LOCMABAR

ÍNDICE

[Descripción del Proyecto 4](#_Toc199185965)

[Justificación del Proyecto 4](#_Toc199185966)

[Análisis de demanda(estado del arte empresarial) 4](#_Toc199185967)

[Estimación de la demanda 5](#_Toc199185968)

[Estado del mercado, empresas 5](#_Toc199185969)

[Viabilidad del Proyecto(DAFO) 6](#_Toc199185970)

[D(debilidades): 6](#_Toc199185971)

[A(amenazas): 6](#_Toc199185972)

[F(fortalezas): 7](#_Toc199185973)

[O(oportunidades): 7](#_Toc199185974)

[Estado del arte tecnológico y normativa 8](#_Toc199185975)

[Tecnologías disponibles 8](#_Toc199185976)

[Normativa que aplica 9](#_Toc199185977)

[Justificación de tecnologías elegidas 11](#_Toc199185978)

[Solución Técnica 12](#_Toc199185979)

[Entornos de desarrollo y tecnología (arquitectura, patrón de diseño elegido) 12](#_Toc199185980)

[Diagrama de Componentes 12](#_Toc199185981)

[Análisis (requisitos funcionales y no funcionales, casos de uso) 14](#_Toc199185982)

[Requisitos funcionales 14](#_Toc199185983)

[Requisitos no funcionales 14](#_Toc199185984)

[Casos de uso 18](#_Toc199185985)

[Diseño e implementación de la persistencia de los datos 31](#_Toc199185986)

[Diagramas: clases y diagrama de flujo 32](#_Toc199185987)

[Diagrama de clases 32](#_Toc199185988)

[Diagramas de Flujo 33](#_Toc199185989)

[Interfaces (desarrollados o mocks) 37](#_Toc199185990)

[Planes de pruebas 38](#_Toc199185991)

[Metodología y estimación de costes 40](#_Toc199185992)

[Diagrama de Gantt 40](#_Toc199185993)

[SW, HW, horas, amortización 40](#_Toc199185994)

[Otros detalles de planificación 42](#_Toc199185995)

[Seguimiento y control del desarrollo 42](#_Toc199185996)

[Aplicación desarrollada 42](#_Toc199185997)

[Manuales 42](#_Toc199185998)

[i. Manual de usuario 42](#_Toc199185999)

[ii. Manual del desarrollador 44](#_Toc199186000)

[iii. Manual de administrador 45](#_Toc199186001)

[Archivo/s de la aplicación. 46](#_Toc199186002)

[Conclusiones 48](#_Toc199186003)

[Propuestas de mejora 49](#_Toc199186004)

[Bibliografía 51](#_Toc199186005)

[Anexos 52](#_Toc199186006)

[Presentación PowerPoint 52](#_Toc199186007)

[Video de presentación aplicación 52](#_Toc199186008)

# Descripción del Proyecto

LocMaBar es una aplicación móvil que ha sido desarrollada en Kotlin con Android Studio facilitando el descubrimiento de bares y restaurantes locales. La app permite a los usuarios consultar establecimientos, ver información detallada, dejar valoraciones y proponer nuevos locales. Para garantizar una experiencia segura, los comentarios e imágenes pasan por un filtro de contenido y validación. LocMaBar busca ofrecer una plataforma de fácil acceso y fiable para mejorar la visibilidad de los negocios locales y la experiencia de los usuarios, integrando geolocalización del usuario y que se hagan recomendaciones en base a esta localización del usuario.

# Justificación del Proyecto

LocMaBar surge como una solución para facilitar el descubrimiento de bares y restaurantes locales, especialmente en pequeños municipios donde las opciones no siempre son visibles en plataformas generalistas. Para justificar la viabilidad del proyecto, hemos analizado la demanda de aplicaciones de este tipo en el mercado español y el estado actual del sector, identificando tanto las necesidades de los usuarios como las oportunidades que LocMaBar puede aprovechar frente a la competencia.

## Análisis de demanda (estado del arte empresarial)

Para demostrar la viabilidad de LocMaBar en el mercado español, se ha realizado un análisis de demanda basándose en el comportamiento que tienen los usuarios y en las tendencias de la actualidad en el uso de apps móviles para encontrar bares y restaurantes. El análisis se centra en identificar las necesidades del usuario, los términos o ideas que buscan relacionados con una app como LocMaBar.

En España, el sector de la hostelería, y los bares en particular, son uno de los pilares culturales, sociales y económicos. Por ejemplo, de acuerdo con los datos corporativos generales sobre el sector, en España hay más de 300,000 empresas de la hostelería. Los ciudadanos españoles salen a diario a estos bares por cuestiones sociales, lugares para tomar algo o por eventos. Esta alta cantidad de bares genera una necesidad constante de encontrar opciones específicas según la ubicación o gustos del usuario.

El comportamiento de los usuarios en el contexto español evidencia patrones claros en la búsqueda de bares y restaurantes. En primer lugar, se identifica una práctica habitual de búsqueda por ubicación específica, tanto en provincias como en municipios concretos. En segundo lugar, la búsqueda mediante geolocalización ha emergido como una funcionalidad indispensable, especialmente para usuarios que requieren opciones inmediatas y en lugares que desconocen. Finalmente, se observa un creciente interés por aplicaciones especializadas en nichos específicos, como los bares ya sea para ir a tomar algo o para ir a comer algo.

Además, se ha detectado un creciente interés por aplicaciones especializadas en nichos concretos, como los bares. Aunque plataformas como Google Maps o TripAdvisor permiten buscar establecimientos de todo tipo, los usuarios valoran cada vez más herramientas específicas que simplifiquen la búsqueda y proporcionen información detallada y relevante, como tipo de bar, horario o ambiente.

## Estimación de la demanda

La estimación de la demanda se basa en un estudio realizado con 20 usuarios mediante [Google Forms](https://forms.gle/ac7rrnMSsVLF4w9i8). Este estudio empleó un cuestionario para evaluar hábitos de búsqueda y preferencias, con una muestra representativa de turistas (30%), residentes rurales (40%) y jóvenes de la ciudad de 18 a 35 años (30%).

Búsqueda mediante geolocalización: Se proyecta una demanda alta, en el estudio el 80% de los participantes (16 personas) indicó que usaría una aplicación que muestre bares y restaurantes cercanos de forma inmediata mediante geolocalización. Este interés fue especialmente alto entre jóvenes urbanos (100% de este grupo), luego residentes rurales (75%) y por último, turistas (67%).

