Logo_Cam_Rojo

CONSEJERÍA DE EDUCACIÓN

Comunidad de Madrid

**IES ENRIQUE TIERNO GALVÁN**

**Parla**

**CFGS DESARROLLO DE APLICACIONES MULTIPLATAFORMA**

**Curso 2024/2025**

**Proyecto DAM**

***TITULO: Aplicación en Flutter, EVENTIFY***

***Alumno*: David Montero Cuesta**

***Tutor*: Julián Parra Perales**

Junio de 2025

Tabla de contenido

[**DESCRIPCIÓN GENERAL DEL PROYECTO** 3](#_Toc197889596)

[**Comparativa de Eventify con otras aplicaciones en el mercado** 4](#_Toc197889597)

[Competencias en el mercado 4](#_Toc197889598)

[**Funcionalidades del Calendario en iOS (Apple Calendar)** 4](#_Toc197889599)

[**Funcionalidades del Calendario en Android (Google Calendar)** 5](#_Toc197889600)

[**Comparación: Apple Calendar vs. Google Calendar vs. Eventify** 8](#_Toc197889601)

[**Análisis Comparativo** 9](#_Toc197889602)

[**ANÁLISIS Y DISEÑO** 10](#_Toc197889603)

[**Requisitos** 10](#_Toc197889604)

[**FLUTTER VS KOTLIN MULTIPLATFORM** 12](#_Toc197889605)

[**DECISIÓN** 13](#_Toc197889606)

[**Elección de base de datos para Eventify** 15](#_Toc197889607)

[**Objetivo** 15](#_Toc197889608)

[**Comparativa de bases de datos en la nube gratuita** 15](#_Toc197889609)

[**Decisión: Firebase Firestore** 16](#_Toc197889610)

[**DISEÑO DE LA BASE DE DATOS** 17](#_Toc197889611)

[**ASPECTOS FUNCIONALES Y DISEÑO DE LA APLICACIÓN** 18](#_Toc197889612)

[**IMPLEMENTACIÓN DE LA APLICACIÓN** 19](#_Toc197889613)

[**PAQUETIZACIÓN** 19](#_Toc197889614)

[**INTERFAZ DEL USUARIO** 21](#_Toc197889615)

[**DISEÑO DE LAS INTERFACES** 21](#_Toc197889616)

[**NAVEGACIÓN** 23](#_Toc197889617)

[**ASPECTOS DESTACABLES DEL DESARROLLO** 24](#_Toc197889618)

[**LÍNEAS DE TRABAJO FUTURAS** 24](#_Toc197889619)

[**CONCLUSIONES** 25](#_Toc197889620)

[**BIBLIOGRAFÍA** 26](#_Toc197889621)

[**ANEXOS** 28](#_Toc197889622)

[***ENLACE AL GITHUB DEL PROYECTO*** 28](#_Toc197889623)

# 

# **DESCRIPCIÓN GENERAL DEL PROYECTO**

Eventify es una agenda multiplataforma desarrollada con el framework Flutter y el lenguaje Dart, diseñada para proporcionar a los usuarios una herramienta intuitiva, potente y adaptable para la gestión de su tiempo y eventos. Eventify está disponible para Android, iOS, macOS, GNU/Linux y Windows, ofreciendo una experiencia unificada, fluida y asistida por un agente de inteligencia artificial integrado, independientemente del dispositivo que utilice el usuario.

La aplicación destaca por su interfaz interactiva y de fácil uso, lo que minimiza la curva de aprendizaje. Un aspecto clave es la personalización visual, permitiendo a los usuarios modificar la apariencia de la aplicación mediante la selección de colores dinámicos según sus preferencias. La eficiencia y la rapidez de respuesta son fundamentales para garantizar una experiencia sin interrupciones en todos los dispositivos compatibles.

Eventify incorpora funcionalidades avanzadas de gestión de eventos, como la posibilidad de añadir, modificar y eliminar eventos en cualquier fecha del calendario. Cuenta con un sistema de búsqueda avanzada por palabras clave, nivel de prioridad y tipo de evento, además de filtros inteligentes según el contexto: por ejemplo, filtrar por localidad en conferencias o por asignatura en exámenes.

