa entre los usuarios y reduciendo posibles malentendidos o sobrecargas.

Complementariamente, se desarrollará un sistema de gestión de eventos y calendario, mediante el cual los usuarios podrán planificar y visualizar de forma clara los compromisos individuales y familiares. Este calendario ofrecerá vistas por día, semana y mes, e integrará los

eventos asignados a cada perfil. Además, se sincronizará con el módulo de tareas, de forma que toda actividad con fecha asociada quede automáticamente reflejada en el calendario, mejorando así la coherencia y planificación global de la familia.

Asimismo, se prestará especial atención a la accesibilidad y facilidad de uso de la aplicación. Se adoptará un diseño centrado en el usuario, aplicando principios de usabilidad y accesibilidad que aseguren que personas de todas las edades y niveles de alfabetización digital puedan utilizar FamSync de forma autónoma. Esto incluirá el uso de tipografías legibles, botones de gran tamaño, iconografía intuitiva y una estructura de navegación clara y coherente. El objetivo es que cualquier miembro de la familia, incluidos adultos mayores, puedan utilizar la aplicación sin dificultad.

De forma transversal, todos los módulos estarán diseñados para facilitar la interacción colaborativa entre los perfiles registrados, fomentando la participación activa y el acceso equitativo a la información. Esta visión integral pretende ofrecer una solución digital completa y adaptable a distintos tipos de familia, consolidando a FamSync como una herramienta útil y versátil en la gestión del entorno familiar.

1.2 Metodologías

Para poder alcanzar los objetivos planteados y garantizar una planificación eficiente del desarrollo, se ha optado por emplear una metodología ágil. En concreto, se ha seleccionado Scrum, debido a su flexibilidad y su capacidad para adaptarse a los cambios durante el proceso de desarrollo. Este enfoque se ajusta especialmente bien a proyectos tecnológicos, como el desarrollo de aplicaciones móviles, ya que permite realizar entregas incrementales, incorporar mejoras constantes y responder con rapidez a los nuevos requisitos que puedan surgir.

1.2.1 Fases del desarrollo bajo Scrum

El desarrollo de FamSync se ha organizado en sprints de dos semanas, durante los cuales se abordan distintos ciclos de trabajo divididos en varias fases clave:

- Planificación del sprint, en esta etapa se definen las historias de usuario, entendidas como descripciones detalladas de funcionalidades que debe incorporar la aplicación. Estas historias están centradas en aspectos esenciales como la gestión de horarios, tareas, y servirán como guía para el trabajo a desarrollar durante ese sprint.

- Desarrollo, durante esta fase se implementan las funcionalidades planificadas. Aunque Scrum está pensado para equipos, en este proyecto el desarrollo ha sido realizado por una sola persona, por lo que se ha adaptado el enfoque de forma individual. No obstante, se han respetado los principios clave del método, asignando cada tarea en función de su prioridad e impacto dentro del funcionamiento general de la aplicación.
- Revisión del sprint, al finalizar cada sprint se realiza una revisión del trabajo implementado. Esta evaluación incluye tanto la parte técnica (correcto funcionamiento de las funcionalidades) como la validación de la experiencia de usuario (UX), con el objetivo de asegurar que el producto evoluciona conforme a las necesidades iniciales y ofrece una interfaz clara y eficaz.
- Retrospectiva, una vez finalizada la revisión, se lleva a cabo una retrospectiva, entendida en este contexto como una reflexión individual orientada a detectar qué ha funcionado bien y qué aspectos pueden mejorarse en futuros sprints. Esta etapa resulta fundamental para optimizar el proceso de desarrollo de manera progresiva y constante.

1.2.2. Iteración y adaptación

Una de las principales ventajas de Scrum es su enfoque iterativo. Durante cada sprint, se podrá evaluar el progreso y realizar ajustes según las necesidades. Si alguna funcionalidad requiere modificaciones o si surgen nuevas necesidades, se podrá incorporar en el siguiente sprint, sin interrumpir el flujo de trabajo.

Este enfoque permite una retroalimentación constante con el proyecto, lo que es crucial cuando se trata de aplicaciones móviles, ya que la experiencia del usuario y la usabilidad son aspectos clave que deben mejorarse continuamente.

1.2.3. Herramientas de gestión

Para llevar un control eficiente del desarrollo y la gestión del proyecto, se utilizarán herramientas que faciliten la planificación, seguimiento y ejecución de las tareas, tales como:

- *Trello*¹: esta herramienta será utilizada para la gestión de tareas y la planificación de los sprints. A través de tableros, listas y tarjetas, se organizan las funcionalidades que deben desarrollarse en cada sprint. Cada tarea será asignada y etiquetada según su prioridad y estado.

¹ <https://trello.com/>

- *GitHub*²: esta herramienta se usará para el control de versiones y colaboración (aunque en este caso soy el único desarrollador), se utilizará GitHub para mantener el código fuente organizado y gestionado. Además, permite hacer un seguimiento del progreso mediante issues y pull requests.

1.2.4. Justificación de la metodología elegida

La elección de Scrum [6] se debe a la necesidad de realizar ajustes rápidos a medida que avanzaba el desarrollo de la aplicación, especialmente dado que el proyecto abarca diferentes funcionalidades interconectadas, como la gestión de horarios, tareas y recordatorios. Scrum permite trabajar de forma incremental, entregando versiones funcionales de la aplicación al final de cada sprint. Esto es especialmente útil para obtener retroalimentación temprana, realizar ajustes y mejorar continuamente la aplicación.

Además, la metodología ágil facilita la adaptación continua, lo que es fundamental en un proyecto donde las necesidades pueden cambiar con el tiempo, como puede ocurrir con una aplicación móvil en constante evolución.

² <https://github.com>

2. Marco contextual y antecedentes tecnológicos

En el desarrollo de cualquier aplicación móvil, es fundamental realizar un análisis previo del contexto en el que se enmarca el proyecto. Este capítulo tiene como objetivo estudiar el estado actual de las tecnologías de desarrollo y aplicaciones relacionadas con la gestión familiar, así como identificar tendencias, soluciones existentes y herramientas utilizadas en el desarrollo de este tipo de software.

Para ello, se llevará a cabo una revisión de aplicaciones similares a FamSync, analizando sus funcionalidades y enfoques, con el fin de identificar fortalezas y limitaciones. Asimismo, se explorarán las tecnologías más utilizadas en el desarrollo de aplicaciones móviles, incluyendo frameworks, bases de datos y arquitecturas recomendadas para este tipo de sistemas.

Además, se examinarán estudios y trabajos previos que aborden la problemática de la organización familiar mediante soluciones digitales, así como tendencias actuales en la industria del software que puedan influir en la implementación de FamSync. Este análisis permitirá justificar las decisiones tomadas en el desarrollo de la aplicación, asegurando que se basa en prácticas consolidadas y ofreciendo una propuesta diferenciadora en el mercado.

2.1 Análisis de soluciones existentes

Antes de desarrollar una solución innovadora, es fundamental conocer las alternativas existentes en el mercado. En este apartado se analizarán diversas aplicaciones que abordan la gestión familiar, con el objetivo de identificar sus fortalezas y debilidades, y así extraer ideas y mejoras para el desarrollo de FamSync.

Actualmente existen diversas aplicaciones centradas en la organización familiar. Cozi, por ejemplo, permite crear calendarios compartidos y listas de tareas familiares, y ha sido destacada por su sencillez de uso [7]. FamilyWall ofrece funcionalidades similares con un enfoque visual más moderno. Estas soluciones reflejan una tendencia creciente hacia la centralización de la información familiar en entornos móviles [8].

Una de las aplicaciones más conocidas es Cozi Family Organizer [9], diseñada específicamente para facilitar la coordinación entre los miembros de una familia. Su propuesta se centra en ofrecer un calendario compartido donde todos los usuarios pueden ver y editar eventos, así

como listas de tareas, recordatorios y un planificador de comidas. La interfaz de Cozi resulta clara y accesible, lo que ha contribuido a su popularidad entre familias con diferentes niveles de experiencia tecnológica. Sin embargo, algunas funcionalidades avanzadas, como la personalización de recordatorios o la exportación de calendarios, solo están disponibles mediante una suscripción premium, lo cual puede suponer una barrera para ciertos usuarios.

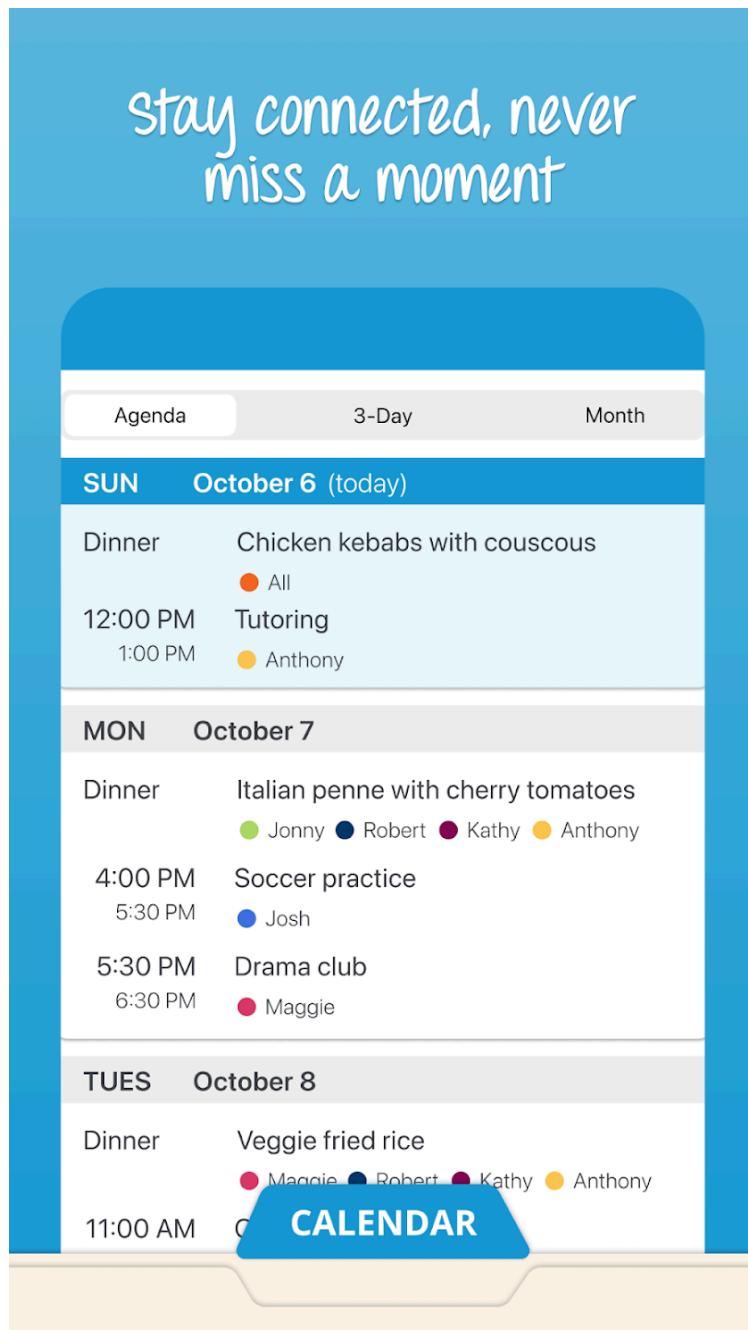


Figura 1. Cozi Family Organizer. (Fuente: Google Play)

Otra propuesta interesante es FamilyWall [10], que amplía el enfoque tradicional de organización al incluir funcionalidades como la localización en tiempo real de los miembros de la familia mediante GPS, la posibilidad de compartir fotografías y la mensajería interna. Esta aplicación busca no solo organizar, sino también reforzar la conexión entre los integrantes del hogar a través de herramientas colaborativas. La integración de distintos módulos en una sola aplicación puede ser una ventaja importante, aunque también puede generar cierta sobrecarga visual en dispositivos con pantallas pequeñas. Además, al igual que en otros casos, algunas funcionalidades requieren acceso premium.



Figura 2. FamilyWall (Fuente: Google Play)

En cuanto a OurHome [11], se trata de una aplicación orientada principalmente a la asignación de tareas del hogar, incorporando un sistema de puntos y recompensas que resulta especialmente útil para fomentar la implicación de los más jóvenes. La aplicación permite también la gestión de listas de la compra, calendarios y mensajes internos. Su enfoque gamificado convierte las responsabilidades domésticas en retos asumibles, motivando a niños y adolescentes a participar en las tareas familiares. No obstante, la herramienta presenta ciertas limitaciones en cuanto a la gestión de eventos o el control de productos, ya que su estructura está claramente centrada en la organización de tareas.

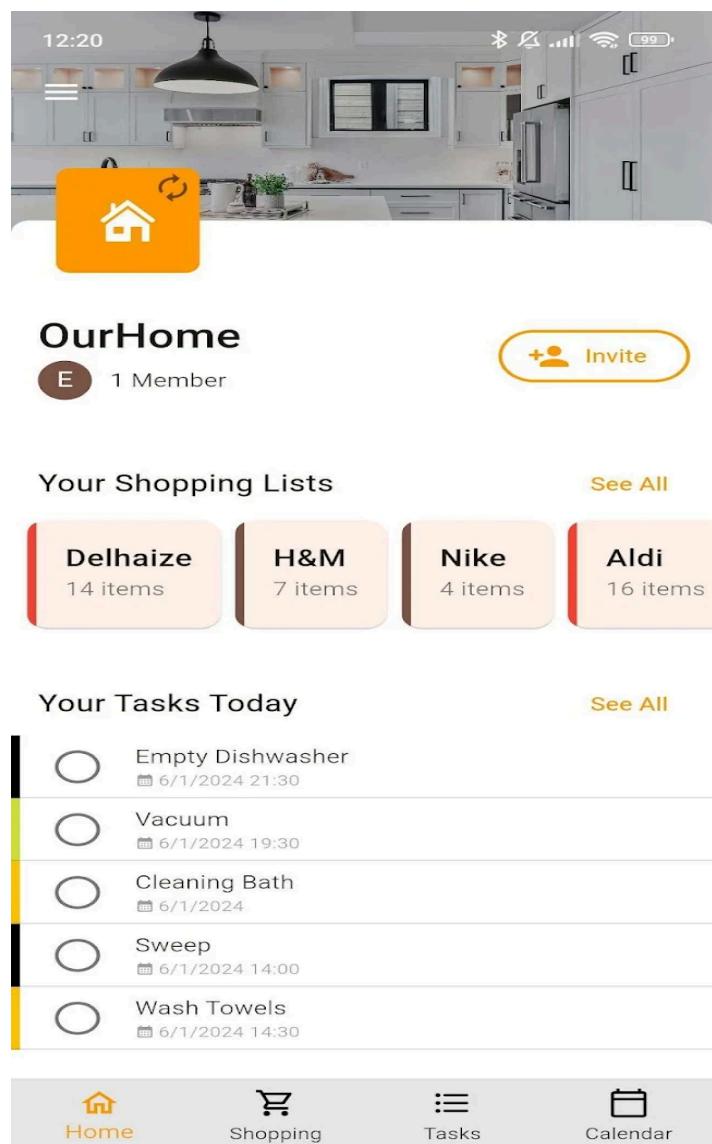


Figura 3. OurHome (Fuente: Google Play)

Por último, TimeTree [12] ofrece una experiencia centrada casi exclusivamente en la gestión de calendarios compartidos. Es ampliamente utilizada tanto por familias como por equipos de trabajo, gracias a su diseño limpio y su facilidad de uso. La aplicación permite crear múltiples calendarios, invitar a colaboradores y mantener conversaciones dentro de cada evento. Aunque su especialización en la coordinación de horarios la convierte en una herramienta muy eficaz para la planificación diaria, carece de funcionalidades complementarias como la gestión de tareas, productos o recompensas, lo que la limita cuando se busca una solución más completa para el entorno familiar.



Figura 4. TimeTree (Fuente: Google Play)

En resumen, las aplicaciones analizadas responden a necesidades específicas dentro del ámbito de la organización del hogar, y cada una presenta fortalezas claras según el enfoque adoptado. Algunas se centran en la coordinación de eventos mediante calendarios compartidos, otras en la asignación de tareas o en la comunicación entre miembros de la familia. Sin embargo, tienden a presentar estas funcionalidades de forma aislada, sin una integración profunda entre módulos que permita una gestión verdaderamente global del entorno familiar. Esta revisión pone de manifiesto tanto los logros como las carencias de las soluciones actuales, y permite identificar oportunidades de mejora en el diseño de nuevas herramientas orientadas a la vida familiar.