Búsqueda por ubicación específica: Se estima una demanda moderada, en el estudio el 60% de los encuestados mostró interés en una aplicación que facilite la búsqueda de bares y restaurantes en localidades específicas, como provincias o municipios concretos. Este interés fue más pronunciado entre turistas (83%) y residentes de municipios (62.5%), quienes destacaron la dificultad de encontrar opciones relevantes en plataformas generalistas.

Interés en aplicaciones especializadas: En el estudio, el 50% de los encuestados votaron que es muy o algo importante que una aplicación se especialice solo en bares y restaurantes, valorando la disponibilidad de datos detallados sobre los locales. Este interés fue más notable entre turistas (67%) y jóvenes urbanos (50%), mientras que los residentes de municipios mostraron un interés menor (37.5%).

**Tendencias**

Tendencias específicas que existen en el sector hostelero en España:

-Crecimiento del uso de aplicaciones móviles: Los usuarios españoles usan el móvil con un promedio de más de 3h 22minutos al día con un aumento constante que va aumentando cada vez más.

-Digitalización del sector de la hostelería: el 74% de la hostelería en España realiza inversiones en la tecnología para mejorar la experiencia del cliente. Por ejemplo, el pago mediante tarjeta con el móvil es lo que más valoran los clientes ya que es una forma rápida de pagar en los bares y restaurantes.

-Sostenibilidad: según un estudio de la agencia Nielsen, el 66% de los consumidores están dispuestos a pagar más por marcas sostenibles, esto se puede observar en que el 62% de los restaurantes están realizando adaptaciones para reducir el uso de los plásticos.

## Estado del mercado, empresas

Se han identificado varias empresas y aplicaciones similares a LocMaBar, que operan en el sector de recomendaciones de bares y restaurantes. Entre ellas se encuentran:

**Google Maps**: Una de las aplicaciones más utilizadas mundialmente para buscar lugares de interés, incluyendo bares y restaurantes. Su principal ventaja es su integración con la navegación GPS y la gran cantidad de reseñas de usuarios, que al ser conocido genera más confianza en los usuarios. Sin embargo, su enfoque es muy amplio y no se especializa en municipios pequeños, lo que puede dificultar la visibilidad de bares menos conocidos.

**TripAdvisor**: Esta plataforma se centra en la recomendación de restaurantes y lugares turísticos, con un sistema de valoraciones detallado por los usuarios. Su alcance es internacional, pero su enfoque principal son destinos turísticos populares, dejando de lado muchas opciones en localidades menos conocidas. A diferencia de LocMaBar, está más dirigida a restaurantes formales y destinos turísticos populares, dejando de lado los bares en pequeños municipios.

**TheFork**: Es una plataforma especializada en reservas de restaurantes, con una fuerte presencia en España. En 2024, TheFork tenía más de 3 millones de usuarios activos mensuales y colaboraba con alrededor de 20,000 restaurantes en el país. Permite realizar reservas directamente desde la aplicación y ofrece promociones, lo que resulta atractivo para los usuarios. A diferencia de LocMaBar, su enfoque está limitado a restaurantes, excluyendo bares y apenas llega la aplicación a los municipios.

Al analizar estas aplicaciones existentes, se han identificado tendencias comunes en el mercado, como la creciente dependencia de las valoraciones de usuarios para la toma de decisiones y la integración de mapas interactivos con opciones de geolocalización.

## Viabilidad del Proyecto(DAFO)

### D(debilidades):

Falta de reconocimiento de la marca: Siendo LocMaBar una app nueva en el mercado, parte desde cero en términos de visibilidad y confianza a los usuarios que ya tienen unas aplicaciones o métodos establecidos. Esto dificulta atraer usuarios a que descarguen LocMaBar sin una fuerte estrategia de marketing.

Dependencia de validación manual: El proceso de revisión manual de los comentarios, aunque asegura calidad y seguridad, puede ser lento y requerir de bastante tiempo por parte de los administradores, principalmente si se da que la app consigue una fuerte entrada en el mercado creciendo así y recibiendo muchos más comentarios y valoraciones.

Dependencia de la actividad de los usuarios: El crecimiento y éxito de la aplicación depende principalmente de que los usuarios usen la app activamente, y si al principio la aplicación no consigue un reducido grupo de usuarios activos puede ser poco útil.

### A(amenazas):

Competencia con apps ya consolidadas: Aplicaciones como Google Maps y TripAdvisor con millones de usuarios y una infraestructura fuerte y ya conocida, son un reto para LocMaBar. Dado que estas aplicaciones ya tienen reseñas y mapas detallados, lo que puede causar que los usuarios prefieran seguir usando lo ya conocido.

Riesgo en la monetización por publicidad: La dependencia inicial de Google AdMob para generar ingresos podría generar que los usuarios lo rechacen si los anuncios no se gestionana bien, afectando así la experiencia del usuario y con contenido molestoso.

Desinterés de los bares locales: Muchos bares o restaurantes, especialmente pequeños, ya sea porque les cuesta adaptarse a lo digital o porque si no perciben beneficio inmediato no les interesaría. Según un estudio sobre 240 mil bares, restaurantes y cafeterías, solo un 42,8% tiene página web, reforzando así esta amenaza a LocMaBar.

Dependencia de la tecnología: Problemas con firebase como fallos de conexión o límites gratuitos o con Google Location Services podrían afectar la funcionalidad principal de la aplicación, principalmente la geolocalización.

### F(fortalezas):

Cobertura de municipios menos explorados: LocMaBar brilla al proporcionar información detallada sobre bares y restaurantes en municipios más pequeños donde incluso Google Maps y TripAdvisor tienden a pasar por alto. Destaca negocios locales que de otro modo permanecerían ocultos.

Geolocalización intuitiva: Con la integración de geolocalización, los usuarios pueden encontrar opciones a su alrededor en cercanía con mínimo esfuerzo, un beneficio clave para turistas o cualquier persona que busque algo de manera espontánea, como café o tapas después del trabajo.

Control estricto de contenido: El sistema de validación para comentarios e imágenes (con filtrado automático y revisión manual adicional) garantiza la seguridad en el uso del servicio y la calidad del contenido, distinguiéndolo de aplicaciones donde la información no siempre está controlada.

Enfoque especializado: Debido a que LocMaBar se enfoca únicamente en bares y restaurantes, los usuarios obtienen una experiencia adaptada a un nicho, a diferencia de la que ofrecen plataformas generalistas, lo que puede llevar a una mayor retención entre los clientes.