Un elemento diferenciador clave es la integración de un agente de inteligencia artificial, siempre disponible para asistir al usuario con tareas como la sugerencia de recordatorios optimizados, la organización del calendario o la localización rápida de eventos relevantes.

Para evitar que se pasen por alto eventos importantes, Eventify ofrece un sistema de notificaciones configurables, que envían recordatorios con antelación según las preferencias del usuario. La persistencia de datos está asegurada mediante el almacenamiento en la nube, lo que permite la sincronización entre dispositivos y abre la puerta a futuras funcionalidades más inteligentes basadas en el análisis de patrones de uso.

**En resumen**, Eventify se presenta como una solución de calendario **multiplataforma completa** que redefine la organización personal al integrar un **agente de inteligencia artificial proactivo**, una experiencia **personalizable** y un diseño centrado en la **eficiencia, sincronización, escalabilidad y seguridad**, ahora disponible no solo en dispositivos móviles, sino también en **sistemas de escritorio modernos**.

## **Comparativa de Eventify con otras aplicaciones en el mercado**

### Competencias en el mercado

Los calendarios de **iOS (Apple Calendar)** y **Android (Google Calendar)** ofrecen una variedad de funcionalidades diseñadas para la gestión de eventos, recordatorios y planificación.

### **Funcionalidades del Calendario en iOS (Apple Calendar)**

Gestión de eventos y recordatorios:

* Creación, edición y eliminación de eventos.
* Posibilidad de establecer eventos repetitivos (diarios, semanales, mensuales, anuales o personalizados).
* Agregar una ubicación al evento con integración de Apple Maps.
* Añadir notas y descripciones dentro del evento.
* Adjuntar archivos y enlaces a eventos.

Notificaciones y alertas:

* Recordatorios antes del evento con opciones personalizadas (minutos, horas o días antes).
* Alertas basadas en ubicación (por ejemplo, recibir un aviso al llegar a un lugar).
* Integración con Siri para establecer recordatorios con comandos de voz.

Sincronización y compatibilidad:

* Sincronización con iCloud, lo que permite acceder a eventos desde todos los dispositivos Apple.
* Compatibilidad con calendarios de terceros: Google Calendar, Microsoft Outlook, Exchange, Yahoo Calendar, CalDAV.
* Integración con aplicaciones de terceros para importar eventos.

Vista y organización:

* Diferentes modos de vista: día, semana, mes y año.
* Posibilidad de crear varios calendarios y asignarles colores diferentes para diferenciarlos.
* Opción de compartir calendarios con otros usuarios de iCloud.

Otras características:

* Modo oscuro compatible con iOS.
* Posibilidad de agregar eventos a través de Siri.
* Integración con Apple Watch para ver eventos en la muñeca.
* Posibilidad de responder a invitaciones de eventos desde la app.

### **Funcionalidades del Calendario en Android (Google Calendar)**

Gestión de eventos y tareas:

* Creación y edición de eventos con opciones de personalización.
* Eventos recurrentes con patrones flexibles de repetición.
* Posibilidad de adjuntar archivos desde Google Drive.
* Agregar descripciones y enlaces dentro del evento.
* Creación de tareas (integrado con Google Tasks).

Notificaciones y alertas:

* Recordatorios programables con distintas opciones de tiempo.
* Notificaciones basadas en ubicación.
* Posibilidad de recibir notificaciones por correo electrónico.

Sincronización y compatibilidad:

* Sincronización con Google Drive, Gmail y Google Tasks.
* Compatibilidad con calendarios de Microsoft Outlook, Exchange, iCloud, CalDAV.
* Integración con asistentes de voz como Google Assistant para programar eventos por voz.