2.2 Tecnologías y metodologías existentes

2.2.1. Tecnologías para el Frontend

El desarrollo del frontend abarca no solo el diseño visual y la interfaz con la que interactúa el usuario, sino también toda la lógica que se ejecuta en el lado del cliente, incluyendo la navegación entre pantallas, la gestión del estado de la aplicación, las validaciones de entrada, la interacción con APIs y la experiencia de usuario en general. Se trata, por tanto, de una parte fundamental del sistema, ya que determina tanto el aspecto como el comportamiento de la aplicación desde la perspectiva del usuario final.

Para este propósito, existen diversos lenguajes de programación y frameworks que permiten construir interfaces atractivas, eficientes y funcionales.

Lenguajes de programación más utilizados:

- *JavaScript* [13]: lenguaje fundamental para el desarrollo web, que permite agregar interactividad y dinamismo a las aplicaciones. Es compatible con la mayoría de los navegadores y es la base de muchos frameworks modernos.
- *Dart* [14]: un lenguaje optimizado para aplicaciones móviles y web, desarrollado por Google. Es el lenguaje principal de *Flutter*, permitiendo la creación de aplicaciones nativas multiplataforma.
- *Kotlin* [15] y *Java* [16]: son los lenguajes predominantes para el desarrollo nativo en Android. *Kotlin*, más moderno, ofrece mayor concisión y seguridad frente a errores, mientras que Java cuenta con una gran comunidad y madurez.
- *Swift* [17]: lenguaje de programación de Apple diseñado específicamente para el desarrollo de aplicaciones en iOS, conocido por su rendimiento y seguridad.

2.2.2 Frameworks más populares para el desarrollo frontend

A la hora de desarrollar aplicaciones modernas, la elección del framework frontend es crucial, ya que influye directamente en la experiencia del usuario, el rendimiento y la mantenibilidad del proyecto. A continuación, se describen algunos de los frameworks más populares, sus ventajas, desventajas y consideraciones clave.

Flutter [18] es un framework de código abierto desarrollado por Google, diseñado para crear aplicaciones nativas para Android, iOS, web y escritorio desde una única base de código. Utiliza el lenguaje Dart y destaca por ofrecer un alto rendimiento y una interfaz de usuario consistente entre plataformas.

Ventajas:

- Interfaz nativa fluida y personalizable.
- Desarrollo verdaderamente multiplataforma con un solo código.
- Amplio soporte de widgets y herramientas propias.

Desventajas:

- Curva de aprendizaje para quienes no conocen Dart.
- Mayor tamaño de las aplicaciones generadas.
- Comunidad aún en crecimiento frente a otras tecnologías.

Su elección resulta adecuada cuando se busca una experiencia unificada y de alto rendimiento entre plataformas sin sacrificar demasiado el aspecto visual o nativo.

React Native [19], desarrollado por Meta (Facebook), permite crear aplicaciones móviles utilizando React y JavaScript. Se basa en el concepto de compartir lógica de negocio con componentes reutilizables, y traduce estos componentes a widgets nativos.

Ventajas:

- Uso de JavaScript, un lenguaje ampliamente conocido.
- Amplia comunidad y abundantes recursos.
- Gran compatibilidad con librerías del ecosistema React.

Desventajas:

- Menor rendimiento que Flutter en animaciones complejas.
- Dependencia de módulos nativos de terceros para ciertas funciones.
- Mayor esfuerzo para lograr una interfaz uniforme entre plataformas.

Es especialmente útil para equipos con experiencia previa en React o proyectos que requieran rápida iteración con tecnologías web.

Ionic [20] es un framework basado en tecnologías web estándar (HTML, CSS y JavaScript) que permite desarrollar aplicaciones híbridas que se ejecutan dentro de un contenedor nativo. Puede usarse con Angular, React o Vue, lo que lo hacen muy versátil.

Ventajas:

- Facilidad de uso para desarrolladores web.
- Alta velocidad de desarrollo.
- Interfaz adaptable a múltiples plataformas.

Desventajas:

- Experiencia de usuario menos fluida comparada con frameworks nativos.
- Dependencia de WebView, lo que puede limitar el rendimiento.
- Integración más compleja con funciones del dispositivo.

Es una opción interesante cuando se prioriza el tiempo de desarrollo y el presupuesto, o se reutiliza código existente de aplicaciones web.

Angular [21] y Vue.js [22] son frameworks de JavaScript orientados al desarrollo de aplicaciones web progresivas (PWA). Angular, mantenido por Google, ofrece una arquitectura robusta y completa, mientras que Vue.js destaca por su sencillez y flexibilidad.

Ventajas (Angular):

- Estructura clara para proyectos grandes.
- Soporte oficial de Google.

Desventajas (Angular):

- Curva de aprendizaje pronunciada.
- Código más extenso para tareas simples.

Ventajas (Vue.js):

- Fácil de aprender e integrar en proyectos.
- Ligero y muy flexible.

Desventajas (Vue.js):

- Menor soporte corporativo en comparación con Angular o React.
- No siempre recomendado para proyectos de gran escala sin estructura clara.

Ambos son más adecuados cuando se desarrollan interfaces web complejas o aplicaciones PWA, aunque no son tan usados para apps móviles nativas como Flutter o React Native.

La elección del lenguaje y framework para el frontend dependerá de factores como el tipo de aplicación, la necesidad de compatibilidad multiplataforma, el rendimiento deseado y la experiencia del equipo de desarrollo.

2.2.3. Tecnologías para el Backend

El backend se encarga de la lógica de negocio, la gestión de bases de datos y la comunicación con el frontend. Su objetivo principal es procesar, almacenar y devolver información de manera eficiente. Las tecnologías para el backend varían en función del lenguaje de programación, el framework y la arquitectura utilizada.

Lenguajes de programación más utilizados en el backend:

- *Node.js (JavaScript)* [23]: entorno de ejecución basado en *JavaScript* que permite desarrollar aplicaciones del lado del servidor. Es ideal para aplicaciones en tiempo real y APIs debido a su enfoque asíncrono y no bloqueante.
- *Python* [24]: popular por su sintaxis simple y clara, así como por su amplio ecosistema de bibliotecas para aplicaciones de IA, análisis de datos y automatización.
- *Java* [25] y *C#* [26]: lenguajes robustos, muy utilizados en entornos empresariales y aplicaciones de gran escala. Java es conocido por su portabilidad, mientras que C# destaca en el ecosistema Microsoft.
- *PHP* [27]: lenguaje de programación ampliamente utilizado para el desarrollo de aplicaciones web, especialmente en sistemas de gestión de contenido y APIs básicas.

2.2.4. Frameworks más utilizados para el desarrollo backend

En el desarrollo de aplicaciones modernas, especialmente aquellas con arquitecturas cliente-servidor, la elección del framework backend adecuado resulta fundamental. Un framework backend proporciona herramientas, estructuras y abstracciones que facilitan la creación de APIs, la gestión de bases de datos, la autenticación de usuarios y otras tareas

esenciales del lado del servidor. A continuación, se analizan algunos de los frameworks backend más utilizados en la industria actual, junto con sus ventajas, desventajas y posibles aplicaciones.

Express [28] es un framework minimalista y flexible para Node.js, diseñado para desarrollar aplicaciones web y APIs de manera sencilla. Su arquitectura ligera permite una gran libertad en la estructura del proyecto, lo que lo convierte en una opción muy popular para el desarrollo de servicios RESTful.

Ventajas:

- Muy rápido gracias al motor de Node.js.
- Gran comunidad y amplia disponibilidad de middleware.
- Ideal para aplicaciones en tiempo real y microservicios.

Desventajas:

- Requiere definir muchas configuraciones desde cero, lo que puede ser complejo para proyectos grandes.
- Menor estructura por defecto comparado con otros frameworks más opinados.

Es especialmente útil si se busca velocidad y flexibilidad, pero requiere experiencia para organizar bien un proyecto grande.

Django [29] es un framework de alto nivel para Python que sigue el principio “batteries included”, lo que significa que viene con muchas herramientas integradas, como autenticación, administración, gestión de bases de datos (ORM), etc. Está orientado al desarrollo rápido y seguro de aplicaciones complejas.

Ventajas:

- Desarrollo rápido gracias a su enfoque de configuración por convención.
- Seguridad incorporada frente a ataques comunes como XSS o CSRF.
- Amplia documentación y comunidad activa.

Desventajas:

- Puede ser excesivo para proyectos pequeños.
- Menor flexibilidad si se quieren realizar personalizaciones profundas del stack.

Ideal si se desea un framework completo y productivo desde el inicio, especialmente útil en proyectos con enfoque en seguridad o IA (gracias a su integración con Python).

Spring Boot [30] es una extensión del ecosistema Spring que permite crear aplicaciones Java listas para producción con mínima configuración. Es ampliamente utilizado en el entorno empresarial y facilita la creación de microservicios escalables y robustos.

Ventajas:

- Escalable, ideal para arquitecturas empresariales.
- Gran cantidad de herramientas integradas y un ecosistema rico.
- Soporte sólido para microservicios y servicios REST.

Desventajas:

- Curva de aprendizaje pronunciada.
- Requiere más recursos del sistema que otros frameworks.

Recomendado si el proyecto debe integrarse con sistemas empresariales o necesita una arquitectura sólida y escalable a largo plazo.

Laravel [31] es un framework para PHP que destaca por su sintaxis elegante, su enfoque en la productividad del desarrollador y una amplia gama de herramientas incorporadas como Eloquent ORM, plantillas Blade y sistema de rutas intuitivo.

Ventajas:

- Fácil de aprender y usar.
- Gran comunidad y ecosistema rico en paquetes.
- Buen rendimiento en desarrollo de APIs y sitios web.

Desventajas:

- PHP tiene una reputación desigual en algunos entornos técnicos.
- Menos eficiente para aplicaciones de gran escala en tiempo real.

Opción muy conveniente para aplicaciones web tradicionales o APIs con enfoque rápido en MVPs.

La decisión sobre qué tecnología emplear debe basarse en factores como el lenguaje de programación preferido, el rendimiento requerido, la escalabilidad, la facilidad de desarrollo y la comunidad de soporte.

2.2.5. Tecnologías para APIs

Las APIs [32] (Application Programming Interfaces) son fundamentales para la comunicación entre el frontend y el backend, permitiendo que diferentes sistemas y servicios interactúen de forma eficiente. Existen distintos estilos de arquitectura y herramientas que facilitan su diseño, desarrollo e integración.

Estilos de arquitectura de APIs:

- *REST* [33]: el estilo más utilizado debido a su simplicidad y flexibilidad. Utiliza métodos *HTTP* estándar (GET, POST, PUT, DELETE) para la gestión de recursos y es fácilmente integrable con aplicaciones móviles y web.
- *GraphQL* [34]: lenguaje de consulta desarrollado por Facebook que permite obtener solo los datos necesarios en una única solicitud. Es especialmente útil para aplicaciones con estructuras de datos complejas.
- *gRPC* [35]: framework de comunicación de alto rendimiento que utiliza el protocolo *HTTP/2* para la transferencia de datos. Es ideal para aplicaciones de microservicios debido a su eficiencia y velocidad.

2.2.6. Herramientas y estándares para el desarrollo y prueba de APIs

El diseño, desarrollo, prueba y documentación de APIs es un proceso esencial en el desarrollo de software moderno. Contar con herramientas y estándares bien definidos permite mejorar la calidad, mantenibilidad y escalabilidad del software, además de facilitar la colaboración entre equipos. A continuación, se describen algunas de las herramientas y estándares más utilizados, incluyendo sus ventajas y desventajas, así como su aplicabilidad según el tipo de proyecto.

Postman [36] es una de las herramientas más populares para el diseño, prueba y documentación de APIs. Permite a los desarrolladores enviar solicitudes *HTTP* a servidores, verificar las respuestas, automatizar pruebas y generar documentación de forma intuitiva.

Ventajas:

- Interfaz gráfica fácil de usar.
- Permite crear colecciones reutilizables de peticiones.
- Soporta pruebas automatizadas y scripts de validación con JavaScript.
- Integra herramientas de colaboración para equipos de desarrollo.

Desventajas:

- Algunas funcionalidades avanzadas requieren una suscripción de pago.
- Puede ser pesado en equipos con pocos recursos.

Aplicaciones: ideal para entornos de desarrollo colaborativo y para la verificación rápida de endpoints en APIs REST o GraphQL.

OpenAPI [37], anteriormente conocido como Swagger, es un estándar ampliamente adoptado para describir APIs RESTful. Define un formato estructurado (generalmente en YAML o JSON) que permite describir todos los aspectos de una API, incluyendo rutas, parámetros, respuestas y seguridad.

Ventajas:

- Facilita la generación automática de documentación interactiva.
- Permite la validación automática de solicitudes y respuestas.
- Es compatible con múltiples lenguajes y frameworks.
- Admite la generación automática de código cliente y servidor.

Desventajas:

- Puede tener una curva de aprendizaje si se utiliza de forma manual.
- Está orientado principalmente a APIs REST, no tanto a gRPC o GraphQL.

Aplicaciones: muy útil en entornos profesionales donde se necesita mantener una documentación técnica precisa y actualizada de la API, y donde se busca facilitar la integración con terceros.

JSON [38] y XML [39] son los formatos más utilizados para el intercambio de datos entre aplicaciones y servicios mediante APIs.

Ventajas (JSON):

- Ligero, fácil de leer y escribir.
- Muy eficiente para aplicaciones web y móviles.
- Soporte nativo en la mayoría de lenguajes de programación modernos.

Desventajas (JSON):

- No permite validación de esquema tan rigurosa como XML (aunque puede complementarse con JSON Schema).
- Menor soporte para metainformación.

Ventajas (XML):

- Altamente estructurado y extensible.
- Admite validación mediante DTD o XSD.
- Mejor compatibilidad con sistemas heredados (legacy).

Desventajas (XML):

- Verboso y menos eficiente en términos de tamaño y procesamiento.
- Más complejo de manipular comparado con JSON.

Aplicaciones: JSON es recomendado para nuevas aplicaciones móviles o web por su ligereza y compatibilidad, mientras que XML se utiliza comúnmente en entornos empresariales, financieros o de integración con sistemas antiguos.

La elección de la arquitectura y las herramientas para el desarrollo de APIs dependerá de las necesidades del proyecto, como el rendimiento, la complejidad de los datos y la escalabilidad de la aplicación.

3. Diseño y Programación de FamSync

La gestión del entorno familiar presenta desafíos significativos derivado de la dispersión de tareas y responsabilidades entre diferentes plataformas. En la práctica, las familias suelen recurrir a varias aplicaciones para coordinar eventos, asignar tareas, preparar listas de la compra o programar recordatorios, lo que provoca una fragmentación de la información y obliga a los usuarios a cambiar constantemente de herramienta. Esta falta de centralización incrementa la complejidad de la planificación diaria y eleva la probabilidad de omisiones o solapamientos de actividades.

Del mismo modo, la comunicación eficiente entre los miembros de la familia se ve mermada por la ausencia de un calendario compartido y sincronizado; la descoordinación de horarios genera confusiones sobre quién debe encargarse de cada tarea o asistir a cada compromiso. Por otra parte, resulta especialmente preocupante la gestión de la medicación, ya que el seguimiento de dosis y horarios de toma suele realizarse de forma manual o mediante alarmas independientes, sin un registro unificado que permita verificar el cumplimiento y el historial de tratamiento.

Ante estas carencias, FamSync se plantea como una solución integral que reúne en una sola plataforma todas las funcionalidades necesarias para organizar eventos, distribuir y monitorear tareas, gestionar productos del hogar e incluso controlar la toma de medicación. A continuación se detalla el diseño de su arquitectura y la implementación de cada uno de los módulos clave.

3.1 Arquitectura

Para el desarrollo de FamSync, se ha adoptado una arquitectura de tipo cliente-servidor con separación de responsabilidades entre las distintas capas del sistema: la interfaz de usuario (frontend), la lógica del negocio (backend) y la persistencia de datos (base de datos). Esta arquitectura modular permite escalar la aplicación, facilitar su mantenimiento y garantizar una experiencia de usuario fluida y consistente, como se muestra en la Figura 5, que presenta el esquema detallado de la arquitectura del sistema.