Tecnología moderna y escalable: El uso de Kotlin, Firebase y Figma asegura una aplicación de buen rendimiento, segura y fácil de mantener, eliminando preocupaciones sobre el mantenimiento a medida que se agregan nuevas características y surge la posibilidad de expansión con nuevas funcionalidades.

### O(oportunidades):

Colaboraciones con bares locales: La implementación de carteles publicitarios por parte de las asociaciones de hostelería, o la inclusión de menús, horarios y promociones actualizadas de ciertos bares, podría ser ventajosa tanto para los negocios como para los usuarios, permitiendo atraer a más personas y poder establecer un ecosistema beneficioso.

Nuevas funcionalidades atractivas: Agregar nuevas funciones tales como reservas directas en los bares, alertas de ofertas, y un sistema de recomendaciones basados en preferencias como “bares con música en vivo” podría atraer más usuarios y optimizar el funcionamiento del sistema.

Expansión a ciudades grandes: Consolidarse en otros municipios es el enfoque principal, pero Madrid y Barcelona representan una gran oportunidad por su concentración de bares y su gran auge por el uso de la geolocalización, lo que garantizaría gran éxito en su utilización.

Monetización alternativa: La inclusión de AdMob no es la única alternativa. La creación de modelos como suscripciones premium de usuarios que deseen filtros avanzados o con los bares que deseen pagar por el anuncio de su local podrían diversificar aún más los ingresos y proveer estabilidad financiera.

Tendencias de sostenibilidad: Realzar el uso de productos locales en bares podría atraer a un público más amplio que se interese en el cuidado del medio ambiente.

Imágenes: La posibilidad de implementar imágenes tanto del local como de la comida dentro de este es una alternativa potencial que podría llamar aún mucho más la atención al usuario y hacer que realice una experiencia muchísimo más única dentro de la app.

# Estado del arte tecnológico y normativa

Para desarrollar LocMaBar, han sido elegidas tecnologías modernas y eficaces que cumplen con los requisitos de seguridad y escalabilidad necesarios para una aplicación móvil dedicada a encontrar bares y restaurantes. A continuación, se explica cada tecnología y su justificación en el proyecto.

## Tecnologías disponibles

**Base de Datos**

Para LocMaBar se ha utilizado Firebase Firestore, una base de datos en la nube en tiempo real de Google para guardar y gestionar la información. Los datos, la lista de bares, valoraciones y solicitudes de sitios nuevos, se mantienen al día para todos los usuarios por la sincronización a tiempo real a través de Firebase. Firestore puede administrar a una gran cantidad de usuarios además, también se conjuga de una forma fácil con otras herramientas como Storage y Authentication, simplificando así el desarrollo.

-**Configuración inicial**:

1.Crear un proyecto en firebase.

2.Habilitar el Firestore y seleccionar la región europe-west1 para mejor conexión en España.

3.Se añaden las dependencias de Firebase en el “build-gradle” del proyecto:

implementation("com.google.firebase:firebase-auth-ktx")

implementation("com.google.firebase:firebase-firestore-ktx")

4.Configurar las reglas para que solo puedan acceder los usuarios autenticados.

**Lenguaje de Programación y Entorno de Desarrollo**

El principal lenguaje que se ha utilizado en LocMaBar es Kotlin, que es el lenguaje de programación principal para crear aplicaciones en Android. Tiene una sintaxis sencilla y clara, que ayuda a reducir errores. Además, al ser compatible con Java, se pueden aprovechar bibliotecas ya existentes si es necesario. Kotlin también soporta las funciones más recientes de Android, que facilitan la gestión de operaciones asíncronas, como las consultas a Firestore.

Android Studio es el entorno de desarrollo integrado (IDE) oficial para desarrollar aplicaciones de Android, y ha sido utilizado como la herramienta principal para desarrollar LocMaBar. Proporciona un emulador para probar la aplicación en diferentes dispositivos, herramientas de depuración y soporte para integrar Firebase y otras bibliotecas.

**Autenticación y Servicios en la Nube**

>>Firebase Authentication: El servicio de Firebase Authentication se encarga de dar soporte y seguridad para el registro e inicio de sesión de los usuarios de LocMaBar. Actualmente, los únicos métodos de autenticación permitidos son el email y contraseña, es el que se ha utilizado en LocMaBar. Se consigue habilitando el método email/contraseña en Firebase y añadiendo su dependencia correspondiente que te la da Firebase.

>>Firebase Firestore: Se almacenan los datos de los bares y restaurantes relacionados con los establecimientos directamente en Firebase Firestore, garantizando un acceso rápido y seguro a los datos de los bares y restaurantes agregados en la aplicación. Es uno de los tantos servicios de Firebase.

>>Google Location Services: Es utilizado para recoger la ubicación del usuario, su latitud y longitud. Este servicio se apoya en datos de GPS, Wi-Fi y redes móviles del teléfono o dispositivo para dar una ubicación muy precisa, especialmente para encontrar bares o restaurantes cercanos mediante geolocalización. Para su funcionamiento se añade al trabajo su implementación que se puede encontrar en la página web buscando “Google Location Services” e implementando la lógica en el código.

>>Google Cloud Geocoding: Este servicio se encarga de convertir las direcciones físicas (como “Calle Paseo, 5, Iniesta, Cuenca”) en coordenadas geográficas(latitud y longitud), permitiendo así colocar los locales solicitados por los usuarios en su ubicación y soportar la búsqueda por gerolocalización. Para habilitarlo en el trabajo hay que crear una clave API en Google Cloud Console y luego añadir la dependencia que te da Google Cloud Console.

>>Figma: Con esta plataforma web se han diseñado los prototipos de la interfaz de usuario y del administrador. Dando gran facilidad para trabajar en equipo y exportar los diseños a código Kotlin a través de uno de los complementos que esta herramienta provee, Upspeed, dentro de la misma aplicación, siendo esencial para que el desarrollo se realizara de manera eficaz.

>>Google AdMob: es la manera de monetización que ha sido usada en LocMaBar para generar ingresos de manera no intrusiva. Permitiendo así, mantener la aplicación gratuita para los usuarios mientras se cubren los costes de desarrollo y mantenimiento.

## Normativa que aplica

**Reglamento General de Protección de Datos (RGPD):**

El RGPD es una normativa de la Unión Europea que protege los datos personales y la privacidad de los usuarios. Se aplica a todas las aplicaciones que recopilen, procesen o almacenen datos de usuarios pertenecientes a la Unión Europea.

-Consentimiento: LocMaBar debe obtener el permiso de los usuarios antes de recopilar sus datos(geolocalización, nombre, correo).