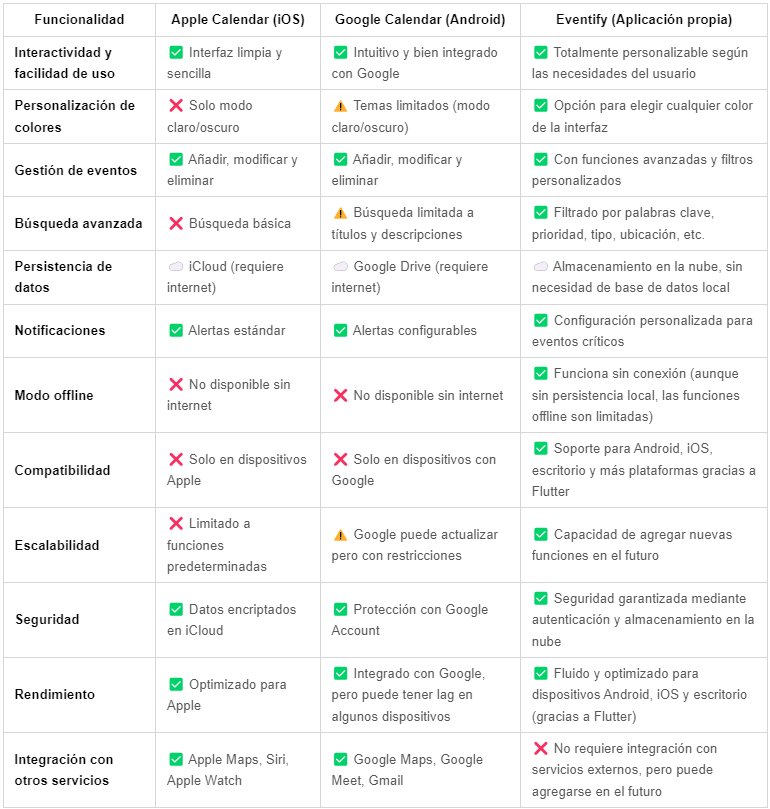
Vista y organización:

* Diferentes modos de vista: día, semana, mes y agenda.
* Uso de colores personalizables para organizar eventos por categoría.
* Posibilidad de superponer varios calendarios.
* Sincronización con eventos de Gmail (reservas de vuelos, reuniones, envíos, etc.).

Otras características:

* Modos de sugerencias automáticas, que completan eventos basados en patrones previos.
* Posibilidad de crear reuniones con enlaces directos de Google Meet.
* Creación de encuestas de disponibilidad con Google Workspace.
* Accesibilidad desde la web en cualquier dispositivo.
* Integración con Wear OS para ver eventos desde un smartwatch.

## **Comparación: Apple Calendar vs. Google Calendar vs. Eventify**



### **Análisis Comparativo**

#### Ventajas de la aplicación frente a Apple Calendar y Google Calendar

**Mayor personalización**: Apple Calendar y Google Calendar no permiten cambios en los colores de la interfaz, mientras que la aplicación de agenda ofrece esta opción de personalización.

**Búsqueda avanzada**: La aplicación permite filtrar eventos por múltiples criterios como palabras clave, prioridad, tipo, ubicación, etc., a diferencia de Apple y Google Calendar, que tienen opciones más limitadas.

**Escalabilidad**: La aplicación tiene la capacidad de agregar nuevas funciones en el futuro, permitiendo una mayor personalización y expansión de sus características.

# **ANÁLISIS Y DISEÑO**

### **Requisitos**

#### **Necesidades del Usuario**

* La aplicación debe ser **interactiva y fácil de usar**, con una interfaz intuitiva y sin curva de aprendizaje elevada.
* El usuario podrá **personalizar** el aspecto de la app, eligiendo colores que cambiarán dinámicamente la apariencia.
* El sistema debe ser **eficiente y rápido**, sin retrasos al interactuar.
* La aplicación debe ser accesible desde dispositivos móviles Android e iOS, así como desde ordenadores de escritorio. Su diseño multiplataforma garantiza una experiencia consistente y optimizada en todos los entornos.

#### **Requisitos Funcionales**

1. **Gestión de eventos**
   * El usuario podrá **añadir, modificar y eliminar** eventos en cualquier fecha del calendario.
2. **Búsqueda avanzada**
   * Se podrán buscar eventos por:
   * Palabras clave
   * Nivel de prioridad
   * Tipo de evento
   * Otros filtros específicos, como:
     + Localidad (para conferencias)
     + Asignatura (para exámenes)
3. **Notificaciones**
   * Los eventos importantes podrán configurarse para **enviar recordatorios** antes de su fecha.
4. **Persistencia de datos**
   * Los eventos se almacenarán de forma **persistente** en una base de datos en la nube.
5. **Modo offline**
   * Algunas funciones estarán disponibles sin conexión a internet
6. **Soporte multiplataforma**
   * La aplicación funcionará correctamente en Android, iOS, Windows, Linux y macOS