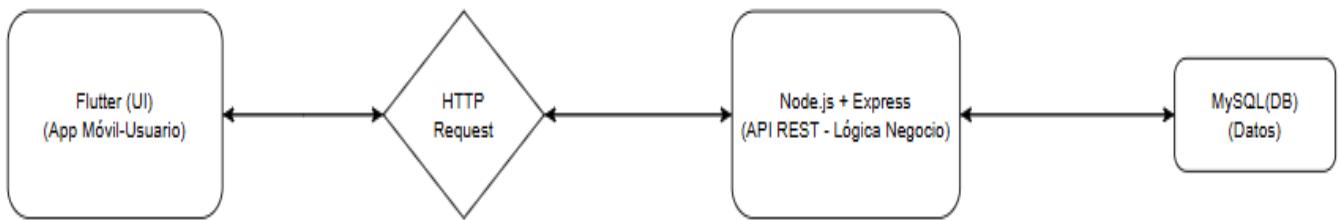


Figura 5. Arquitectura cliente-servidor.

3.1.1. Esquema de comunicación

Como se aprecia en el esquema representado en la Figura 1, la arquitectura de comunicación de FamSync sigue un modelo cliente-servidor, en el cual los distintos componentes del sistema interactúan entre sí de forma estructurada. La aplicación móvil, desarrollada con Flutter, actúa como cliente y se encarga de realizar peticiones HTTP hacia una API REST que constituye el núcleo del backend. Esta API, implementada con Node.js y el framework Express, recibe las solicitudes procedentes del cliente, procesa la lógica de negocio correspondiente y gestiona el acceso a los datos almacenados.

Una vez procesada la petición, el backend se comunica con la base de datos MySQL, la cual se encarga de almacenar de forma persistente toda la información relevante de la aplicación, incluyendo usuarios, eventos, tareas, productos o listas de la compra. Tras acceder a los datos requeridos o realizar las operaciones necesarias, la base de datos responde al backend, que a

su vez construye una respuesta adecuada y la envía de vuelta al frontend, donde se muestra al usuario en una interfaz clara y estructurada.

3.1.2 FrontEnd (Cliente - Aplicación Móvil)

El desarrollo del frontend de FamSync se ha llevado a cabo utilizando Flutter, un framework de código abierto creado por Google que permite la creación de aplicaciones móviles multiplataforma desde una única base de código. Esta tecnología utiliza el lenguaje de programación Dart, diseñado específicamente para interfaces gráficas reactivas, ofreciendo un rendimiento muy cercano al de las aplicaciones nativas.

Flutter ha sido elegido por varias razones clave. En primer lugar, su capacidad multiplataforma permite desarrollar simultáneamente para Android e iOS, lo cual resulta fundamental para maximizar el alcance de la aplicación sin duplicar esfuerzos de desarrollo. Además, Flutter proporciona un alto rendimiento, gracias a su uso del motor gráfico Skia, que renderiza la interfaz de usuario de forma eficiente y fluida.

Otro aspecto destacable es su capacidad de personalización visual, ya que incorpora un sistema de widgets reutilizables que permiten construir interfaces atractivas y adaptadas a cada necesidad. Por último, cabe mencionar la funcionalidad de Hot Reload, que facilita un desarrollo más ágil y permite probar cambios en tiempo real sin necesidad de reiniciar la aplicación, lo que mejora significativamente el proceso de prueba y depuración.

3.1.3 Backend (Servidor - API REST)

El backend de FamSync está desarrollado utilizando Node.js, un entorno de ejecución basado en JavaScript que se ejecuta en el servidor, junto con Express.js, un framework minimalista que facilita la creación de APIs RESTful de forma estructurada, escalable y eficiente.

La elección de esta tecnología se basa en varias ventajas clave. En primer lugar, Node.js ofrece un alto rendimiento gracias a su modelo de ejecución asíncrono y basado en eventos, lo cual lo convierte en una solución especialmente eficaz para aplicaciones que requieren manejar múltiples solicitudes concurrentes, como es el caso de FamSync. Esta capacidad resulta esencial en escenarios donde varios miembros de la familia pueden estar accediendo y actualizando datos simultáneamente.

Además, Node.js utiliza JavaScript tanto en el cliente como en el servidor, lo que simplifica la comunicación entre ambas capas, especialmente al trabajar con datos en formato JSON, nativo en ambos entornos. Por su parte, Express.js aporta un amplio ecosistema de middleware, que

permite gestionar tareas comunes como la autenticación, la validación de datos, la seguridad de las rutas y el manejo de errores, entre otras.

Gracias a esta estructura, el backend de FamSync es capaz de ofrecer una API robusta, flexible y bien organizada, que actúa como intermediaria entre la interfaz de usuario y la base de datos, garantizando que toda la lógica del negocio se gestione correctamente.

3.1.4 Base de datos (Persistencia de datos)

Para la capa de persistencia de datos, FamSync utiliza MySQL, un sistema de gestión de bases de datos relacional ampliamente extendido y reconocido por su fiabilidad, eficiencia y soporte de integridad referencial.

La estructura de datos se ha diseñado para reflejar de forma coherente las relaciones entre los distintos elementos que conforman la aplicación, como usuarios, perfiles, eventos, tareas, productos y listas. MySQL permite definir claves primarias y foráneas que aseguran la consistencia de los datos, y su modelo relacional facilita la implementación de consultas complejas y relaciones jerárquicas, algo esencial para representar la organización familiar en la aplicación.

Otra ventaja de MySQL es su escalabilidad, que permite gestionar grandes volúmenes de información sin comprometer el rendimiento del sistema. Para mejorar aún más la eficiencia, se han definido índices en las tablas más consultadas, optimizando el acceso a los datos y reduciendo los tiempos de respuesta.

En conjunto, MySQL proporciona una solución sólida para el almacenamiento de datos en FamSync, permitiendo un control preciso sobre la estructura de la información y una integración fluida con el backend.

3.2 Gestión de Perfiles

El módulo de gestión de perfiles es uno de los pilares fundamentales de FamSync, ya que permite representar a cada miembro de la familia dentro de la aplicación y vincular sus eventos, tareas, productos y configuraciones personales. Su correcta implementación garantiza que la información esté organizada, personalizada y claramente asociada a cada usuario.

En primer lugar, el sistema distingue entre usuarios y perfiles familiares. Cada usuario accede a la aplicación mediante un sistema de autenticación que requiere correo electrónico y contraseña (véase Figura 6).

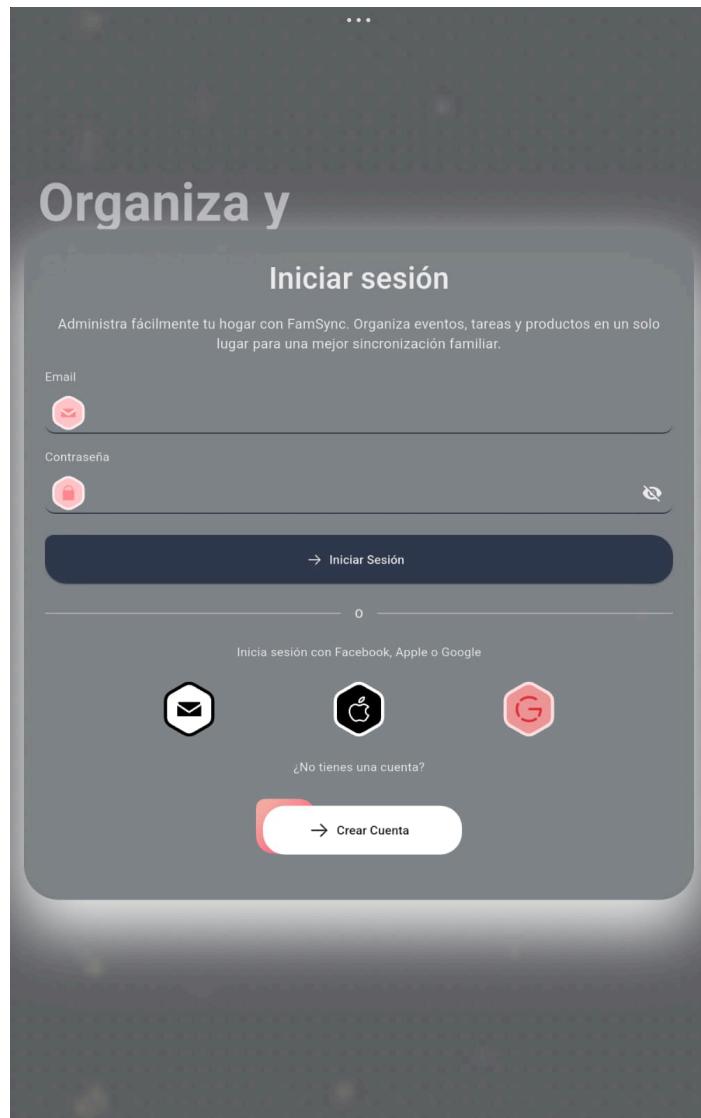


Figura 6. Formulario Login

Una vez dentro, el usuario puede crear o unirse a una familia existente, generando así un entorno compartido donde se gestionan los datos comunes. Cada familia puede estar compuesta por varios perfiles, representando a cada uno de sus integrantes, incluyendo tanto adultos como niños o personas dependientes.

Para comprender mejor cómo se representa internamente esta entidad en la aplicación, a continuación en la *Tabla 1* se detallan los principales atributos que conforman la estructura de un usuario en FamSync.

Campo	Tipo de dato	Descripción
Id	int	Identificador único del usuario.
Correo	String	Dirección de correo electrónico utilizada para el inicio de sesión.
Teléfono	int	Número de teléfono del usuario. Puede utilizarse para recuperación o contacto.
Nombre	String	Nombre del usuario titular de la cuenta.
Password	String	Contraseña cifrada que protege el acceso del usuario a la aplicación.

Tabla 1. Estructura de usuario

Los perfiles permiten definir información básica como el nombre, la edad, una imagen identificativa para facilitar la visualización y, en versiones futuras, podrían incluir el rol dentro de la familia (por ejemplo: madre, hijo, abuelo). Esta distinción entre perfil y usuario facilita que un solo usuario pueda gestionar varios perfiles desde su cuenta, algo útil en contextos donde un adulto se encarga de varios miembros del hogar.

Para representar esta entidad dentro del sistema, FamSync define una estructura específica que recoge los campos necesarios para gestionar la identidad y configuración de cada perfil.

A continuación en la Tabla 2, se detallan los atributos principales de un perfil.

Campo	Tipo de dato	Descripción
Id	int	Identificador único del perfil.
Usuarioid	int	Identificador del usuario al que está vinculado este perfil.
Nombre	int	Nombre del miembro familiar representado por el perfil.
FotoPerfil	String	Ruta o URL de la imagen asignada al perfil.
Pin	String	Código PIN asociado al perfil para su protección y acceso restringido.
FechaNacimiento	String	Fecha de nacimiento del perfil, usada para calcular la edad y adaptar funciones.

Tabla 2. Estructura de perfil

Cada perfil actúa como un punto de enlace con el resto de módulos de la aplicación. Los eventos del calendario, las tareas asignadas, los productos creados y las listas de productos están asociados directamente a uno o varios perfiles, permitiendo una gestión clara y precisa de la información. Además, el sistema permite modificar, eliminar o reasignar perfiles dentro de la familia, manteniendo siempre la integridad de los datos relacionados.

La gestión de perfiles se ha diseñado con un enfoque en la simplicidad y la usabilidad, de forma que cualquier usuario pueda añadir, editar o consultar los perfiles sin dificultad. Asimismo, se ha contemplado la posibilidad de que algunos perfiles no estén vinculados a un usuario con cuenta propia, permitiendo que, por ejemplo, un adulto gestione las actividades de un niño sin que este tenga que iniciar sesión por separado.



Figura 7. Vista de selección de perfil.

La figura 7 muestra la vista de selección de perfil, una de las pantallas iniciales tras el inicio de sesión del usuario en FamSync. Esta interfaz permite seleccionar el perfil familiar con el que se desea interactuar dentro de la aplicación. Cada perfil está representado visualmente mediante un avatar personalizado, junto con el nombre del miembro al que pertenece. Esta disposición facilita una identificación rápida y accesible para todos los usuarios, incluidos niños o personas con menor experiencia digital.

Al seleccionar un perfil, se despliega una ventana emergente de verificación por PIN que solicita al usuario introducir el código de acceso vinculado al perfil. Esta funcionalidad permite proteger la privacidad y personalización de cada perfil dentro del entorno familiar compartido. La ventana emergente incluye un campo de entrada de texto para el PIN, junto con dos botones: uno para cancelar la acción y otro para iniciar sesión en el perfil seleccionado.

Además, se muestra una opción para añadir un nuevo perfil, indicada con un icono de “+”, lo que permite a los usuarios ampliar la familia registrada de forma sencilla y visualmente intuitiva.

3.2.1 Implementación técnica

Con el objetivo de mejorar la experiencia del usuario y facilitar el acceso rápido a su entorno dentro de la aplicación FamSync, se implementa un sistema de persistencia del perfil activo utilizando la funcionalidad de almacenamiento local del dispositivo. Este sistema se encarga de guardar ciertos datos clave de forma permanente para que estén disponibles incluso después de cerrar la aplicación.

En concreto, cuando un usuario selecciona su perfil e introduce correctamente su PIN de acceso, el identificador único de ese perfil se guarda en las preferencias compartidas del dispositivo. Este almacenamiento se realiza de forma local y segura, y se utiliza únicamente para reconocer qué perfil estaba activo en la última sesión.

Gracias a esta técnica, la siguiente vez que se inicie la aplicación, el sistema puede consultar automáticamente si existe un perfil previamente autenticado. Si lo encuentra, y no se ha cerrado sesión manualmente, se evita el paso de selección de perfil y la introducción del PIN, accediendo directamente al contenido personalizado de ese perfil. Esto resulta especialmente útil en contextos familiares donde los usuarios suelen ser los mismos y se busca una experiencia de uso más ágil y fluída.

Este enfoque no sólo mejora la usabilidad, sino que también reduce el número de pasos necesarios para comenzar a interactuar con la aplicación, manteniendo a su vez un nivel adecuado de seguridad, ya que sólo se guarda un identificador no sensible y no se almacenan directamente credenciales como el PIN.

3.3 Gestión de Inventario

El módulo de inventario constituye una de las funcionalidades clave de FamSync, permitiendo a los usuarios llevar un control detallado de los productos disponibles en el hogar. Este sistema está diseñado para facilitar dos aspectos fundamentales de la organización doméstica:

- Gestión del inventario del hogar: permite registrar, visualizar y actualizar los productos disponibles, lo que ayuda a llevar un control preciso de los artículos que se tienen en casa.
- Planificación de listas de la compra: ofrece herramientas para crear listas personalizadas, añadir productos desde el inventario y compartirlas entre varios perfiles familiares, optimizando así el proceso de compra.

Ambas funcionalidades fomentan una organización eficiente y colaborativa, reduciendo olvidos, compras duplicadas y mejorando la comunicación entre los miembros del hogar.

Cada producto registrado en la aplicación se representa mediante una instancia estructurada que recoge toda la información relevante sobre el mismo. Esta instancia actúa como una unidad lógica dentro del sistema y contiene los siguientes atributos.

Campo	Tipo de dato	Descripción
Id	int	Identificador único del producto dentro de la base de datos. Permite distinguirlo del resto de registros.
Nombre	String	Nombre del producto, utilizado para su identificación visual y en las búsquedas.
Imágenes	List<String>	Lista de URLs o rutas de imágenes asociadas al producto. Permite representar visualmente el artículo.
Tienda	String	Nombre de la tienda donde habitualmente se adquiere el producto. Facilita su agrupación por establecimiento.
Precio	double	Precio estimado del producto, expresado en euros. Se utiliza para cálculos dentro de las listas de compra.
PerfilCreador	int	Identificador del perfil familiar que añadió el producto. Permite saber a qué miembro está vinculado.

UsuarioCreador	int	Identificador del usuario con cuenta que creó el producto. Puede diferir del perfil si es un gestor.
Visible	List<int>	Lista de IDs de perfiles que tienen acceso para ver e interactuar con el producto. Controla su visibilidad.

Tabla 3. Estructura de producto

Esta estructura permite una representación completa y flexible del producto dentro del sistema, contemplando tanto su descripción material como su contexto de uso en el entorno familiar. Gracias a este modelo, FamSync es capaz de ofrecer un control granular sobre la información visible para cada perfil, garantizando una experiencia personalizada y respetuosa con los distintos roles que pueden existir en un hogar (adultos, niños, personas dependientes, etc.).