-Transparencia: LocMaBar debe informar a los usuarios sobre como se usarán sus datos(por ejemplo, geolocalización o valoraciones) en una política de privacidad a la que se pueda acceder desde la ventana principal.

-Medidas de seguridad: Se han utilizado Firebase Authentication y Firestore con reglas de seguridad en las que se restringe el acceso a los datos solo a los usuarios autenticados en la aplicación, además de cifrado en tránsito proporcionado por Firebase.

-Notificación a AEPD: En caso de que se un robo de la información, la aplicación debe notificar a la Agencia Española de Protección de Datos(AEPD) en un plazo máximo de 72 horas.

**-Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales (LOPDGDD):**

Esta ley adapta el RGPD(reglamento general de protección de datos) a la normativa española y añade los derechos y obligaciones específicas en el ámbito de protección de datos.

-Transparencia: La ley exige que la información sobre el tratamiento de datos sea clara y comprensible además de lo que exige RGPD.

-Gestión de cuenta: La ley regula la creación, uso y eliminación de cuentas de usuario, garantizando que los datos del usuario se eliminen en su totalidad si este así lo solicita.

-Contenido digital: Esta ley establece responsabilidades sobre el contenido realizado por usuarios, como los comentarios, asegurando que no se infrinja derechos de terceros.

**-Ley 34/2002, de Servicios de la Sociedad de la Información y de Comercio Electrónico(LSSI-CE):**

Esta ley regula las actividades realizadas a través de internet y aplicaciones móviles en España, incluyendo LocMaBar. Esta ley establece obligaciones para las plataformas digitales que ofrecen un servicio a usuarios.

-Identificación del prestador: En la app se debe mostrar el nombre del desarrollador, dirección y datos de contacto.

-Protección de datos: En esta ley se complementan las 2 anteriores, RGPD y LOPDGDD, pidiendo permiso de uso de las cookies.

-Consentimiento de notificaciones: La app requiere el consentimiento del usuario para cualquier envío de notificaciones push o emails promocionales.

**-Ley General para la Defensa de los Consumidores y Usuarios(Real Decreto Legislativo 1/2007):**

Esta ley protege los derechos de los consumidores en todo tipo de transacciones comerciales, incluyendo servicios digitales como aplicaciones móviles que son gratuitas y con publicidad. Es importante porque se protege al usuario frente a intentos de estafa y asegura transparencia.

-Información clara y veraz: LocMaBar debe proporcionar información sobre los bares y restaurantes que sea precisa y no falsa. Por ejemplo, locales que estén cerrados no se los debe mostrar como abiertos.

-Publicidad: Los anuncios de AdMob deben ser transparentes y no inducir a error sobre los servicios de la app.

-Canales e reclamación: La app tiene que ofrecer un medio accesible para que los usuarios puedan presentar quejas, como un formulario o un email de soporte(Por ejemplo, support@locmabar.com).

-Derecho de abandono: Si se incluyen compras dentro de la aplicación, los usuarios deben tener unos 14 días para poder cancelarlas.

**-Ley de Propiedad intelectual-Real Decreto Legislativo 1/1996:**

Esta ley protege los derechos de autor como el código de la aplicación, los diseños en Figma y el contenido proporcionado por los usuarios.

-Uso de material ajeno: Cualquier imagen, texto o canción cogido de terceros debe tener el permiso o estar bajo licencia libre.

-Protección de contenido propio: El código y diseños de la aplicación están protegidos por derechos de autor, y se debe evitar una copia no autorizada.

-Contenido de usuarios: LocMaBar tiene que asegurarse de que en caso de que se puedan subir imágenes o textos no se infrinjan los derechos de autor.

## Justificación de tecnologías elegidas

En LocMaBar, han sido seleccionadas las siguientes tecnologías para su capacidad para cubrir las necesidades funcionales, de seguridad y escalabilidad de una aplicación móvil cuyo objetivo es hacer llegar los bares y restaurantes cercanos al usuario, garantizando así una experiencia única y una gestión eficaz de los datos. Con un enfoque especial en los pequeños municipios del país a donde otras apps o páginas similares no suelen llegar:

**-Android Studio y Kotlin**: se ha optado por usar Kotlin como lenguaje de programación por la claridad, seguridad y por su soporte dentro del entorno de desarrollo de aplicaciones móviles. Es el lenguaje más adecuado para crear una app fluida y fácil de mantener. Android Studio, como entorno de desarrollo, que nos permite crear una interfaz sencilla y optimizada para diferentes dispositivos, garantizando una experiencia cómo al usuario.

**-Firebase**: ha sido seleccionada por su capacidad para ofrecer un servicio que cumple con los requisitos de autenticación y almacenamiento de datos en tiempo real, lo necesario para LocMaBar.

•Firebase Authentication: Se encarga de gestionar el registro y la autenticación de usuarios usando email y contraseña, incluyendo cuentas de usuarios estándar y de administrador. Es un sistema confiable y fácil de usar, con medidas de seguridad como cifrado y gestión de sesiones, además de cumplir con normativas como el RGPD. Firebase Authentication también permitiria agregar otros métodos en el futuro, como iniciar sesión con Google, dándole así más flexibilidad a LocMaBar.

•Firebase Firestore: Ha sido elegida como la base de datos en la nube para guardar la información sobre bares, valoraciones y nuevas propuestas de locales. Su capacidad para sincronizar los datos en tiempo real quiere decir que todos los usuarios siempre tendrán datos actualizados, incluso cuando muchas personas están usando la aplicación al mismo tiempo. Esto es súper importante para una aplicación como esta, que depende de datos que cambian constantemente, como horarios o comentarios. Para esta aplicación que se espera que vaya en crecimiento cada vez más, Firestore es capaz de soportar el gran volumen de usuarios que puede llegar a tener sin que la app se vuelva lenta. Su forma de organizar los datos en colecciones y documentos encaja perfectamente con el funcionamiento de LocMaBar.

**-Figma**: Para diseñar los prototipos de la interfaz de usuario de LocMaBar, se ha utilizado Figma, una herramienta online que nos permite trabajar en equipo de forma sencilla y en línea. Facilita la colaboración en tiempo real, lo que hace que el proceso de mejorar y ajustar los diseños sea mucho más rápido. Además, una de las funciones clave de Figma es su plugin “Upspeed”. Este plugin convierte los diseños directamente en código Kotlin, lo que hace que sea mucho más fácil integrarlos en Android Studio. También podemos crear componentes reutilizables, como botones o campos de texto, que se mantienen iguales en toda la app. Esto es súper importante para que la interfaz sea visualmente coherente y fácil de usar, cumpliendo así con uno de los requisitos clave de usabilidad y ofreciendo una experiencia sencilla e intuitiva para los usuarios.