#### **Requisitos No Funcionales**

1. **Rendimiento**
   * La aplicación deberá **responder rápidamente** y mantener una experiencia fluida.
2. **Rendimiento**
   * **La aplicación deberá ofrecer tiempos de respuesta inferiores a 1 segundo en acciones comunes.**
3. **Escalabilidad**
   * El sistema deberá poder incorporar nuevas funcionalidades sin necesidad de reestructurar su arquitectura.
4. **Compatibilidad**
   * La app deberá ejecutarse de manera fluida y coherente en diferentes dispositivos y sistemas operativos.
5. **Escalabilidad**
   * El sistema debe permitir añadir nuevas funcionalidades en el futuro, +como:
     + Sincronización con cuenta
     + Exportación a otros formatos
     + Widgets de escritorio
6. **Personalización**
   * El usuario podrá **cambiar el tema de la app** mediante una paleta amplia de colores.
7. **Portabilidad**
   * La app podrá ser trasladada a otros entornos o sistemas con ajustes mínimos gracias a flutter.

## **FLUTTER VS KOTLIN MULTIPLATFORM**

Flutter y Kotlin Multiplatform son dos soluciones potentes para el desarrollo de aplicaciones multiplataforma. Mientras Flutter ofrece una única base de código con una interfaz coherente en todas las plataformas, Kotlin Multiplatform destaca por su interoperabilidad nativa y reutilización de lógica de negocio. La elección entre ambos depende del enfoque deseado en diseño, desarrollo y escalabilidad.



### **DECISIÓN**

Dado que el objetivo es desarrollar una aplicación de agenda disponible en **Android, iOS y escritorio**, **Flutter** representa la mejor opción por las siguientes razones:

**1️. Un solo código para todas las plataformas**

Flutter permite escribir **una única base de código**, evitando la necesidad de desarrollar interfaces separadas para Android, iOS y escritorio. En contraste, **Kotlin Multiplatform** requiere escribir la lógica compartida, pero las interfaces deben desarrollarse por separado en cada plataforma (Jetpack Compose para Android, SwiftUI para iOS, etc.).

**2. Interfaz personalizada y flexible**

Flutter ofrece un control total sobre la apariencia de la aplicación, permitiendo diseñar una **UI atractiva y homogénea** en todas las plataformas sin depender de los componentes nativos. Esto resulta ideal para una aplicación de agenda, donde la experiencia de usuario y la usabilidad son aspectos fundamentales.

**3️. Desarrollo rápido con Hot Reload**

El **Hot Reload** de Flutter permite visualizar los cambios en tiempo real sin necesidad de recompilar la aplicación por completo. Esto agiliza el proceso de desarrollo en comparación con **Kotlin Multiplatform**, donde las pruebas pueden ser más lentas debido a la necesidad de compilar código nativo para cada plataforma.

**4️. Soporte para escritorio y web**

Flutter ofrece un soporte más avanzado para **aplicaciones de escritorio y web**, facilitando la expansión de la aplicación de agenda a otras plataformas en el futuro. Por otro lado, **Kotlin Multiplatform** todavía presenta limitaciones en este aspecto, especialmente en la parte de la UI.

**5️. Amplia comunidad y paquetes listos para usar**

Flutter cuenta con una comunidad extensa y un **ecosistema de paquetes bien desarrollado**, lo que facilita la integración de funcionalidades clave como bases de datos, notificaciones, sincronización en la nube y autenticación. Esto reduce el tiempo de desarrollo en comparación con **Kotlin Multiplatform**, donde algunos recursos aún son limitados.

Para desarrollar una **aplicación multiplataforma eficiente, con una interfaz moderna y un desarrollo ágil**, **Flutter se presenta como la mejor opción**. Gracias a su capacidad para compartir código entre todas las plataformas y a su amplio ecosistema de herramientas, permite optimizar el tiempo de desarrollo y garantizar una **experiencia de usuario consistente**.