3.3.1 Asignación de productos a listas de la compra

Dentro del módulo de inventario, FamSync incorpora un sistema de listas de la compra que permite a los perfiles planificar y organizar la adquisición de productos de forma estructurada y colaborativa. Estas listas funcionan como contenedores dinámicos donde los usuarios pueden agrupar productos previamente registrados en el inventario, facilitando así la gestión de compras tanto individuales como familiares.

Cada lista representa una instancia independiente, identificada de forma única y vinculada a uno o varios perfiles. Esta flexibilidad permite crear listas adaptadas a distintas situaciones del día a día: por ejemplo, una lista para la compra semanal, otra para una tienda específica, o incluso listas diseñadas para eventos concretos como celebraciones o vacaciones.

Para permitir una gestión precisa y eficiente, cada lista de la compra se define mediante una estructura interna que recoge los atributos más relevantes. A continuación, se muestra una tabla con los principales campos que componen una instancia de lista en FamSync:

Campo	Tipo de dato	Descripción
Id	int	Identificador único de la lista dentro del sistema. Permite diferenciarla del resto de listas creadas.
Nombre	String	Título de la lista que describe su propósito (por ejemplo, "Compra semanal", "Limpieza", etc.).
IdPerfil	int	Identificador del perfil familiar que creó la lista. Se usa para asociar la lista a su creador.
IdUsuario	int	Identificador del usuario con cuenta que generó la lista. Puede diferir del perfil si gestiona varios.
Visible	List<int>	Lista de identificadores de perfiles que tienen acceso a consultar o modificar la lista.
Productos	List<int>	Conjunto de identificadores de productos incluidos en la lista. Representa el contenido de la misma.

Tabla 4. Estructura de lista

Una lista contiene un conjunto de productos, referenciados mediante su identificador único. Como se muestra en la Tabla 4, estos productos pueden visualizarse dentro de la lista con sus detalles asociados, tales como nombre, imagen o tienda habitual. La interfaz permite añadir o eliminar productos de manera rápida e intuitiva, así como marcar productos como "comprados", lo que facilita el seguimiento durante el proceso de adquisición, como se aprecia en la Figura 8.

Un aspecto especialmente destacable de este sistema es su enfoque colaborativo. Cada lista puede estar asociada a varios perfiles, lo que permite que distintos miembros de la familia puedan consultarla y editarla.

Gracias a estas funcionalidades, FamSync no solo permite llevar un control detallado del inventario doméstico, sino también planificar de forma eficiente las compras familiares, fomentando la colaboración entre perfiles y evitando duplicidades, olvidos o tareas repetidas en la gestión diaria del hogar.

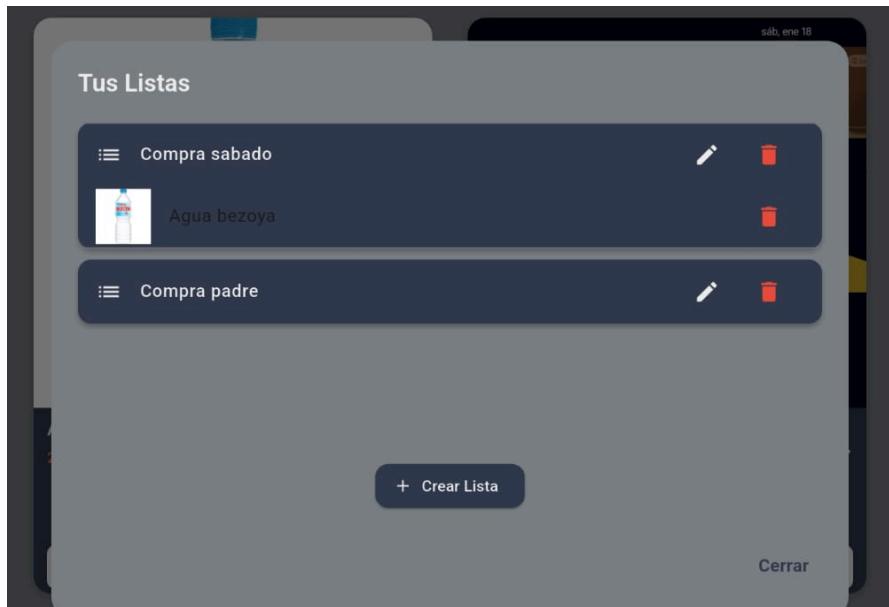


Figura 8. Vista de listas de productos.

3.3.2 Visualización en formato Almacén

Para facilitar el acceso a la información y la gestión de productos dentro del hogar, el módulo de inventario de FamSync ofrece diferentes vistas adaptadas a las necesidades del usuario. Estas vistas permiten consultar el inventario de forma clara, organizada y dinámica, mejorando la experiencia de uso y reduciendo el tiempo necesario para localizar un producto concreto.

La interfaz principal del módulo corresponde a la vista general, en la que se muestran todos los productos registrados en el inventario familiar. Tal como se observa en la Figura 9, esta vista presenta una disposición en forma de tarjetas o bloques que permiten identificar rápidamente los productos, mostrando información básica como el nombre, la imagen asociada, la tienda habitual y el precio estimado.

Para facilitar la navegación, los productos pueden visualizarse según distintos criterios de organización, entre los que se incluyen:

- Recientes: muestra los productos añadidos más recientemente, facilitando el acceso rápido a los elementos recién introducidos.
- Por tienda: agrupa los productos según el establecimiento asignado, permitiendo planificar compras por ubicación.
- Todos los productos: lista completa del inventario, sin filtros aplicados.

Cada elemento de esta vista general actúa como punto de entrada a la vista específica, que se activa al seleccionar un producto. Esta vista individual, mostrada en la Figura 10, presenta en detalle toda la información registrada sobre el producto seleccionado, incluyendo nombre completo, imágenes, tienda, precio, creador, perfiles con visibilidad y cualquier otra información relevante. Desde esta pantalla, el usuario también puede editar o eliminar el producto, así como añadirlo a una lista de la compra.

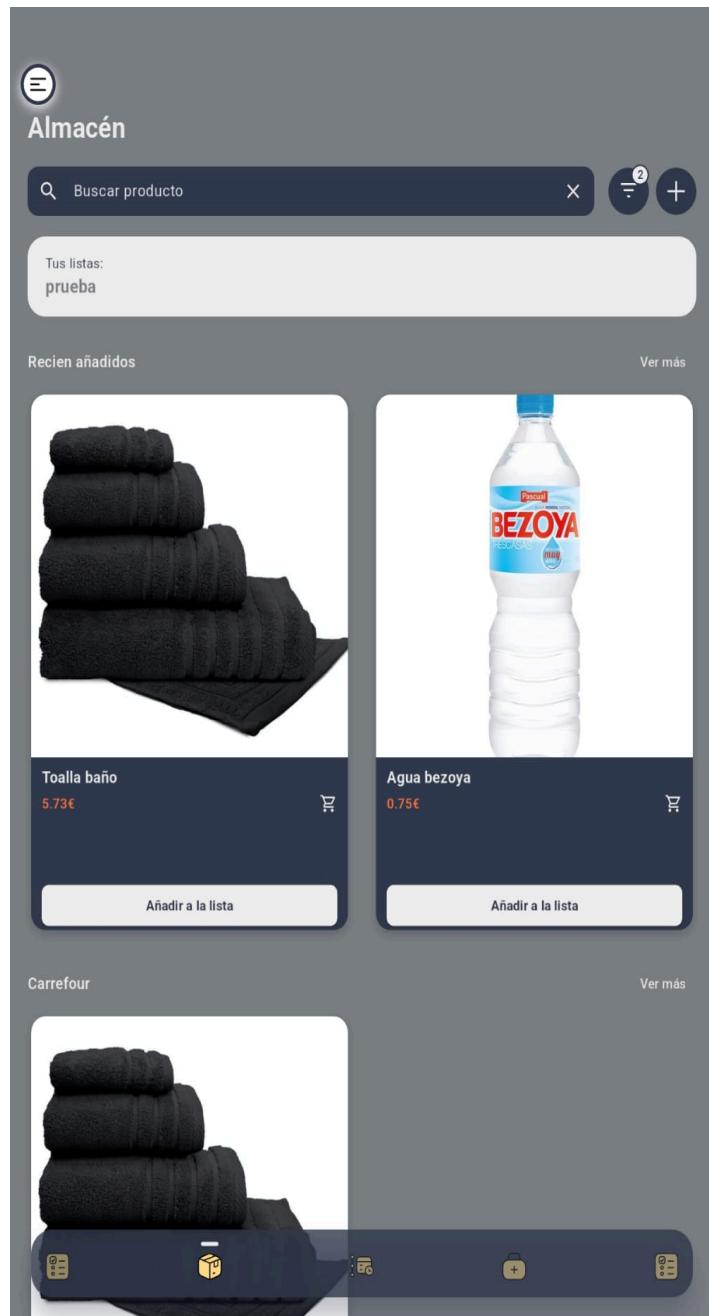


Figura 9. Vista General

Estas vistas están diseñadas para adaptarse de forma fluida al número de productos existentes, con mecanismos de carga dinámica que optimizan el rendimiento, especialmente en dispositivos móviles. Además, se han aplicado principios de diseño limpio y responsivo, asegurando una correcta visualización tanto en pantallas pequeñas como en dispositivos con mayores dimensiones.

Gracias a este sistema de visualización dual, FamSync permite consultar y gestionar el inventario del hogar de forma intuitiva, facilitando tanto la visión global del estado del almacén como la consulta detallada de cada artículo.

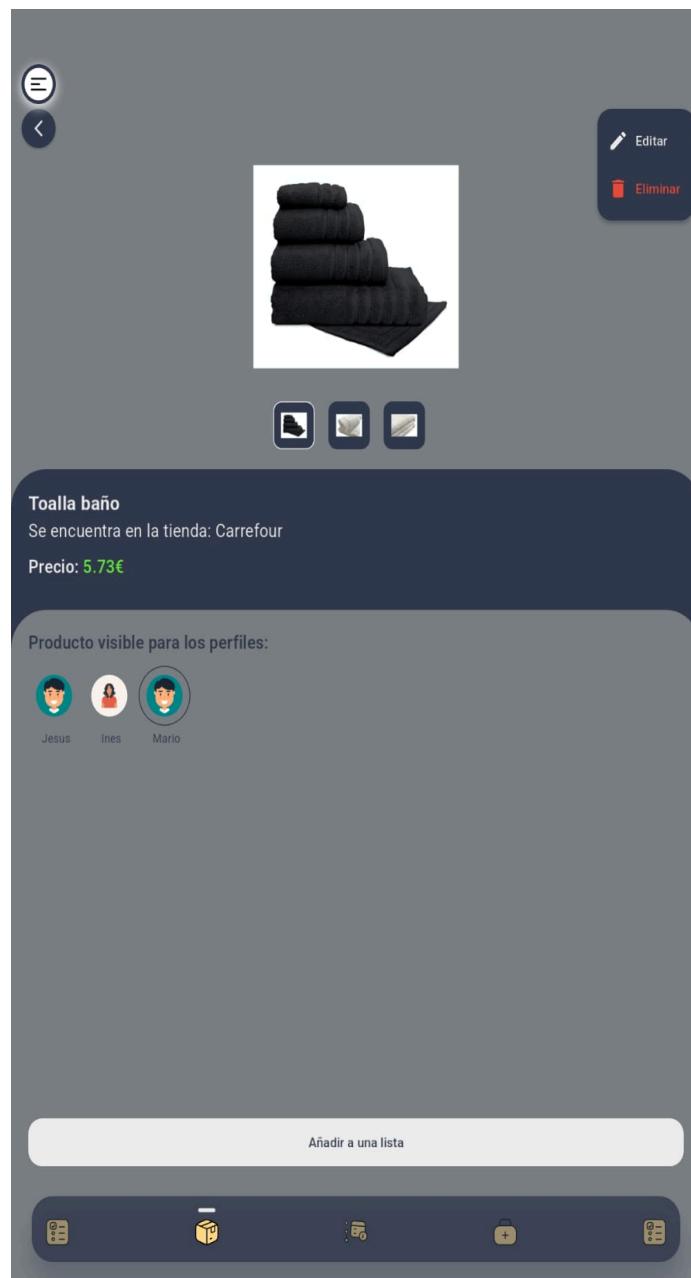


Figura 10. Vista Individual

3.3.3 Implementación técnica

La implementación del módulo de inventario en FamSync se ha centrado en garantizar una experiencia fluida, reactiva y eficiente tanto en términos de rendimiento como de interacción del usuario. Para ello, se han aplicado diversas técnicas de desarrollo centradas en la gestión del estado, la optimización de la carga de datos y la navegación entre vistas.

En primer lugar, se ha integrado un gestor de estado que permite actualizar la interfaz de usuario (UI) en tiempo real. Esta solución asegura que cualquier modificación en el inventario, como la creación, edición o eliminación de un producto, se refleje de forma inmediata en pantalla sin necesidad de recargar manualmente la vista. Esta gestión reactiva de los datos es esencial para mantener la coherencia entre el estado de la aplicación y la información presentada al usuario, y mejora significativamente la usabilidad de la aplicación.

Por otro lado, se han empleado estructuras de datos y técnicas de carga optimizadas para garantizar un buen rendimiento, especialmente en dispositivos móviles con recursos limitados. En lugar de cargar todo el inventario completo en memoria, el sistema ha sido diseñado para renderizar únicamente los productos que se encuentran visibles en pantalla en cada momento. Esta técnica, conocida como carga perezosa o “lazy loading”, reduce el consumo de recursos y mejora los tiempos de respuesta, permitiendo escalar el uso del módulo incluso con inventarios extensos.

Finalmente, la navegación entre vistas —por ejemplo, al pasar de la vista general del almacén a la vista específica de un producto— se ha desarrollado utilizando transiciones animadas y suaves, que refuerzan la continuidad visual y mejoran la experiencia del usuario. Estas transiciones ayudan a mantener el contexto del usuario, haciendo que la interacción con la aplicación resulte más natural e intuitiva.

En conjunto, estos aspectos técnicos permiten que el módulo de inventario no solo sea funcional, sino también ágil, moderno y agradable de utilizar, alineándose con los estándares actuales del desarrollo móvil centrado en el usuario.

3.4 Gestión de Tareas

El módulo de tareas de FamSync está orientado a facilitar el reparto y el seguimiento de las responsabilidades diarias dentro del entorno familiar. Su principal objetivo es permitir que todos los perfiles registrados puedan visualizar, gestionar y colaborar en tareas compartidas, fomentando la participación equitativa y reduciendo los olvidos o sobrecargas de trabajo.

Cada tarea se representa en el sistema como una instancia estructurada que contiene información clave para su gestión. A continuación en la *Tabla 5*, se muestran los atributos más importantes de una tarea:

Campo	Tipo de dato	Descripción
Id	int	Identificador único de la tarea dentro de la base de datos.
Creador	int	ID del perfil que ha creado la tarea. Permite mantener trazabilidad.
Destinatario	List<int>	Lista de perfiles a los que se ha asignado la tarea.
Nombre	String	Título breve que describe la acción principal a realizar.
Descripción	String	Texto libre con información adicional o instrucciones detalladas.
Categoría	int? (nulleable)	Categoría opcional para clasificar la tarea temáticamente.
Evento	int? (nulleable)	Identificador del evento asociado en caso de estar vinculada al calendario.
Prioridad	int	Nivel de prioridad de la tarea (por ejemplo, 1: baja, 2: media, 3: alta).
Progreso	int	Porcentaje de completado de la tarea.

Tabla 5. Estructura de tarea

Además del almacenamiento y la representación estructurada, el módulo de tareas de FamSync ofrece una serie de funcionalidades clave:

- Creación rápida de tareas individuales o compartidas, con posibilidad de asignarlas a uno o varios perfiles familiares.

- Clasificación temática mediante categorías personalizadas, que permiten organizar las tareas por ámbitos como “Casa”, “Colegio” o “Salud”.
- Priorización de tareas según nivel de urgencia (baja, media o alta), facilitando la identificación de lo más importante en cada momento.
- Seguimiento del estado de cada tarea mediante un sistema de progreso (pendiente, en proceso o completada), visualizado en forma de porcentaje.
- Visualización personalizada por perfil, mostrando a cada usuario únicamente las tareas que le corresponden o en las que participa.
- Vistas filtradas y buscador integrado, que permiten localizar fácilmente tareas por nombre, estado, categoría o prioridad.
- Edición, eliminación y reasignación dinámica de tareas, lo que permite adaptarse a cambios imprevistos o redistribuciones dentro del hogar.
- Asociación opcional de tareas con eventos del calendario, lo que permite programarlas con fecha y hora específica si es necesario.