**-Google AdMob**: Se ha usado para la monetización, permitiendo incluir anuncios no intrusivos a la aplicación. Es una solución que permite generar ingresos sin comprometer la experiencia del usuario, permitiendo así mantener la aplicación gratuita.

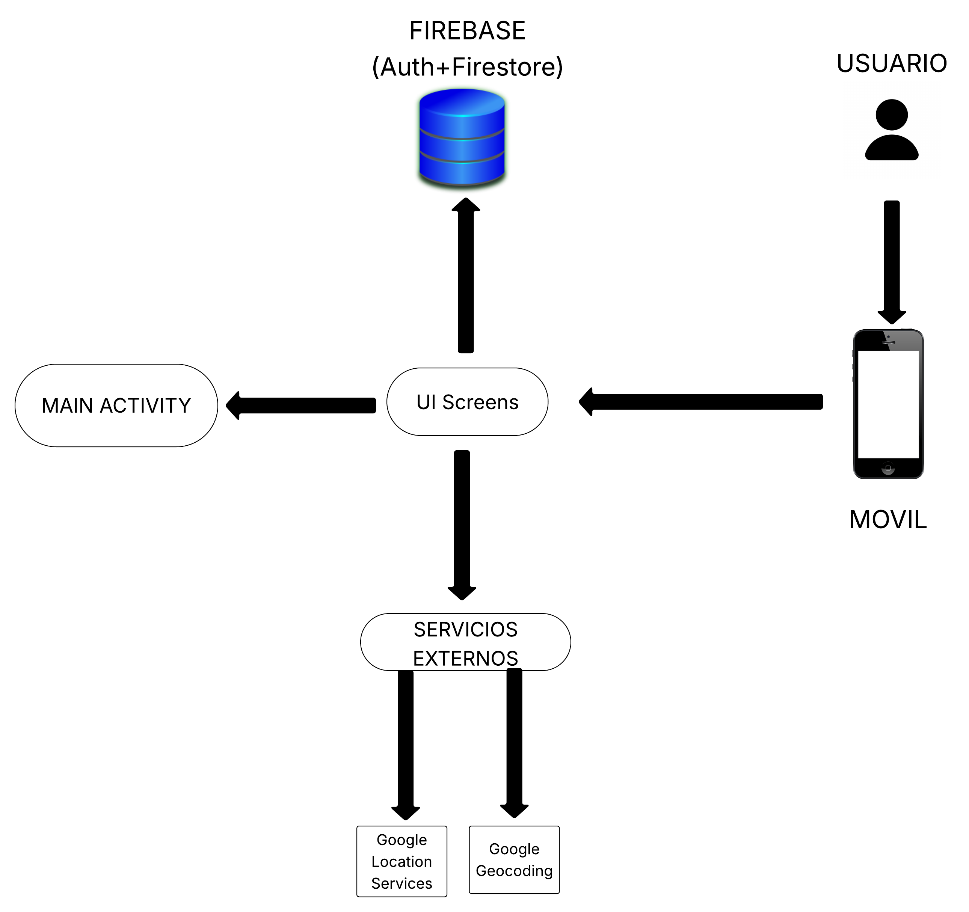
**-Geolocalización**: La función de geolocalización en la aplicación está basada en Google Location Services. La razón de usarlo es que es muy preciso y consume poca batería, algo clave para LocMaBar. Gracias a esta tecnología, podemos detectar la ubicación del usuario (latitud y longitud) combinando datos de GPS, Wi-Fi y redes móviles. Esto permite ofrecer resultados precisos y eficientes, como buscar bares en un radio de 50 km. La elección de esta tecnología también se debe a su popularidad en apps Android, ya que más del 80% de las aplicaciones de geolocalización en España en 2024 la usan . Además, cumple con las normas de privacidad del RGPD, ya que pide permiso de ubicación de forma clara y explícita. Integrarlo en Android Studio es sencillo, lo que facilita que LocMaBar pueda ofrecer recomendaciones basadas en la ubicación del usuario rápidamente y de forma segura.

**-Geocoding API**: La API de Geocoding de Google Cloud se ha elegido para que nuestra aplicación pueda convertir direcciones físicas, como 'Calle Paseo, 1, Iniesta, Cuenca', en coordenadas geográficas, es decir, latitud y longitud. Esto es muy importante para LocMaBar, porque ayuda a asociar esas coordenadas con los bares que los usuarios solicitan que sean agregados y a mostrar su ubicación exacta cuando los usuarios buscan por geolocalización. La razón por la que optamos por esta API es porque ofrece una gran precisión y cubre casi todo el mundo. La integración con Google Location Services nos permite tener un flujo de datos ágil, y las coordenadas que obtenemos se almacenan en Firestore, para usarlas en futuras búsquedas por ubicación. Además, esta API cumple con las reglas de privacidad, no guarda datos del usuario más allá de lo necesario para procesar la información. En cuanto a su costo, es de unos 5 euros más o menos cada 1000 solicitudes, lo que lo hace una opción asequible para la fase inicial de LocMaBar.