## **Elección de base de datos para Eventify**

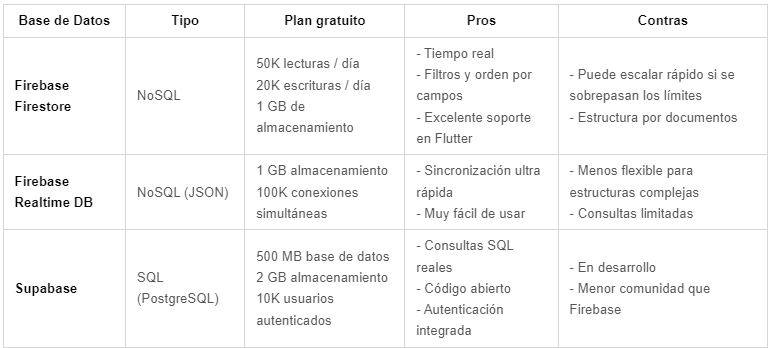
### **Objetivo**

Para el desarrollo de Eventify, se necesita una base de datos que:

* Permita guardar eventos personalizados (fecha, título, descripción, etc.).
* Sea accesible desde varios dispositivos.
* Soporte usuarios distintos y gestión de sus eventos.
* Sincronice cambios en tiempo real o al menos de forma eficiente.
* Tenga una opción gratuita suficiente para el uso inicial de la app.

Por ello, se ha decidido utilizar una base de datos en la nube, ya que así los datos se almacenan online y no se pierden, incluso si el usuario cambia de dispositivo o reinstala la app.

### **Comparativa de bases de datos en la nube gratuita**



### **Decisión: Firebase Firestore**

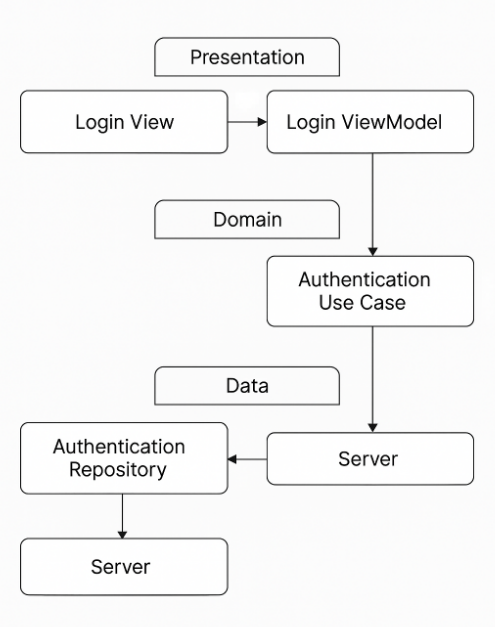
Se ha decidido usar Firebase Firestore como base de datos para Eventify por las siguientes razones:

* Su plan gratuito es suficiente para empezar, con 50.000 lecturas y 20.000 escrituras al día.
* Permite almacenar los eventos por usuario y ordenarlos por fecha fácilmente.
* Soporta autenticación, por lo que cada usuario tendrá su propia agenda.
* Se sincroniza en tiempo real entre dispositivos.
* Está muy bien integrada con Flutter y tiene documentación abundante.

# **DISEÑO DE LA BASE DE DATOS**

# **ASPECTOS FUNCIONALES Y DISEÑO DE LA APLICACIÓN**

En Eventify, para la pantalla de login, se ha utilizado la arquitectura Clean combinada con el patrón Model-View-ViewModel (MVVM).

****

# **IMPLEMENTACIÓN DE LA APLICACIÓN**

## **PAQUETIZACIÓN**

La estructura de código de la aplicación Eventify, ubicada en la carpeta lib, se organiza en varios paquetes y ficheros clave. Para la gestión de la autenticación de usuarios, incluyendo las funcionalidades de inicio y registro de sesión, se ha creado el paquete auth. El paquete common alberga código reutilizable y utilidades compartidas por diferentes partes de la aplicación. La gestión de dependencias e inyección se realiza a través del paquete di. Por último, el paquete l10n está dedicado a la internacionalización y localización, conteniendo los recursos para las traducciones de la aplicación.

Además de estos paquetes, se encuentran dos ficheros principales en la raíz de lib. firebase\_options.dart contiene la configuración necesaria para establecer la conexión con la base de datos Firebase. El fichero main.dart sirve como punto de entrada de la aplicación, encargándose de su inicialización y el arranque de la interfaz de usuario.