Gracias a este conjunto de funcionalidades, el módulo de tareas no solo actúa como una lista compartida de pendientes, sino como un sistema dinámico de organización colaborativa. Permite a cada miembro de la familia mantenerse informado, asumir responsabilidades y colaborar en el mantenimiento del hogar de forma activa, estructurada y visualmente accesible.

3.4.1 Asignación de tareas a perfiles

Una de las funcionalidades clave del módulo de tareas de FamSync es la posibilidad de asignar tareas a uno o varios perfiles familiares. Esta característica permite distribuir de forma clara y estructurada las responsabilidades del hogar, evitando confusiones y garantizando que cada miembro sepa con exactitud qué tareas le han sido encomendadas.

La asignación se realiza en el momento de crear una nueva tarea, mediante una interfaz intuitiva que permite seleccionar los perfiles deseados entre los que forman parte del núcleo familiar (como se muestra en la Figura 11). Asimismo, es posible modificar esta asignación posteriormente desde la pantalla de edición de tareas, accediendo a un formulario específico que permite reasignar la tarea a otros perfiles si es necesario, como se aprecia en la Figura 12. Esta flexibilidad resulta especialmente útil en contextos familiares cambiantes, donde las responsabilidades pueden redistribuirse con frecuencia.

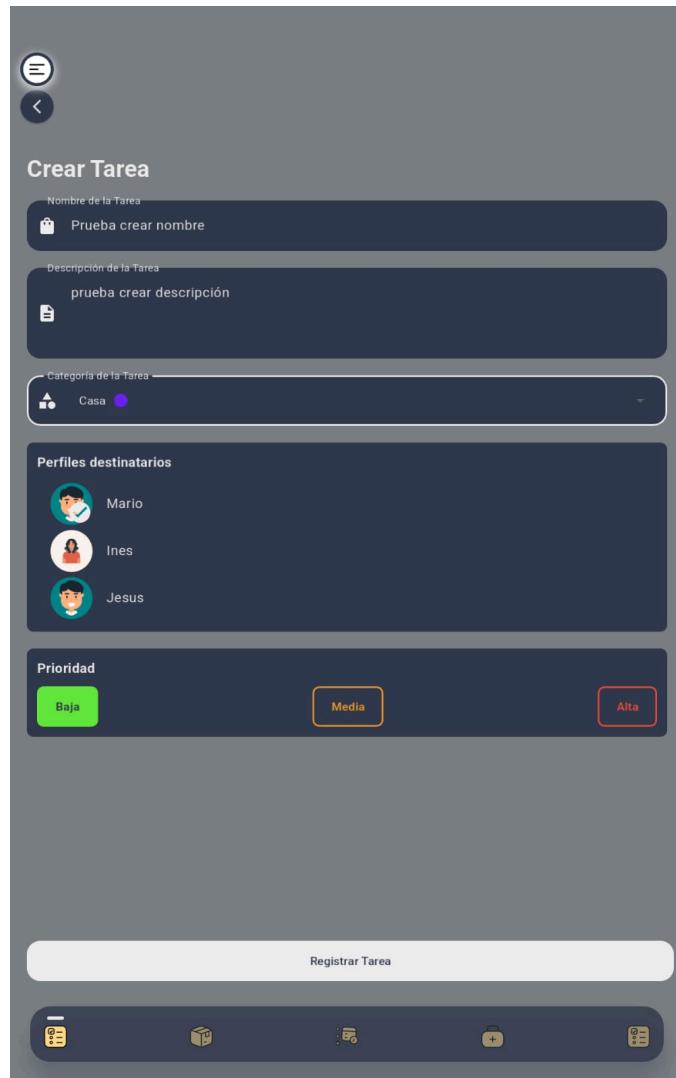


Figura 11. Creación de tarea

Desde el punto de vista funcional, esta asignación está directamente vinculada a cada perfil, lo que implica que todos los miembros de la familia pueden acceder desde su espacio personal a un listado con las tareas que tienen asociadas. Este listado está organizado por estados —pendiente, en progreso o completado— y se actualiza en tiempo real, lo que facilita el seguimiento del cumplimiento de las responsabilidades asignadas sin necesidad de navegar por tareas ajenas o irrelevantes para cada usuario.

Este enfoque personalizado contribuye a mejorar la organización interna del hogar, pero también promueve la corresponsabilidad, ya que permite que todos los miembros —incluidos los más jóvenes o dependientes— puedan implicarse activamente en las tareas domésticas. Además, al tratarse de un sistema compartido y visual, se favorece la transparencia, la planificación conjunta y la reducción de la carga mental que muchas veces recae de forma desigual sobre un único perfil familiar.

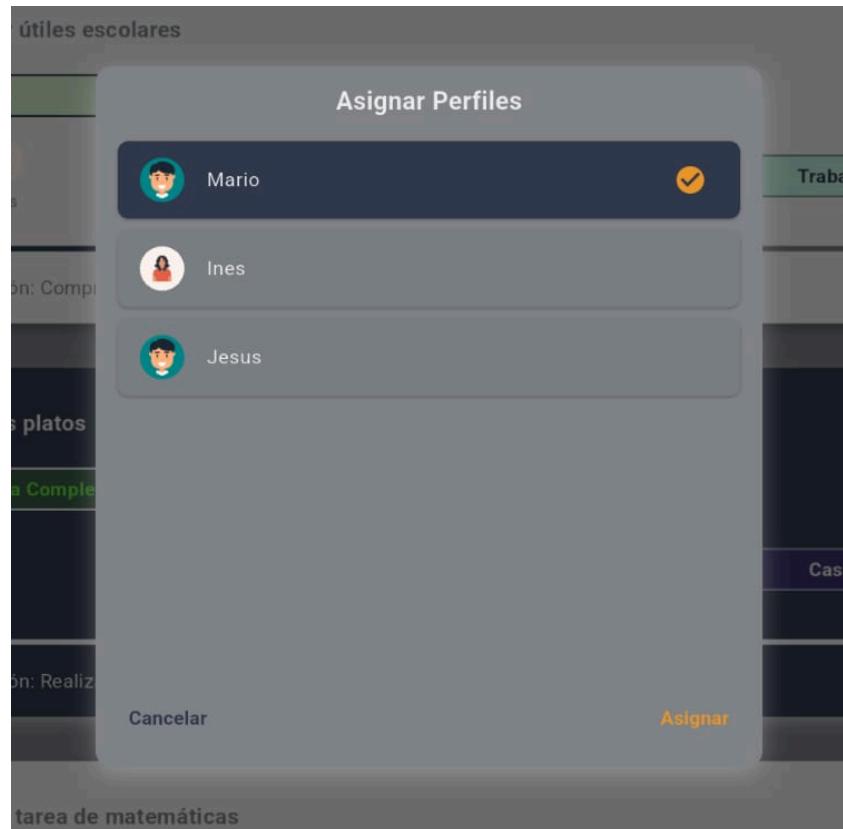


Figura 12. Asignación de perfiles

3.4.2 Visualización en formato Agenda

La vista principal del módulo de tareas, representada en la Figura 13, actúa como un panel de control desde el cual el usuario puede visualizar, buscar y filtrar fácilmente las tareas registradas en la aplicación. Esta pantalla ha sido diseñada para ofrecer una experiencia de usuario organizada, intuitiva y adaptable a distintos volúmenes de información.

En la parte superior se encuentra un buscador de tareas, que permite localizar rápidamente tareas específicas mediante palabras clave, lo que resulta especialmente útil cuando la cantidad de registros es elevada. A su derecha, un botón flotante con el icono de suma (+) da acceso directo al formulario de creación de una nueva tarea, facilitando así su registro en cualquier momento.

Debajo del buscador, se presentan los distintos estados de las tareas:

- Totas: muestra el total de tareas sin importar su estado.
- Programadas: agrupa las tareas que tienen una fecha concreta asignada.
- Urgentes: destaca aquellas tareas con un nivel de prioridad elevado que requieren atención inmediata.

- Por hacer: incluye todas las tareas que aún no han comenzado, es decir, están pendientes de ser iniciadas.
- En proceso: reúne aquellas tareas que ya han sido iniciadas, pero aún no han sido finalizadas.
- Completadas: muestra las tareas que ya han sido finalizadas y marcadas como realizadas por el usuario.

Estas tarjetas permiten al usuario obtener una visión rápida del número de tareas existentes en cada estado y acceder directamente a cada listado filtrado.

Más abajo, la interfaz presenta un bloque denominado "Mis categorías", que organiza las tareas según categorías personalizadas creadas por el usuario (por ejemplo: Trabajo, Casa, GYM, etc.). Cada categoría se muestra en forma de tarjeta con su color distintivo, el número total de tareas que contiene y un ícono de flecha que permite acceder al detalle de esa categoría específica. Esta agrupación visual mejora la comprensión y navegación, favoreciendo una mejor planificación de las tareas según su contexto o ámbito.

Esta vista centraliza así la gestión de tareas de forma clara y funcional, ofreciendo múltiples puntos de entrada a la información sin saturar visualmente al usuario.

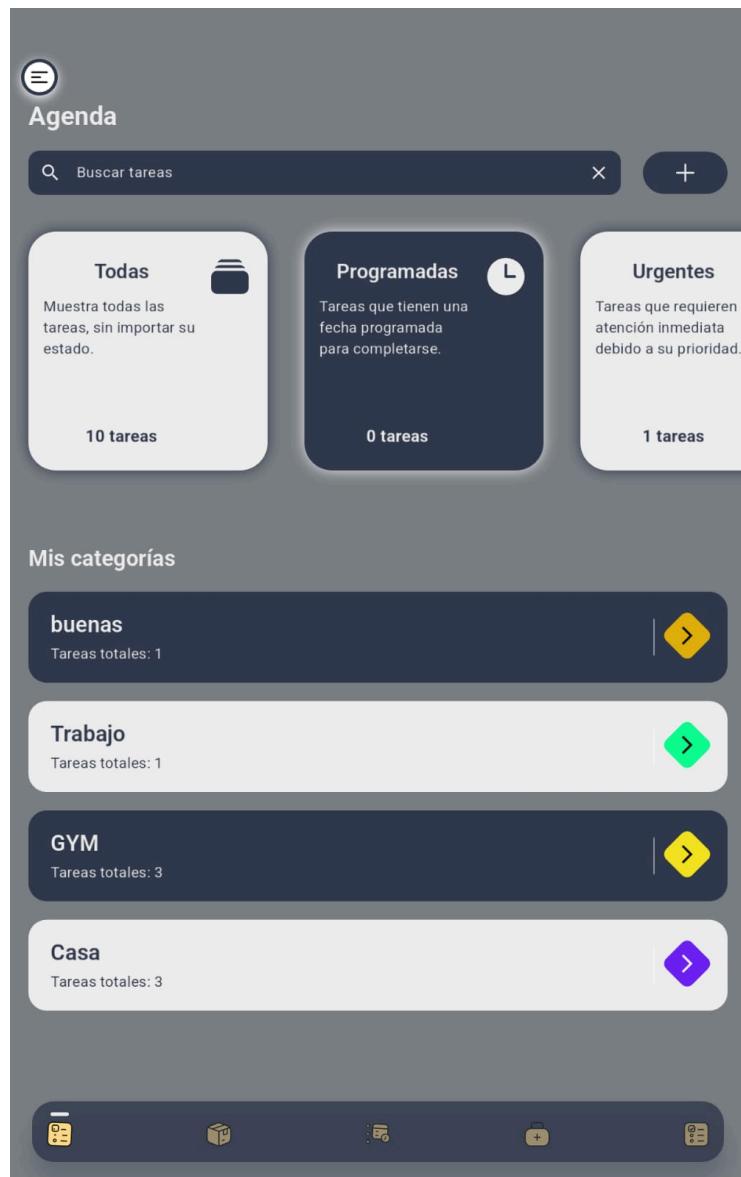


Figura 13. Vista general tareas

Cada categoría, ya sea correspondiente a un estado de la tarea (como Por hacer, En proceso o Completadas) o a una clasificación personalizada (como Casa, Trabajo o GYM), cuenta con una vista dedicada en la que se agrupan y muestran todas las tareas asociadas. Esta vista facilita al usuario la consulta y gestión de las tareas de forma segmentada, promoviendo una organización clara y eficiente.

En la parte superior se encuentra una barra de búsqueda que permite filtrar rápidamente las tareas por nombre o contenido, así como un botón flotante de creación rápida ("+"") para añadir nuevas tareas directamente desde esa categoría o estado.

Justo debajo se presentan las tareas en formato de tarjeta, tal como se muestra en la Figura 14. Cada tarjeta incluye de forma resumida y visualmente atractiva los datos más relevantes de la tarea: el nombre, el nivel de prioridad, los perfiles asignados, una barra de progreso que indica el avance, una descripción breve y la categoría correspondiente.

Además, cada tarjeta cuenta con un icono de tres puntos que despliega un menú contextual con distintas opciones de gestión. Este menú permite realizar acciones como editar, eliminar, duplicar o actualizar el progreso de la tarea de manera rápida y directa, sin necesidad de acceder a la vista detallada. Gracias a este diseño, el usuario puede interactuar fácilmente con sus tareas, manteniendo un control total con tan solo unos pocos toques.

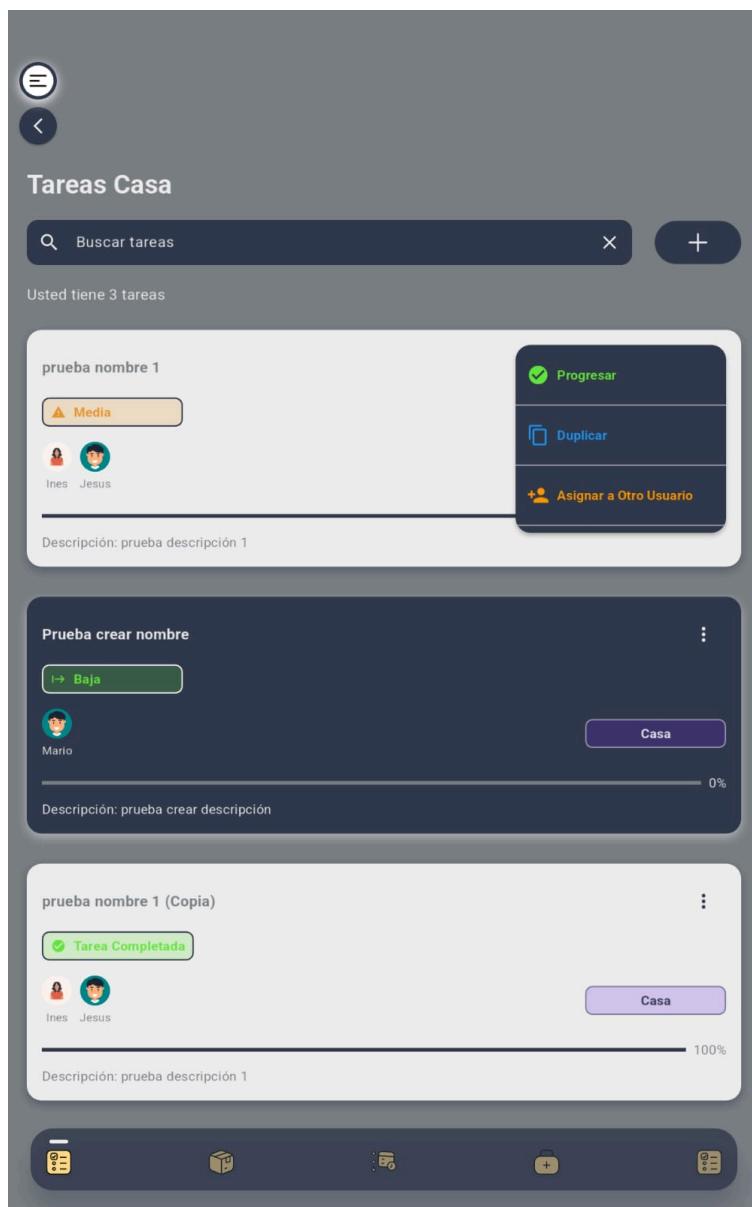


Figura 14. Vista filtrada tareas

3.4.3 Implementación técnica

La capa técnica del módulo de tareas se ha diseñado para ser reactiva, flexible y consistente con el resto de la aplicación. A continuación se detallan los mecanismos principales que lo hacen posible:

En lugar de mantener listas estáticas, las categorías de tareas se generan dinámicamente analizando los atributos de cada instancia de tarea. Por ejemplo, para construir la sección “Por hacer” se filtran todas las tareas cuyo campo Progreso sea igual a 0 (pendiente). De igual modo, la categoría “En proceso” agrupa aquellas con Progreso == 1 y “Completadas” las de Progreso == 2. Las secciones “Programadas” o “Urgentes” se crean filtrando según la presencia de una fecha en IdEvento o si la prioridad supera un umbral predefinido. Gracias a este enfoque, cualquier cambio en los datos de la tarea (como la actualización de su progreso o prioridad) provoca que automáticamente entren o salgan de la categoría correspondiente, sin necesidad de mantenimiento manual de las listas.