# Solución Técnica

## Entornos de desarrollo y tecnología (arquitectura, patrón de diseño elegido)

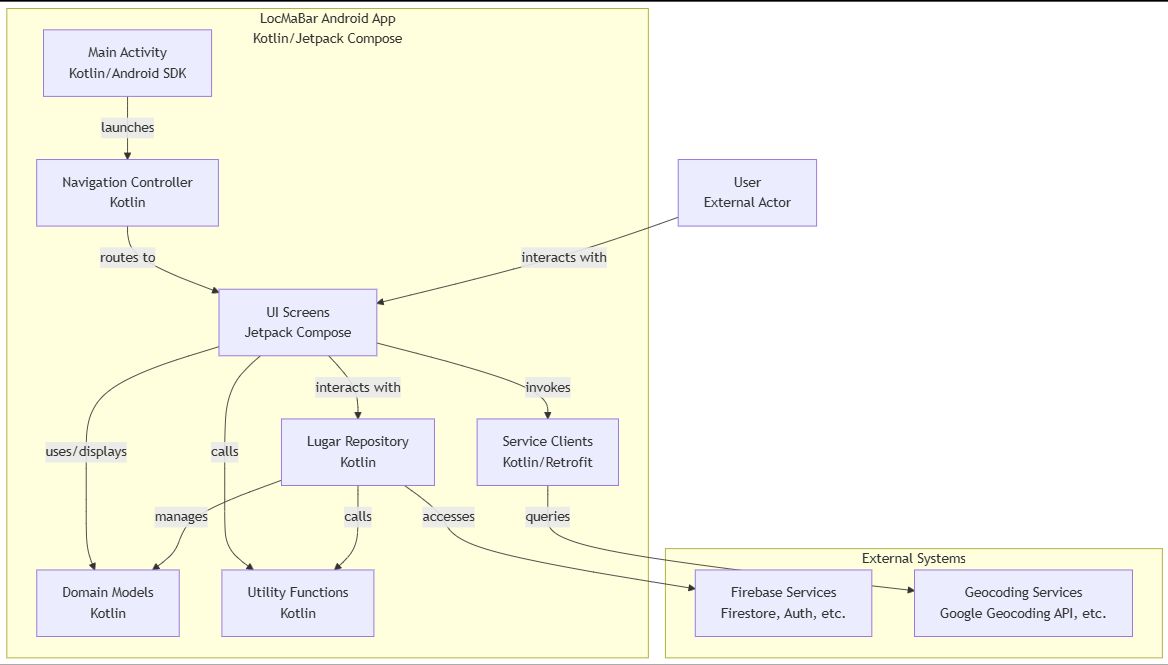
### Diagrama de Componentes



El siguiente diagrama ha sido realizado mediante una extensión de la que dispone Visual Studio

Code llamada “CodeViz” en la que simplemente tu abres tu proyecto en Visual Studio y llamas a

esta estensión y te saca el diagrama de componentes siguiente:



•**User Interface**: Incluye todas las pantallas y elementos que manejan la interacción del usuario. Aquí se agrupan las principales actividades implementadas con Jetpack Compose, como la ventana principal para administradores, la ventana de gestión de solicitudes y locales, ventana de gestión de usuarios y la ventana de gestión de comentarios.

•**Controllers**: Contiene las clases y servicios que gestionan la lógica entre la interfaz de usuario y los modelos.

•**Models**: Incluye las clases de datos que representan los modelos de la aplicación.

•**Data Sources**: Incluye las fuentes de datos remotas que proveen los datos necesarios para la aplicación. LocMaBar utiliza Firebase Firestore como fuente de datos principal, las colecciones que hay creadas almacenan y gestionan los datos en tiempo real, garantizando sincronización inmediata para todos los usuarios.

•**External Components**: Incluye servicios externos integrados en la aplicación como google location services, google geocoding o google admob.

**Entornos de desarrollo:**

Para el desarrollo de LocMaBar, se ha seleccionado Android Studio para el desarrollo de la aplicación en Kotlin.

**Patrón de diseño:** Se ha optado por el patrón de diseño MVC (Modelo-Vista-Controlador) para organizar el código de manera eficiente, asegurando una separación clara entre la lógica, la presentación y la gestión de datos.

## Análisis (requisitos funcionales y no funcionales, casos de uso)

### Requisitos funcionales

**Autenticación y gestión de usuarios:**

• Registro de usuarios: debe permitir que los usuarios se registren usando su nombre, correo y contraseña, guardándose los datos en firebase.

• Iniciar sesión: Los usuarios deben poder entrar a la app con su correo y contraseña, verificando sus datos en firebase.

• Editar perfil: la opción de editar el perfil debe estar disponible, y los cambios deben guardarse en firestore.

• Gestión de usuarios para administradores: los administradores deben tener la capacidad de editar o eliminar cuentas de usuarios.

**Funcionalidades de búsqueda y sugerencias:**

• Buscar por provincia y municipio: la app debe permitir buscar bares seleccionando la comunidad autónoma, provincia y ciudad.

• Buscar por geolocalización: debe ser posible buscar bares cercanos usando la ubicación actual, con Google Location Services y limitando el radio a unos 50 km.

• Sugerir nuevos locales: Los usuarios pueden proponer nuevos locales, usando Google Cloud Geocoding API para obtener la ubicación y guardar esas solicitudes en firestore.

**Gestión de valoraciones y locales:**

• Poner valoraciones: los usuarios pueden valorar los bares, con esas valoraciones almacenadas en firestore y en estado "PENDIENTE".

• Gestionar comentarios: los administradores tienen la posibilidad de aprobar o rechazar comentarios.

• Administrar locales: debe permitir a los administradores aprobar, rechazar, editar o eliminar locales.

**Persistencia:**

• Almacenamiento de datos: toda la información debe guardarse en firestore para asegurar que la app tenga datos en todo momento y que se sincronicen en tiempo real.

**Monetización:**

• Publicidad: debe mostrar anuncios a través de Google AdMob en momentos

específicos, procurando que no interfieran con la experiencia del usuario.

### Requisitos no funcionales

**Rendimiento:**

• Tiempo de respuesta: Las acciones principales deberían hacerse en menos de 4 segundos siempre que la conexión sea estable.

• Escalabilidad: Firestore debe poder manejar al menos unos 5,000 registros sin que la app se vuelva lenta o deje de funcionar bien.

• Optimización: La aplicación tiene que funcionar en dispositivos Android desde una API 24 hasta API 35, incluyendo teléfonos como el Xiaomi Redmi 12C.

**Seguridad:**

• Protección de datos: los datos de los usuarios deben estar seguros usando Firebase

Authentication y reglas específicas de Firestore, para que los usuarios no autorizados no

tengan acceso.

• Cumplimiento normativo: necesita cumplir con leyes como el RGPD y la LOPDGDD,

pidiendo permiso para usar la ubicación y protegiendo la información personal.

**Usabilidad:**

• Interfaz de usuario: debe ser intuitiva, fácil de usar y no intrusiva.