Contenido de auth (implementada arquitectura clean y MVVM):

* data
  + data\_sources
  + models
  + repositories
* domain
  + entities
  + presentation
    - screen
    - view\_model
  + repositories
  + use\_cases

Contenido de common:

* theme
  + colors
* utils
  + auth
  + dates
  + errors
* widgets
  + auth
    - animations
    - widgets
  + calendar
    - animations
    - widgets
  + chat
  + Profile

# **INTERFAZ DEL USUARIO**

## **DISEÑO DE LAS INTERFACES**

**CALENDARIO ANUAL CALENDARIO MENSUAL CREACIÓN DE EVENTO**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

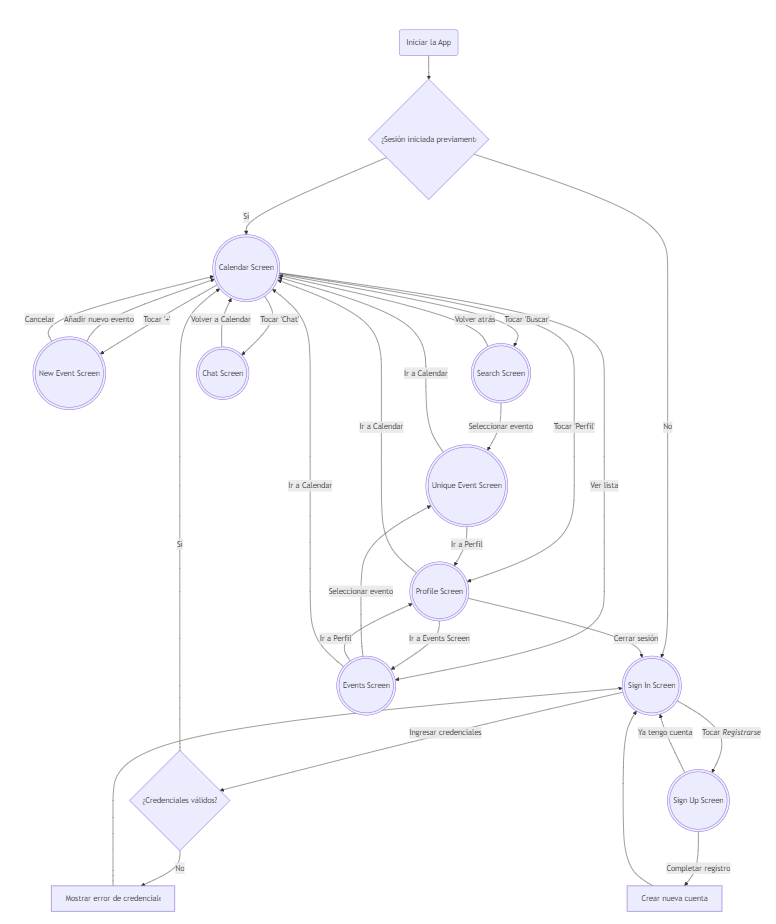
**INICIO DE SESIÓN REGISTRO**

|  |  |
| --- | --- |
|  |  |

**PERFIL CHAT CON EVENTIFY**

|  |  |
| --- | --- |
|  |  |

## **NAVEGACIÓN**



# **ASPECTOS DESTACABLES DEL DESARROLLO**

**Investigación sobre Mistral (Fase Inicial):** La exploración de la IA con Mistral presentó complicaciones iniciales debido a la novedad de la tecnología y la necesidad de comprender sus fundamentos.

**Adaptación a Firebase (Fase de Backend):** La transición a Firebase implicó una complejidad significativa al requerir un cambio de mentalidad respecto a la gestión y estructura de los datos.

**Problemas de Traducción (Fase de Localización):** La implementación de la localización con l10n en Flutter generó diversas complicaciones relacionadas con la configuración, la gestión de los archivos de traducción y la correcta visualización en la aplicación.