Para mostrar los perfiles responsables de cada tarea se reutiliza el mismo diseño empleado en el módulo de inventario: cada perfil se representa mediante una tarjeta o un Circle Avatar acompañado de su nombre. Este componente común garantiza coherencia visual entre módulos y facilita la identificación rápida de los usuarios asignados. Al pulsar sobre el avatar, se despliega información adicional o se permite navegar al perfil correspondiente, tal como ocurre en la sección de productos.

Se ha integrado un gestor de estado (por ejemplo, Provider o Riverpod) que vigila el conjunto de tareas en memoria. Cada operación de creación, edición o eliminación se envía a la API y, una vez confirmada la respuesta, se actualiza el estado local. Gracias a esto, la UI se reconstruye de forma inmediata, mostrando las nuevas tareas, los cambios de asignación o el progreso actualizado sin necesidad de refrescar toda la pantalla. Esta reactividad garantiza que los usuarios siempre vean datos sincronizados y evita inconsistencias.

Con estos tres pilares —clasificación dinámica, componentes visuales compartidos y reactividad inmediata— el módulo de tareas de FamSync ofrece una experiencia de uso coherente, eficiente y altamente adaptable a los cambios que se producen en el día a día de la gestión familiar.

3.5 Gestión de Eventos

El módulo de eventos de FamSync está diseñado para ayudar a las familias a organizar y gestionar los compromisos y actividades programadas de sus miembros de manera eficiente. Al igual que en el caso de las tareas, los eventos pueden ser asignados a uno o varios perfiles familiares, permitiendo una planificación conjunta y una mejor coordinación de horarios.

Para representar internamente los eventos programados, FamSync define una estructura que recoge todos los atributos necesarios para su gestión y asociación con usuarios, perfiles y tareas. En la siguiente tabla se detallan los campos principales que conforman la entidad Evento:

Campo	Tipo de dato	Descripción
Id	int	Identificador único del evento en la base de datos.
Nombre	String	Título descriptivo del evento, utilizado para su identificación rápida.
Descripción	String	Texto libre con información adicional o instrucciones relacionadas con el evento.
FechaInicio	String	Fecha y hora de inicio del evento, en formato ISO (por ejemplo, "2025-05-01T14:00").
FechaFin	String	Fecha y hora de finalización del evento, en formato ISO.
IdUsuarioCreador	int	Identificador del usuario que creó el evento.
IdPerfilCreador	int	Identificador del perfil familiar que originó el evento, para asociarlo a un miembro concreto.
IdCategoria	int	Identificador de la categoría a la que pertenece el evento (por ejemplo: médico, colegio, ocio).
Participantes	List<int>	Lista de IDs de perfiles invitados o asignados al evento.
IdTarea	int? (nulleable)	Enlace opcional a una tarea relacionada (si existe), para sincronizar calendario y gestión de responsabilidades.

Tabla 6. Estructura de Evento

Cada evento representa una actividad específica, ya sea de tipo social, médico, educativo o cualquier otro aspecto de la vida diaria. A través de este módulo, los usuarios pueden crear, visualizar, editar y eliminar eventos, y asignarlos a los perfiles correspondientes. De esta forma, cada miembro de la familia puede acceder a su agenda personal y mantenerse al tanto de los compromisos de los demás.

Más allá del simple registro de actividades, el módulo de eventos ofrece una serie de funcionalidades específicas orientadas a facilitar su uso diario:

- Creación rápida de eventos personalizados, con campos editables para título, descripción, categoría, fecha y hora.
- Asignación múltiple de participantes, permitiendo invitar a varios perfiles familiares, que serán notificados y podrán visualizar el evento en su calendario personal.
- Sincronización con tareas relacionadas, lo que permite transformar tareas con fecha en eventos temporales, evitando duplicidades.
- Categorización temática de eventos (como médicos, escolares, personales, etc.), lo que mejora la organización y el filtrado.
- Visualización clara en distintas vistas (diaria, mensual o lista completa), adaptadas a diferentes necesidades del usuario.
- Edición y eliminación de eventos en cualquier momento, con actualización automática para todos los perfiles implicados.

Gracias a este conjunto de funcionalidades, FamSync no solo permite gestionar compromisos individuales o colectivos, sino que convierte el calendario en un verdadero centro de coordinación familiar. Esta flexibilidad facilita la organización del día a día, mejora la comunicación entre perfiles y reduce la posibilidad de olvidos o solapamientos.

3.5.1 Asignación de tareas a eventos

FamSync permite asociar tareas con eventos del calendario de forma directa, integrando así ambos módulos para facilitar la planificación de actividades que requieren una programación temporal más concreta. Esta funcionalidad está pensada para aquellos casos en los que una tarea, además de necesitar seguimiento, debe realizarse en un día y hora específicos, como por ejemplo una cita médica, una reunión escolar o una limpieza programada.

La asociación de una tarea a un evento se realiza desde el propio menú contextual de la tarea, a través de la opción “Añadir al calendario”. Al seleccionarla, la aplicación solicita al usuario la información necesaria para completar el evento, como la fecha y hora de inicio y finalización.

A partir de ahí, se genera automáticamente una instancia de evento que recoge los datos esenciales de la tarea original: su nombre, descripción, perfiles asignados y categoría, manteniendo así la coherencia entre módulos y evitando la duplicación de información.

En la Figura 15, se muestra el formulario de creación de un evento generado a partir de una tarea existente. Como puede observarse, parte de los campos ya se encuentran pre-llenados con la información extraída de la tarea, y el usuario únicamente debe completar los datos relativos a la planificación temporal.

Una vez vinculada, la tarea quedará conectada al evento mediante un identificador común, lo que permite desde el propio calendario acceder a los detalles de la tarea asociada. Esta integración favorece una gestión más visual y estructurada, permitiendo que las actividades planificadas puedan verse tanto como tareas en seguimiento como compromisos programados dentro del calendario familiar.

The screenshot displays a mobile application interface for creating an event from an existing task. The title "Crear Evento a partir de Tarea" is at the top. Below it, there are four main sections:

- Nombre del Evento:** Lavar los platos
- Descripción del Evento:** Realizar la limpieza de los utensilios de cocina.
- Categoría del evento:** (dropdown menu)
- Perfiles destinatarios:** Mario, Ines, Jesus (with small user icons)

At the bottom, there is a calendar section showing the month of April 2025, with April 22nd highlighted. A large "Registrar Evento" button is located at the very bottom of the form.

Figura 15. Creación de evento mediante tarea

3.5.2 Visualización en formato Calendario

El módulo de eventos de FamSync ofrece un sistema de visualización que permite consultar las actividades programadas de manera clara y estructurada. Este sistema se compone de tres vistas principales, cada una enfocada en un nivel distinto de profundidad y detalle: la vista de eventos diarios, la vista de todos los eventos y la vista de detalles de un evento concreto.

La vista de eventos diarios que se muestra en la Figura 16, es la principal y está diseñada para mostrar de forma ordenada los eventos relacionados con la jornada actual. Al acceder a esta vista, se presenta primero la fecha del día en curso, seguida de un bloque con los eventos próximos, es decir, aquellos programados para fechas posteriores. Más abajo, se muestran los eventos programados para el día actual, organizados cronológicamente. Cada uno se representa mediante una tarjeta visual que incluye elementos clave como el nombre del evento, la fecha y hora de inicio y finalización, y los perfiles participantes. Esta vista está pensada para ofrecer un acceso rápido a las actividades más relevantes e inmediatas.

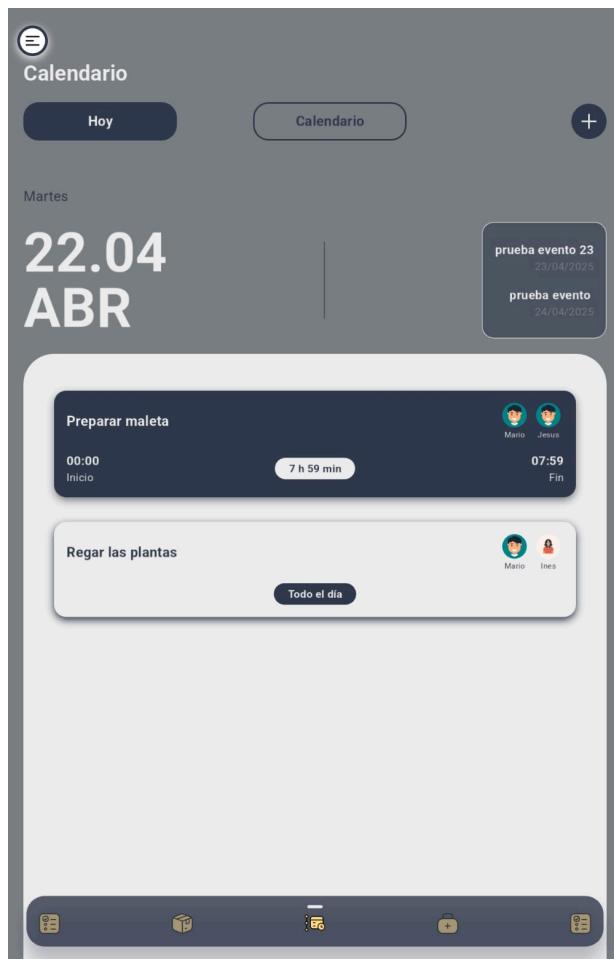


Figura 16. Página de eventos diarios

La segunda es la vista de todos los eventos, como muestra la Figura 17, donde se muestra una lista general con todos los eventos registrados en la aplicación, sin importar la fecha en la que se celebren. Esta vista está orientada a ofrecer una perspectiva global, útil para localizar rápidamente un evento pasado o próximo, consultar el historial de actividades o repasar la carga de planificación familiar. Los eventos se muestran en orden cronológico, y el usuario puede desplazarse fácilmente para encontrar el evento que busca.

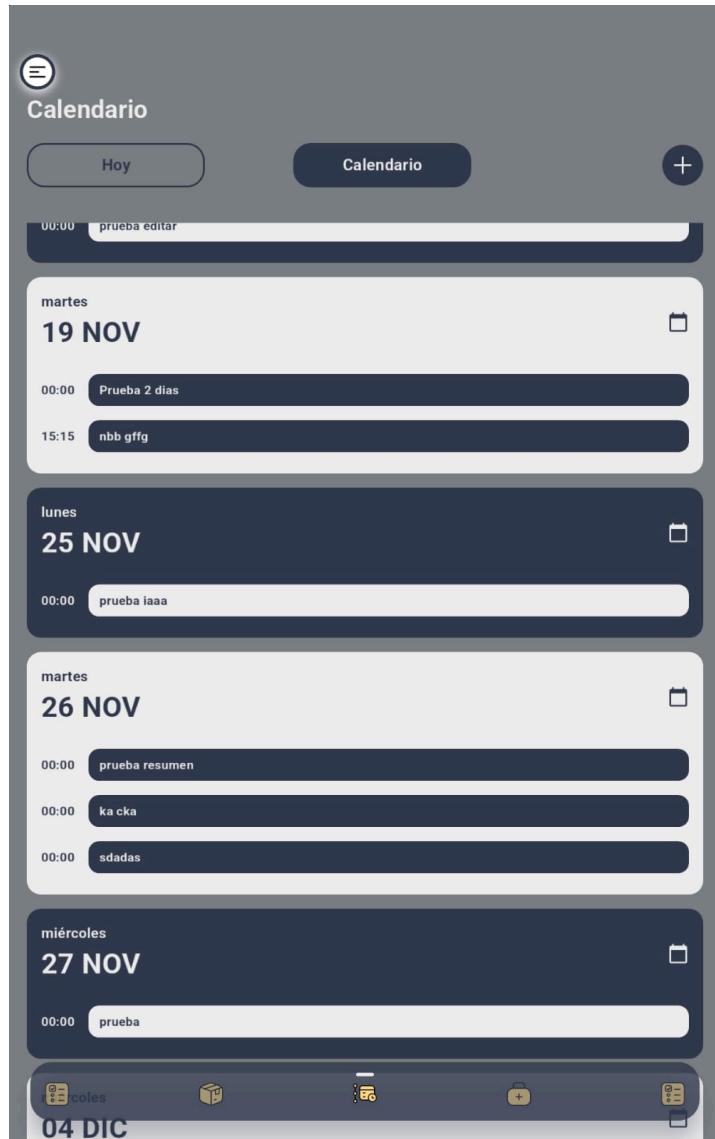


Figura 17. Página de todos los eventos

Por último, la vista de detalles de un evento se activa al pulsar sobre cualquier tarjeta de evento, ya sea desde la vista diaria o desde la lista completa. Esta vista muestra toda la información asociada al evento seleccionado: el nombre, la descripción completa, la categoría, la duración total, los perfiles implicados, el creador del evento y, si corresponde, el vínculo con una tarea asociada.

Esta vista permite consultar con precisión todos los datos relacionados con un compromiso concreto, como se aprecia en la Figura 18, y está diseñada para facilitar tanto la comprensión del evento como su posterior edición o eliminación.

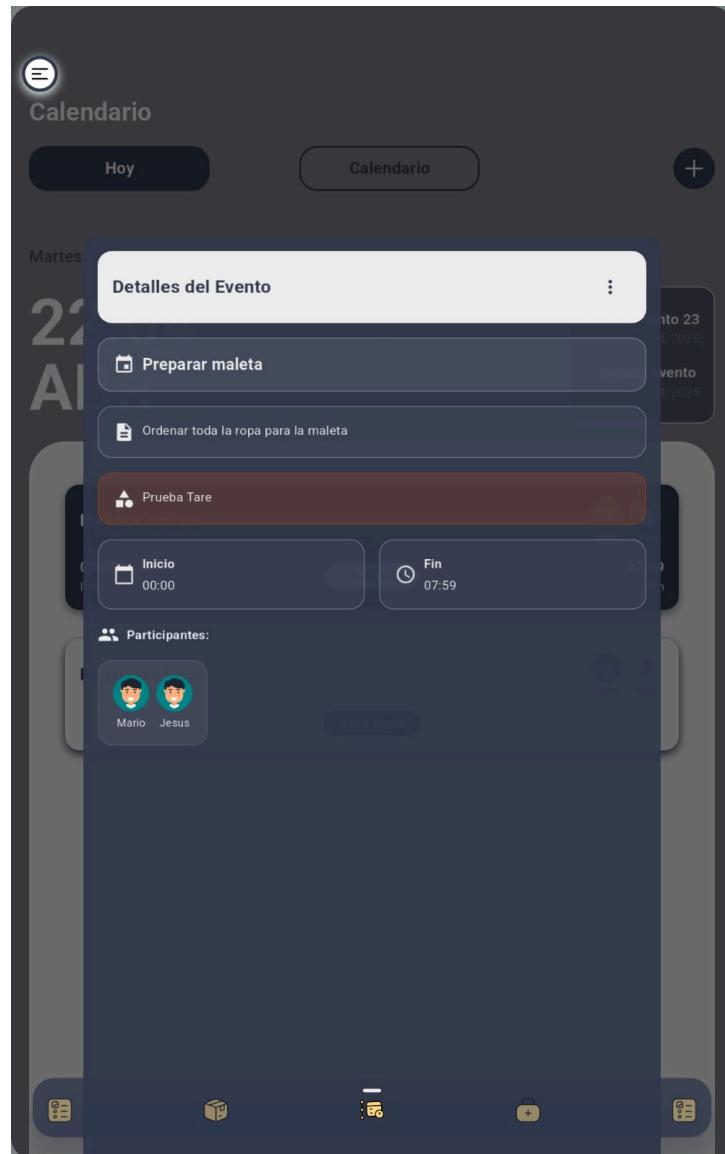


Figura 18. Página de detalles de evento

Estas tres vistas trabajan de forma coordinada para ofrecer una experiencia flexible, intuitiva y accesible, adaptada tanto a usuarios que buscan un resumen rápido como a aquellos que necesitan profundizar en la planificación familiar detallada.