• Accesibilidad: debe ser funcional para personas con diferentes niveles de

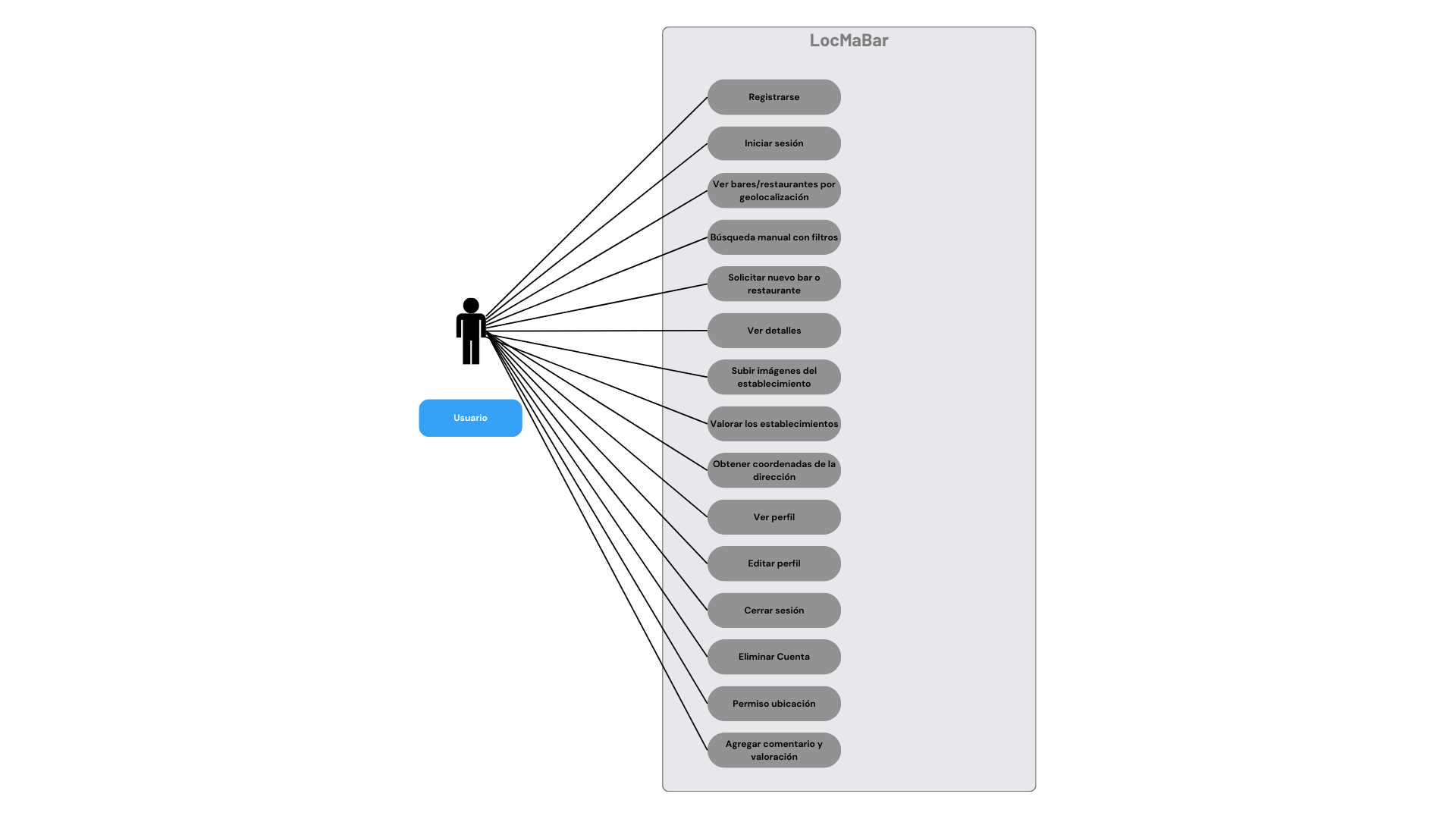
conocimientos tecnológicos.

**Disponibilidad:**

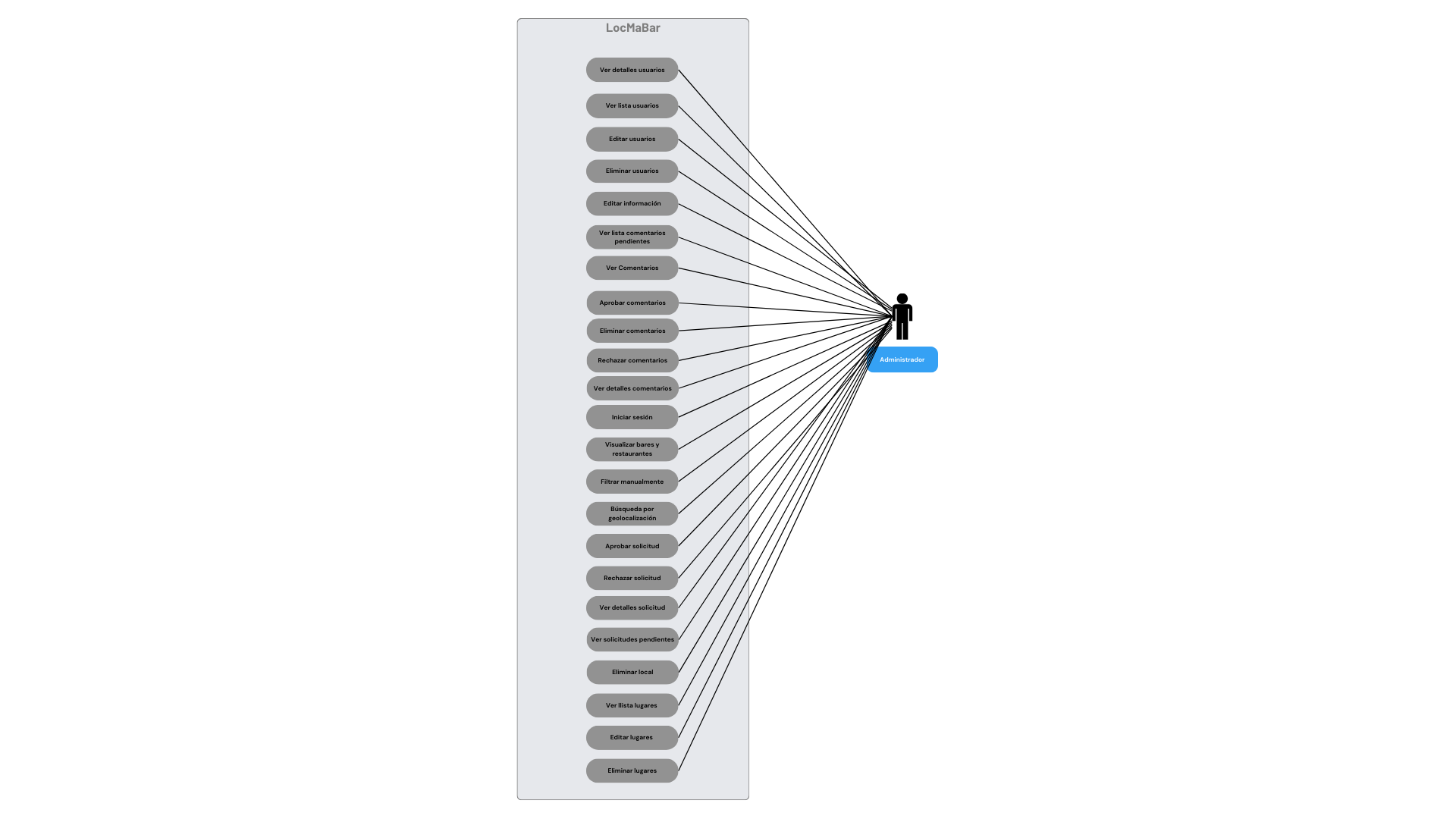
• Acceso: debe estar siempre disponible cuando el usuario tenga

internet, y mostrar mensajes si hay problemas con la app.