# **LÍNEAS DE TRABAJO FUTURAS**

En las próximas fases del proyecto, se tiene prevista una serie de líneas de trabajo fundamentales para la evolución y optimización de la aplicación. Una de las principales áreas de enfoque será la conexión completa con la base de datos, lo que permitirá integrar todas las funcionalidades necesarias para una gestión eficaz de los datos mediante Firebase. Esto incluirá la implementación de operaciones de lectura, escritura y actualización de datos de manera fluida y segura.

Asimismo, se continuará con la adopción de una arquitectura Clean y el patrón MVVM en todo el proyecto. Esta extensión a todas las capas y módulos del sistema mejorará significativamente la organización del código, su testabilidad y mantenibilidad, garantizando que el proyecto sea escalable y fácil de modificar en el futuro.

Para mejorar la experiencia de usuario desde el primer momento, se desarrollará una **pantalla de carga inicial** que se mostrará al inicio de la aplicación. Esta pantalla proporcionará una transición más suave entre el arranque de la aplicación y la interacción con el usuario, asegurando que el proceso de inicialización sea lo menos invasivo posible.

En cuanto a la interfaz de usuario y la lógica de negocio, se trabajará en la creación de pantallas clave, como la de búsqueda de eventos, la de creación de nuevos eventos y la de visualización de la lista de eventos. Cada una de estas pantallas contará con su propia funcionalidad y diseño, optimizados para ofrecer una experiencia de usuario intuitiva y fluida.

Finalmente, se planea la integración del agente de inteligencia artificial dentro de la aplicación. Este agente proporcionará asistencia inteligente y proactiva, mejorando la interacción del usuario con la aplicación y ofreciendo recomendaciones personalizadas según sus necesidades y preferencias.

# **CONCLUSIONES**

El proyecto Eventify es una iniciativa en curso para crear una agenda móvil multiplataforma intuitiva y potente, desarrollada con Flutter y Dart para su despliegue en Android e iOS. La base de datos en la nube de Firebase asegura la persistencia de los datos y sienta las bases para la futura escalabilidad de la aplicación.

Si bien la aplicación ofrece funcionalidades fundamentales de gestión y búsqueda de eventos, características distintivas como la asistencia inteligente (similar a la de un agente de IA) y la amplia gama de opciones de personalización de colores aún se encuentran en fase de implementación. La visión del proyecto es ofrecer una experiencia de usuario enriquecida con capacidades proactivas y una interfaz altamente adaptable a las preferencias individuales.

La estructura de la aplicación, con una adopción inicial de principios de arquitectura limpia y MVVM, busca facilitar la futura expansión y el mantenimiento del código. En resumen, Eventify, actualmente en desarrollo, se perfila como una solución de agenda móvil con un potencial significativo, combinando la eficiencia de Flutter con la promesa de funcionalidades avanzadas de asistencia y personalización que la diferenciarán en el mercado.

El proyecto Eventify representa una emocionante aventura en el desarrollo de una agenda móvil multiplataforma, construida con Flutter y Dart para alcanzar usuarios en Android e iOS. La elección de Firebase como backend proporciona una base sólida para la persistencia de datos y la escalabilidad futura.

El desarrollo de Eventify, aunque actualmente enfocado en funcionalidades esenciales de gestión y búsqueda de eventos, es un proceso que encuentro profundamente gratificante y que siento que está impulsando significativamente mi crecimiento como desarrollador. La perspectiva de implementar características distintivas como la asistencia inteligente y una amplia paleta de personalización visual, actualmente en desarrollo, me entusiasma enormemente.

Anticipo que la culminación de este proyecto, con su combinación de una interfaz de usuario intuitiva creada con Flutter y la promesa de funcionalidades avanzadas, me brindará una satisfacción increíble al ver cómo una herramienta tan completa y personalizada llega a los usuarios. Eventify no es solo una aplicación; es un reflejo de mi progreso y dedicación como desarrollador.

# **BIBLIOGRAFÍA**

[OpenAI](https://chat.openai.com)**.** ChatGPT. Recuperado en abril de 2025. Utilizado como asistente virtual para generar contenido, resolver dudas técnicas y aclarar conceptos de programación.

[Google](https://gemini.google.com)**.** Gemini de Google. Recuperado en abril de 2025. Empleado para contrastar información y obtener diferentes perspectivas sobre temas de desarrollo.