3.5.3 Implementación técnica

La implementación técnica del módulo de eventos en FamSync se ha llevado a cabo siguiendo principios de coherencia con el resto de módulos, reutilizando componentes y patrones ya establecidos, como el gestor de estado global, la sincronización en tiempo real con la base de datos y la estructura modular de vistas.

Cuando se crea o edita un evento, la aplicación se comunica con la API REST para registrar o actualizar la información correspondiente. Una vez confirmada la operación por el servidor, el gestor de estado actualiza los eventos locales, lo que permite que los cambios se reflejen de forma inmediata en la interfaz del usuario, sin necesidad de reiniciar la aplicación o recargar manualmente las vistas.

Para renderizar los eventos en las distintas vistas (diaria o general), el sistema utiliza filtros dinámicos en función de la fecha actual y la información contenida en cada evento. Los días con eventos se marcan visualmente en el calendario, y los datos se agrupan y ordenan cronológicamente para facilitar la consulta por parte del usuario.

Además, al tratarse de un módulo conectado con otros como el de tareas o perfiles, la lógica de asociación se basa en relaciones entre identificadores únicos, lo que permite, por ejemplo, acceder desde un evento a la tarea relacionada o ver los participantes del evento en formato visual reutilizando los componentes de perfil ya empleados en otras secciones de la aplicación.

Con este enfoque técnico, se garantiza que el sistema de eventos sea ágil, escalable y fácilmente integrable con futuras funcionalidades como notificaciones, recordatorios automáticos o sincronización externa con otros calendarios personales.

3.6 Pruebas y despliegue

Una vez finalizado el desarrollo funcional de la aplicación FamSync, resulta imprescindible validar su correcto funcionamiento mediante un proceso de pruebas estructurado. Este proceso permite detectar errores, comprobar que se cumplen los requisitos establecidos y garantizar una experiencia de usuario fluida y estable. Además, tras la fase de validación, se procede al despliegue de la aplicación en un entorno real, lo que implica preparar su instalación en dispositivos móviles, configurar la conexión con la API y asegurar su disponibilidad para los usuarios finales. En este apartado se detallan tanto las estrategias y tipos de pruebas llevadas a cabo como el procedimiento seguido para realizar el despliegue de la aplicación.

Una vez completado el desarrollo funcional de FamSync, se llevó a cabo un proceso de validación enfocado a comprobar la estabilidad, utilidad y comportamiento real de la aplicación. Este proceso incluyó tanto pruebas manuales e internas como una fase de evaluación en manos de usuarios reales, con el objetivo de identificar errores, confirmar que se cumplían los requisitos y evaluar la experiencia de uso desde una perspectiva externa.

3.6.1 Pruebas realizadas

Durante las fases finales del proyecto, se proporcionó una versión funcional de la aplicación a varios miembros del círculo cercano del desarrollador —personas no relacionadas con el ámbito del desarrollo de software— y se les animó a utilizarla de manera natural, como si formara parte de su rutina diaria. Esta estrategia tuvo como objetivo obtener una retroalimentación honesta y espontánea, libre de tecnicismos, que permitiera detectar tanto puntos fuertes como áreas de mejora desde el punto de vista del usuario final.

Estas pruebas permitieron validar que las funcionalidades principales del sistema funcionaban correctamente en escenarios reales. Los usuarios crearon y editaron tareas, gestionaron productos en el inventario, planificaron eventos en el calendario familiar, y utilizaron las listas de la compra de forma colaborativa. Además, se comprobó que el sistema de perfiles funcionaba correctamente, permitiendo que una única cuenta de usuario gestionara varios perfiles familiares con distintos permisos y accesos.

A nivel técnico, también se realizaron pruebas funcionales para verificar que cada módulo cumplía los requisitos previamente definidos. Se comprobó la correcta ejecución de operaciones básicas como la creación, modificación y eliminación de entidades (tareas, productos, eventos, listas, etc.), así como la correcta visualización de los datos en las diferentes vistas.

Por otro lado, se validó la integración entre el frontend desarrollado en Flutter y el backend basado en Node.js + Express. A través de llamadas HTTP a la API REST, se verificó que la transferencia de datos era correcta, que los errores eran gestionados adecuadamente y que las operaciones se reflejaban en tiempo real dentro de la aplicación.

Durante estas pruebas, se detectaron varios errores y detalles de usabilidad que fueron corregidos antes de finalizar el proyecto, como inconsistencias en la validación de formularios, problemas con la carga condicional de elementos en pantalla o fallos puntuales en la navegación. Gracias a la colaboración de los usuarios de prueba, se pudieron resolver estas incidencias y mejorar la experiencia general.

3.6.2 Herramientas utilizadas durante las pruebas

Para llevar a cabo las pruebas y el seguimiento de errores, se utilizaron las siguientes herramientas:

- Dispositivo Android real: se instaló una versión de la aplicación (.APK) en varios teléfonos móviles del entorno cercano para realizar pruebas en condiciones reales de uso.
- Postman: para probar manualmente los endpoints del backend, validar el funcionamiento de la API y verificar los datos enviados y recibidos.
- Flutter DevTools: utilizado durante el desarrollo y pruebas para depurar la interfaz y verificar el comportamiento de los widgets y del estado de la aplicación.
- Android Studio: como entorno principal de desarrollo y ejecución, así como para simular diferentes tamaños de pantalla mediante emuladores.

3.6.3 Despliegue

El despliegue de la aplicación se llevó a cabo de manera local y manual en dispositivos Android, generando el archivo .apk desde Android Studio. Posteriormente, el archivo fue transferido directamente a los dispositivos mediante conexión por cable o a través de servicios de almacenamiento en la nube, permitiendo que los usuarios lo instalaran y ejecutaran sin necesidad de acudir a plataformas oficiales como Google Play.

En cuanto al backend, tanto la API desarrollada con Node.js como la base de datos MySQL se alojaron en un servidor local, funcionando como servidor principal durante las pruebas. Para permitir que otros dispositivos, fuera de la red local, pudieran conectarse al servidor, se configuró una redirección de puertos en el router del domicilio, exponiendo así el servicio a través de la dirección IP pública del hogar. Este enfoque permitió que los usuarios de prueba, conectados desde distintas redes, pudieran acceder correctamente a la API y a los datos almacenados en la base de datos, como si el sistema estuviera desplegado en un entorno en la nube.

Aunque esta solución no es la ideal para un entorno de producción, ha sido completamente funcional durante las pruebas, permitiendo validar el comportamiento de la aplicación en condiciones reales de uso. Además, demuestra la viabilidad del sistema en situaciones donde no se dispone de un servidor externo o infraestructura profesional.

4. Casos de Uso

El apartado de casos de uso tiene como finalidad mostrar, de forma práctica, cómo interactúan los usuarios con la aplicación FamSync en distintos escenarios cotidianos. A través de estos ejemplos se pueden comprender mejor las funcionalidades principales, la navegación entre módulos y el valor real que la herramienta aporta a la organización de la vida familiar diaria.

FamSync ha sido diseñada con un enfoque centrado en la usabilidad y en la resolución de problemas reales dentro del entorno familiar. Por ello, los casos de uso que se muestran a continuación están basados en situaciones comunes, como la planificación de tareas del hogar, la gestión de eventos importantes, la organización de las compras y la consulta de actividades a través del calendario.

Cada caso describe una situación concreta, el objetivo del usuario y los pasos necesarios para conseguirlo dentro de la aplicación. Además, se acompaña de capturas ilustrativas (que pueden consultarse en las figuras correspondientes), lo que permite al lector visualizar cómo se traduce la lógica de funcionamiento de FamSync en una experiencia real de uso.

4.1 Asignación de una tarea a varios perfiles

Un padre desea repartir una tarea del hogar entre sus hijos, para fomentar la responsabilidad compartida. Para ello:

- Identificarse como el padre indicando su Pin correspondiente.
- Accede al módulo de tareas desde la pantalla principal.
- Pulsa en el botón de creación y escribe el título “Recoger los juguetes”.
- Añade una descripción breve, marca la tarea como “Por hacer” y asigna la responsabilidad a los perfiles de sus dos hijos.
- Guarda la tarea, que aparece automáticamente en la vista de ambos perfiles.

Gracias a este flujo, cada perfil puede consultar la tarea desde su sección personal, seguir su progreso, y actualizar su estado cuando esté realizada.

The screenshot shows a mobile application interface for creating a new task. At the top, there are navigation icons: a circular arrow and a back arrow. Below them is the title "Crear Tarea". The form consists of several input fields and buttons:

- Nombre de la Tarea:** A text input field containing "Recoger juguetes".
- Descripción de la Tarea:** A text input field containing "Recoger juguetes lo más pronto posible".
- Categoría de la Tarea:** A dropdown menu showing "Casa" selected.
- Perfiles destinatarios:** A section listing three user profiles: "Hijo", "Hija", and "Padre", each with a small profile icon.
- Prioridad:** A horizontal button bar with three options: "Baja" (highlighted in green), "Media" (highlighted in orange), and "Alta" (highlighted in red).
- Registrar Tarea:** A large, light-colored button at the bottom center.

Figura 19. Formulario creación de tarea

En la Figura 19, se muestra el formulario de creación de una nueva tarea dentro del módulo correspondiente. En él pueden apreciarse los distintos campos que el usuario debe completar: el nombre de la tarea, su descripción, la selección de prioridad y progreso, así como la asignación de uno o varios perfiles responsables. Esta interfaz ha sido diseñada para ser clara e intuitiva, permitiendo configurar la tarea de forma rápida sin renunciar a la personalización.

Por otro lado, si accedemos desde el perfil de uno de los niños, se puede observar cómo dicha tarea queda reflejada en su vista general de tareas (Figura 20).

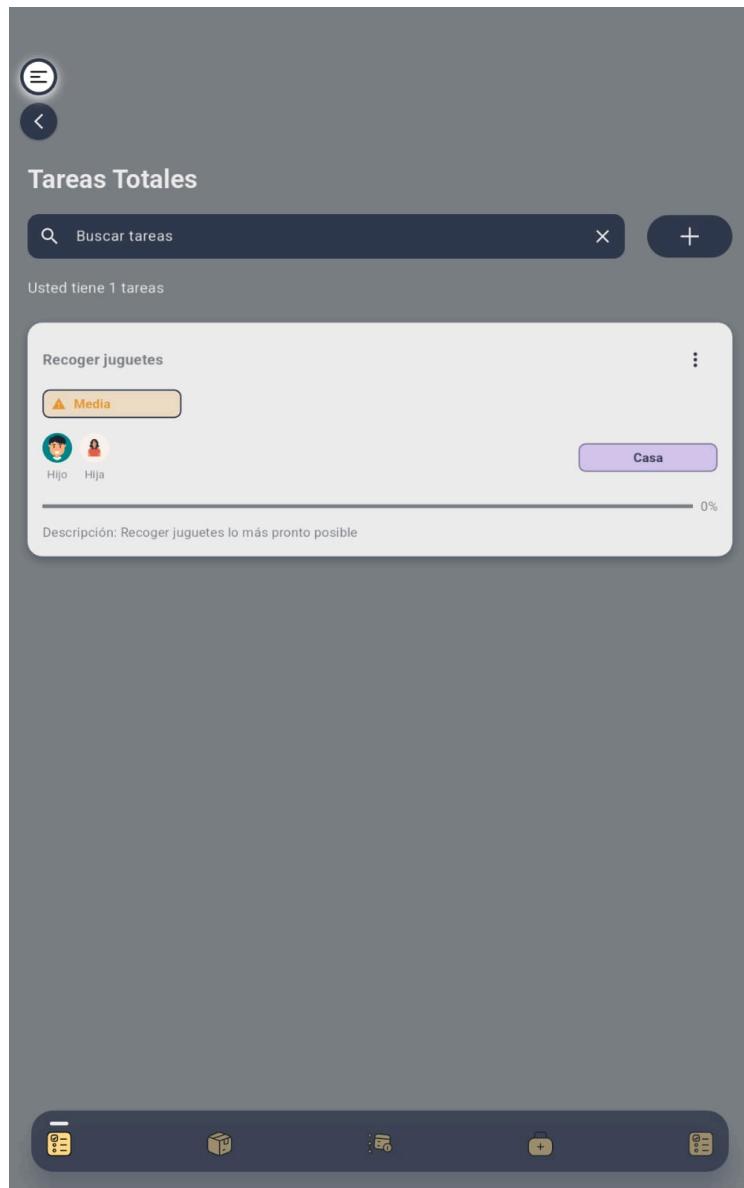


Figura 20. Visualización nueva tarea

La tarjeta muestra los elementos más importantes de forma visual y accesible, incluyendo el título, los perfiles asignados mediante iconos, y una barra de progreso. Esta vista permite al usuario seguir el estado de las tareas asignadas sin necesidad de navegar entre menús, contribuyendo así a una experiencia más fluida y organizada.

4.2 Planificación de un evento a partir de una tarea

El padre crea una tarea para recordar que debe llevar a su padre al médico el próximo jueves, pero quiere que también aparezca en el calendario familiar. Para ello:

- Crea la tarea “Llevar al abuelo al médico” desde el módulo de tareas.
- Una vez creada, pulsa sobre el icono de tres puntos para abrir el menú contextual.
- Selecciona “Añadir al calendario” y completa la fecha y la hora del evento.
- Se crea automáticamente un evento asociado a la tarea, que hereda su nombre, participantes y descripción.
- El evento queda reflejado tanto en la vista diaria como en la vista general del calendario, marcado en el día correspondiente.

De esta forma, cualquier tarea con fecha puede visualizarse también como evento, permitiendo una planificación más estructurada y visual, como se aprecia en la Figura 21.

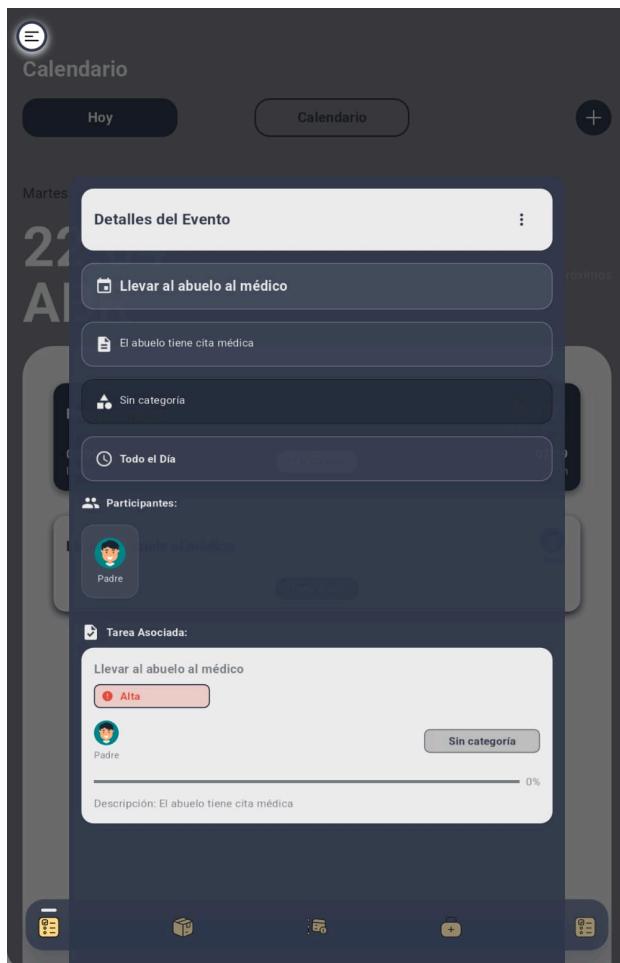


Figura 21. Evento creado mediante Tarea

4.3 Creación de una lista de la compra semanal

Un usuario desea organizar las compras del fin de semana para toda la familia. Utiliza el módulo de listas de la compra de FamSync para ello:

- Accede al módulo de inventario y pulsa en “Crear nueva lista”.
- Asigna el nombre “Compra del sábado” y define qué perfiles tendrán acceso a ella (Figura 22).
- Añade productos previamente registrados en el inventario, como leche, pan, fruta o detergente (Figura 23).