[YouTube](https://youtu.be/3_dwKqCsbis?si=NPt1LLq2j8XMXkV1)**.** Desarrollo de Aplicaciones moviles en 2024. Recuperado en abril de 2025. Este video proporciona una guía completa para aspirantes a desarrolladores de aplicaciones móviles en 2024, cubriendo el conocimiento esencial para comenzar en este campo. Explica las dos principales metodologías de desarrollo: nativo y multiplataforma. Para el desarrollo nativo, se mencionan Objective-C y Swift para iOS, y Java y Kotlin para Android. En cuanto al desarrollo multiplataforma, se discuten frameworks como React Native, Flutter, Kotlin Multiplatform y .NET MAUI. El video también ofrece una hoja de ruta para el aprendizaje, incluyendo el aprendizaje de un lenguaje de programación base, un framework/SDK, Git y GitHub, y servicios backend.

[YouTube](https://youtu.be/3_dwKqCsbis?si=lBsnUmf4qzd5lxjO)**.** Desarrollo de Aplicaciones moviles en 2024. Recuperado en abril de 2025. Utilizado como referencia para analizar diferentes variantes de estructuras de proyecto y tecnologías como Flutter y Kotlin Multiplatform, con el objetivo de tomar decisiones informadas durante el desarrollo de la aplicación.

[YouTube](https://youtu.be/FYwkAdffIKo?si=UM3Hp6vGFZzxJ7E6)**.** Flutter vs Kotlin Multiplatform ¿Por qué Google invierte en dos tecnologías enfrentadas?. Recuperado en abril de 2025. Se ha tenido en cuenta la opinión del creador del video y también las reflexiones aportadas por los usuarios en la sección de comentarios, como parte del análisis sobre la viabilidad y enfoque del uso de Flutter en el desarrollo del proyecto.

[YouTube](https://youtu.be/pw1AUMs6o8s?si=Ym-t7fQEhdCrkiEN)**.** Flutter vs Kotlin Multiplatform: Google se posiciona. Recuperado en abril de 2025. Se analiza la estrategia de Google al invertir en dos tecnologías aparentemente competidoras para el desarrollo de aplicaciones móviles: Flutter y Kotlin Multiplatform. El video explora los posibles enfoques y el posicionamiento de Google frente a estas dos opciones, considerando sus fortalezas y debilidades en el ecosistema del desarrollo móvil.

[FlutterFlow](https://app.flutterflow.io)**.** FlutterFlow - UI Builder para Flutter. Recuperado en abril de 2025. Se utilizará como herramienta principal para el diseño visual de la interfaz de usuario del proyecto. Gracias a su editor gráfico tipo drag and drop, permite construir pantallas de manera intuitiva, similar al sistema de diseño de vistas de Android Studio. Esta plataforma acelera el proceso de prototipado y desarrollo frontend, facilitando también la exportación del código Flutter resultante para integrarlo con la lógica personalizada del proyecto.

[DartPackages](https://pub.dev/packages/firebase_core/install)**.** Firebase Core - Instalación de Firebase en Flutter. Recuperado en abril de 2025. Se ha utilizado este recurso para instalar y configurar **Firebase Core**, la biblioteca base necesaria para la integración de Firebase en Flutter. La documentación oficial ha servido de guía para la correcta implementación de los servicios de Firebase dentro del proyecto, asegurando compatibilidad y optimización en el entorno de desarrollo.

[YouTube](https://youtu.be/s-1vO0nDpzE?si=DwUnzPco5QopWUa3)**.** Tutorial Crear Agentes AI Mistral - Crea tus propios agentes de IA con Mistral. Recuperado en abril de 2025. Utilizado como referencia para la creación y comprensión del funcionamiento del agente de IA basado en Mistral.

[YouTube](https://youtu.be/LjO2aXUZeWo?si=Lafzt1gHrKhbyAdo)**.** Traducciones y localización en Flutter con l10n | Internacionaliza tu app. Recuperado en abril de 2025. Utilizado como referencia para comprender el proceso de internacionalización y localización de aplicaciones desarrolladas con Flutter, específicamente mediante el uso de la herramienta l10n.

# **ANEXOS**

[***ENLACE AL GITHUB DEL PROYECTO***](https://github.com/DavidMonteroCuesta/TFG-DAM_EVENTIFY.git)