Esta función facilita una compra más rápida, organizada y colaborativa, permitiendo incluso que varios perfiles consulten o modifiquen la lista en tiempo real.

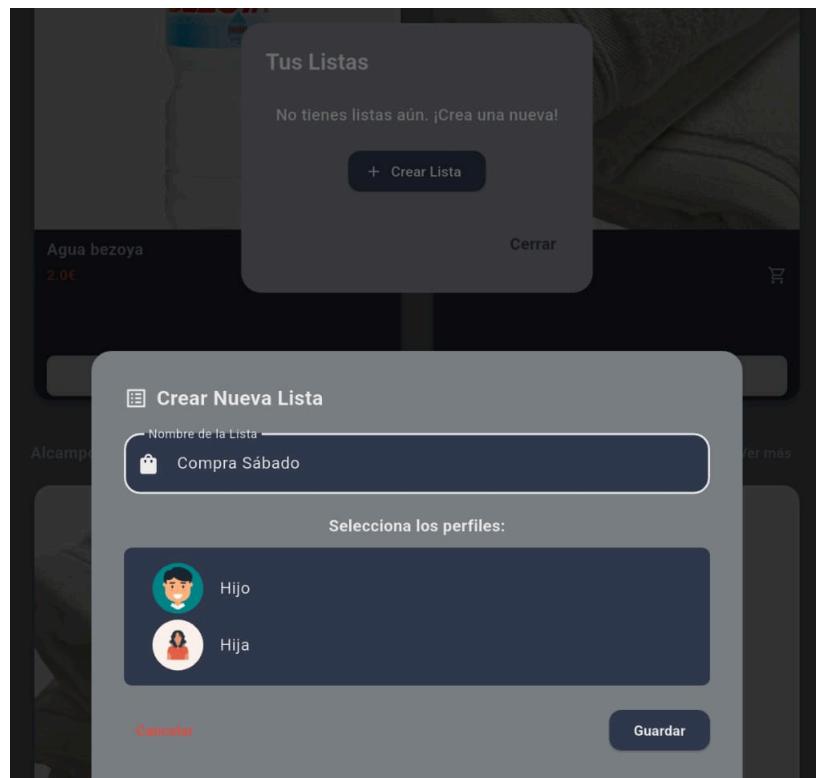


Figura 22. Creación de lista

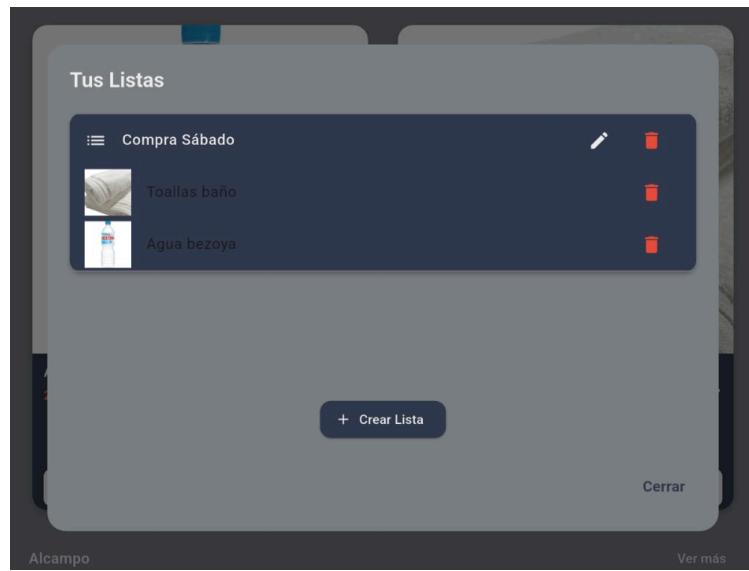


Figura 23. Visualización de lista

4.4 Consulta del calendario familiar en vista mensual

Un miembro de la familia quiere consultar todos los eventos del mes de noviembre para organizar una escapada familiar sin coincidir con otras actividades. Para ello:

- Accede al módulo de eventos y cambia a la vista general del calendario.
- Navega hasta el mes de noviembre, donde observa qué días tienen eventos marcados.
- Al pulsar sobre un día señalado, puede ver los eventos programados para esa fecha y acceder a sus detalles.
- Selecciona un evento para ver su duración, descripción, y qué perfiles están implicados.

Esta vista, como se aprecia en la Figura 24, ofrece una perspectiva global de la planificación familiar, permitiendo evitar solapamientos y tomar mejores decisiones sobre nuevas actividades.

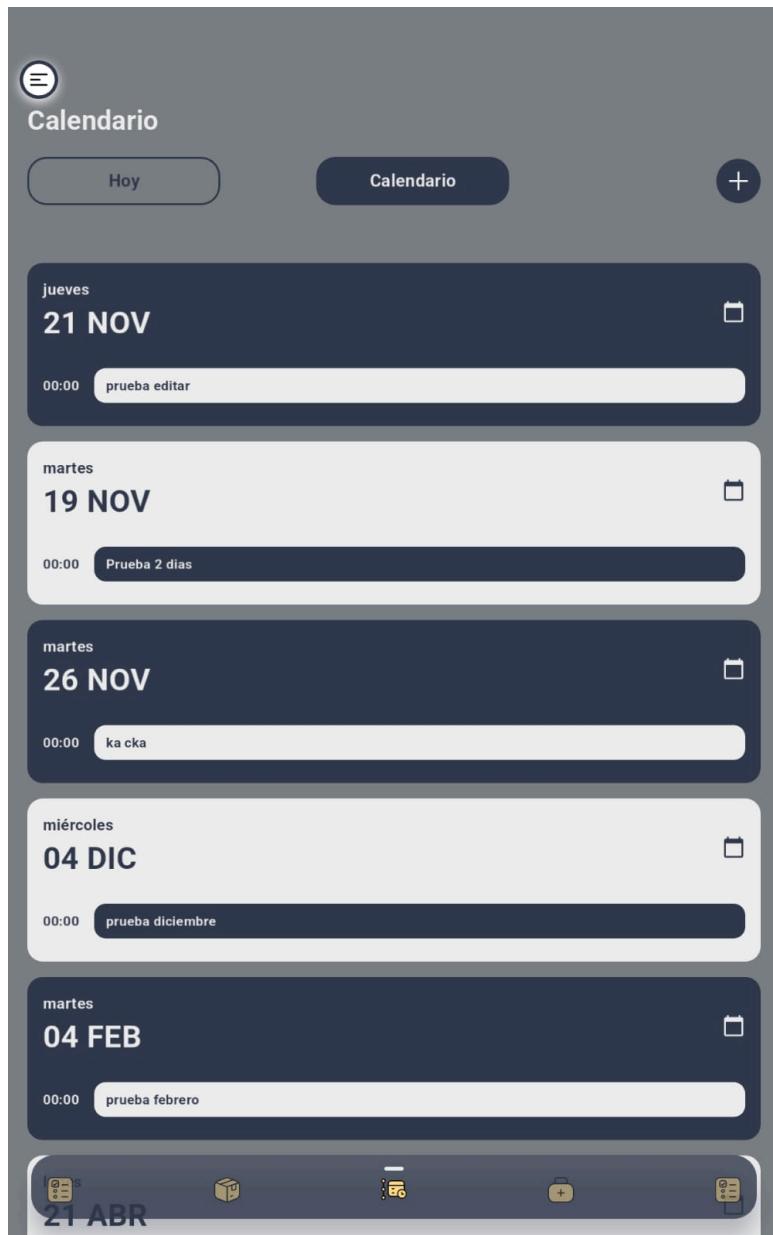


Figura 24. Visualización de eventos

5. Conclusiones y trabajo futuro

FamSync es una aplicación móvil desarrollada con el objetivo de mejorar la organización familiar mediante una plataforma centralizada y colaborativa. A lo largo del desarrollo del proyecto se ha logrado implementar un sistema funcional que permite gestionar tareas, eventos y recursos compartidos en el entorno doméstico.

A continuación, se va a realizar una evaluación de los objetivos planteados al inicio del proyecto en relación con los resultados obtenidos. Además, se presentarán posibles mejoras e implementaciones futuras que podrían hacer que la aplicación FamSync sea más completa, versátil y funcional para su uso cotidiano en el entorno familiar.

Tal y como se ha expuesto a lo largo de esta memoria, se ha alcanzado el objetivo principal del proyecto: desarrollar una aplicación móvil destinada a facilitar la organización y coordinación de las tareas, eventos y recursos de una unidad familiar, centralizando toda la información en una única plataforma accesible y colaborativa.

Las funcionalidades principales previstas y que se han conseguido implementar con éxito han sido:

- Crear un sistema de gestión de tareas, con asignación de responsables, estados de progreso, prioridad y seguimiento.
- Desarrollar un módulo de eventos familiares, con visualización en formato calendario y asociación directa con tareas.
- Implementar un inventario compartido de productos, organizado por tiendas, con opción de añadir imágenes, precios y control de visibilidad por perfil.
- Incorporar un sistema de listas de la compra colaborativas, con posibilidad de notificaciones y control de productos marcados como adquiridos.
- Gestionar usuarios y perfiles familiares de forma independiente, permitiendo que un usuario administre varios perfiles sin necesidad de que todos tengan cuenta propia.
- Asegurar una experiencia de usuario accesible y amigable para distintos perfiles, incluyendo adultos mayores o personas con menor familiaridad tecnológica.

Junto con el desarrollo funcional, también se han cumplido otros objetivos fundamentales relacionados con la planificación y diseño del sistema:

- Se ha realizado un estudio de aplicaciones similares para identificar carencias en soluciones existentes y proponer una alternativa más integral.

- Se han definido los requisitos funcionales y no funcionales del sistema antes del desarrollo, incluyendo el diseño de la base de datos, la API REST y la estructura modular de la app.
- Se ha llevado a cabo un proceso iterativo de pruebas y correcciones, obteniendo una versión estable de la aplicación, funcional en dispositivos móviles Android.

Aunque el proyecto ha cumplido los objetivos propuestos, existen múltiples posibilidades de ampliación y mejora que podrían abordarse en futuras versiones. Algunas ideas destacadas son:

- Inclusión de un sistema de notificaciones inteligentes que sugiera tareas o compras según el historial de uso.
- Integración con calendarios externos (como Google Calendar) para facilitar la compatibilidad con herramientas ya utilizadas por los usuarios.
- Implementación de recordatorios automáticos para tareas críticas o eventos con fechas próximas.
- Mejora en la gestión de permisos y roles, permitiendo, por ejemplo, limitar ciertas acciones según la edad o el tipo de perfil.
- Ampliación del módulo de inventario con un historial de consumo o alertas por productos caducados o faltantes.
- Mejora visual y accesibilidad de la interfaz para adaptarse a distintos niveles de alfabetización digital.
- Publicación oficial en plataformas como Google Play o App Store.
- Desarrollo de nuevos módulos como Medicina, ropa, etc...

En conclusión, se ha logrado desarrollar una aplicación totalmente funcional, centrada en la mejora de la vida familiar mediante la organización, la colaboración y la centralización de la información diaria. FamSync sienta así las bases de una solución tecnológica adaptable a diferentes modelos de familia, con potencial real de crecimiento y evolución.

Referencias

- [1] Contreras, A., Ramírez, F., & Moreno, J. (2022). Aplicaciones móviles en la gestión del hogar: una revisión sistemática. *Revista Iberoamericana de Tecnología Educativa*, 18(2), 45-61.
- Los dispositivos móviles en el hogar. Interés formativo de las familias españolas | Request PDF
- [2] Muñoz, D., & García, M. (2021). La digitalización del entorno familiar: retos y oportunidades. *Tecnología y Sociedad*, 29(1), 22-38.
- Retos y Oportunidades de la Educación Digital.pdf
- [3] TFG. Trabajo fin de grado. Grado en Ingeniería Multimedia
- [6] Bailon, Guillermo (2019) ¿Por qué utilizar Scrum en la gestión de proyectos?
¿Por qué utilizar Scrum en la gestión de proyectos? - TAKTIC
- [7] TechCrunch. (2021). Cozi: la app que digitaliza la vida familiar.
<https://techcrunch.com/cozi-app-review>
- [8] López, M., Sánchez, R., & Díaz, C. (2020). Soluciones digitales para la vida familiar: revisión de aplicaciones móviles. *Revista Española de Informática*, 12(4), 31-47.
- [9] Cozi Family Organizer
Sitio web oficial: <https://www.cozi.com/>
- [10] FamilyWall
Sitio web oficial: <https://familywall.com/>
- [11] OurHome
Sitio web oficial: <https://www.ourhomeapp.com/>
- [12] TimeTree
Sitio web oficial: <https://timetreeapp.com/>
- [13] Mozilla. (2024). JavaScript Guide. Mozilla Developer Network (MDN).
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>

[14] Google. (2024). Dart language overview. Dart.dev.

<https://dart.dev/guides>

[15] JetBrains. (2024). *Kotlin documentation*. Kotlinlang.org.

<https://kotlinlang.org/docs/home.html>

[16] Oracle. (2024). The Java™ Tutorials. Oracle.

<https://docs.oracle.com/javase/tutorial/>

[17] Apple. (2024). Swift Programming Language. Swift.org.

<https://www.swift.org/documentation/>

[18] Google. (2024). Flutter Documentation. Flutter.dev.

<https://docs.flutter.dev>

[19] Meta. (2024). React Native Docs. Reactnative.dev.

<https://reactnative.dev/docs/getting-started>

[20] Ionic. (2024). What is Ionic?. Ionic Framework.

<https://ionicframework.com/why-ionic>

[21] Google. (2024). Angular Documentation. Angular.io.

<https://angular.io/docs>

[22] Vue.js. (2024). Introduction - Vue.js. Vuejs.org.

<https://vuejs.org/guide/introduction.html>

[23] Node.js. (2024). About Node.js. Node.js Foundation.

<https://nodejs.org/en/about>

[24] Python Software Foundation. (2024). Welcome to Python.org.

<https://www.python.org>

[25] Oracle. (2024). Java Overview. Oracle.

<https://www.oracle.com/java/technologies/javase-overview.html>

[26] Microsoft. (2024). C# documentation. Microsoft Learn.

<https://learn.microsoft.com/en-us/dotnet/csharp/>

[27] PHP Group. (2024). What is PHP? PHP.net.

<https://www.php.net/manual/en/intro-whatis.php>

[28] Express.js. (s.f.). Express - Node.js web application framework.

<https://expressjs.com/>

[29] Django Software Foundation. (s.f.). The Web framework for perfectionists with deadlines.

<https://www.djangoproject.com/>

[30] Pivotal Software, Inc. (s.f.). Spring Boot Reference Documentation.

<https://spring.io/projects/spring-boot>

[31] Taylor Otwell. (s.f.). Laravel - The PHP Framework For Web Artisans.

<https://laravel.com/>

[32] Microsoft Learn. (s. f.). What is an API?

<https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design>

[33] Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures (Doctoral dissertation, University of California, Irvine).

<https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>

[34] Facebook. (2015). GraphQL: A data query language.

<https://engineering.fb.com/2015/09/14/core-infra/graphql-a-data-query-language/>

[35] gRPC Authors. (2016). gRPC on HTTP/2: Engineering a Robust, High-performance Protocol.

<https://grpc.io/blog/grpc-on-http2/>

[36] SmartBear. (s. f.). What is Postman? Postman Learning Center.

<https://learning.postman.com/docs/getting-started/introduction/>

[37] OpenAPI Initiative. (s. f.). OpenAPI Specification. The Linux Foundation.

<https://www.openapis.org/>

[38] ECMA International. (2017). ECMA-404 The JSON Data Interchange Standard.

<https://www.ecma-international.org/publications-and-standards/standards/ecma-404/>

[39] Bray, T. (Ed.). (2008). Extensible Markup Language (XML) 1.0 (Fifth Edition). W3C.

<https://www.w3.org/TR/xml/>

Anexo I. Enlaces

Código del proyecto:

Enlace a la carpeta con el código de la API:

https://drive.google.com/file/d/1JVQhOJMDqrQ1Wmst0WJEWnIcN3YTprOF/view?usp=drive_link

Enlace a la carpeta con el código de la aplicación:

FamSync-main.zip

Apk de la aplicación:

<https://drive.google.com/file/d/10XvZGEM4dILIF4HJlphhTX3Fw0MpvNLA/view?usp=sharing>

Vídeo DEMO

<https://youtu.be/h8I9bihaDTA?si=Cb34n6uKr4t31m2s>