Usuario:



Administrador:



### Casos de uso

Usuario

|  |  |  |
| --- | --- | --- |
| CU-01 | Registro de Usuario | |
| Objetivo asociado | Gestión de Usuarios | |
| Descripción | La app permite a un usuario registrarse con un formulario, creando una cuenta en Firebase y un documento en Firestore. | |
| Precondición | Que ese usuario no esté ya registrado en firebase y que se tenga conexión a internet. | |
| Secuencia  Normal  (Flujo de eventos) | **Paso** | **Acción** |
| N1 | Usuario solicita el registro. |
| N2 | El usuario rellena los datos que se le solicitan. |
| N3 | El sistema verifica que el usuario no exista. |
| N4 | Se validan los datos ingresados por el usuario. |
| N5 | Si se ha rellenado todo bien se registra el nuevo usuario. |
| N6 | Se muestra el mensaje de que el registro ha sido exitoso. |
| Postcondición | Queda registrado como nuevo usuario en el sistema. | |
| Excepciones | **Paso** | **Acción** |
| N1 | Si el usuario ya está registrado, se informa al solicitante y el caso de uso finaliza |
| N2 | Si los datos no son correctos, el sistema muestra un mensaje de error y solicita corrección |
| N2 | Si los campos están vacíos se muestra un mensaje pidiendo que se rellenen los campos necesarios |
| N3 | Si se cancela el registro, se descartan los datos y el caso de uso termina. |
| Rendimiento | **Paso** | **Acción** |
| N1 | 5 segundos |
| Frecuencia esperada | 1 vez por uso de cada usuario | |

|  |  |  |
| --- | --- | --- |
| CU-02 | Login de Usuario | |
| Objetivo asociado | Gestión de acceso | |
| Descripción | La app permite a los usuarios autenticarse | |
| Precondición | El usuario debe estar registrado en el sistema | |
| Secuencia  Normal  (Flujo de eventos) | **Paso** | **Acción** |
| N1 | El usuario ingresa el correo y contraseña |
| N2 | El sistema valida las credenciales introducidas |
| N3 | Se permite el acceso y se muestra la siguiente ventana |
| Postcondición | El usuario ha sido autenticado como usuario del sistema. | |
| Excepciones | **Paso** | **Acción** |
| N1 | Si las credenciales son incorrectas se muestra un mensaje de error y se permite reintentar el ingreso |
| N2 | Si los campos están vacíos se muestra un mensaje pidiendo que se rellenen los campos necesarios |
| Rendimiento | **Paso** | **Acción** |
| N1 | 4 segundos |
| Frecuencia esperada | 1 vez al día | |

|  |  |  |
| --- | --- | --- |
| CU-03 | Búsqueda por comunidad, provincia y municipio | |
| Objetivo asociado | Búsqueda de bares y restaurantes | |
| Descripción | La app permite la búsqueda manual de los bares y restaurantes eligiendo la comunidad, provincia y municipio. | |
| Precondición | El usuario debe estar autenticado y que funcione Firestore. | |
| Secuencia  Normal  (Flujo de eventos) | **Paso** | **Acción** |
| N1 | El sistema muestra 3 desplegables. |
| N2 | El usuario selecciona comunidad autónoma. |
| N3 | En el 2 desplegable se muestran las provincias correspondientes a esa CCAA. |
| N4 | Selección de una de las provincias. |
| N5 | En el 3 desplegable se muestran los municipios correspondientes a la provincia. |
| N6 | Se selecciona uno de los municipios. |
| N7 | Con los filtros seleccionados se consulta en la colección de firestore. |
| N8 | Se muestran los resultados encontrados. |
| Postcondición | Se muestran los bares y restaurantes buscado con filtros. | |
| Excepciones | **Paso** | **Acción** |
| N1 | Si no se encuentran resultados, se muestra un mensaje de que no se ha encontrado nada |
| N2 |  |
| Rendimiento | **Paso** | **Acción** |
| N1 | 6 segundos |
| Frecuencia esperada | Varias veces | |

|  |  |  |
| --- | --- | --- |
| CU-04 | Búsqueda por geolocalización | |
| Objetivo asociado | Búsqueda de bares y restaurantes por medio de la localización del usuario. | |
| Descripción | La aplicación busca automáticamente los bares y restaurantes cercanos a la ubicación del usuario que usa la app. | |
| Precondición | Haber dado permiso de acceso a la ubicación. | |
| Secuencia  Normal  (Flujo de eventos) | **Paso** | **Acción** |
| N1 | Consentimiento de ubicación del usuario obtenido. |
| N2 | El sistema obtiene las coordenadas del usuario, consulta la colección en firestore y muestra los lugares cercanos. |
| Postcondición | Se muestran los bares y restaurantes buscado por geolocalización. | |
| Excepciones | **Paso** | **Acción** |
| N1 | Si el usuario no concede permiso a la ubicación, se activa la búsqueda manual y se muestra un mensaje. |
| N2 | Si no hay conexión con Firebase, se muestra un mensaje de error. |
| N2 | Si no se visualizan los locales, es porque no hay cercanos. |
| Rendimiento | **Paso** | **Acción** |
| N2 | 1 segundo |
| Frecuencia esperada | Varias veces al día. | |

|  |  |  |
| --- | --- | --- |
| CU-05 | Editar Perfil de Usuario | |
| Objetivo asociado | Gestión de Usuarios | |
| Descripción | El sistema permite a los usuarios autenticados editar su perfil. | |
| Precondición | El usuario debe estar autenticado y se debe tener acceso a la base de datos. | |
| Secuencia  Normal  (Flujo de eventos) | **Paso** | **Acción** |
| N1 | Usuario accede a la ventana de perfil. |
| N2 | Usuario pulsa el botón de “Editar Perfil”. |
| N3 | Se muestran los datos del usuario. |
| N4 | El usuario modifica los datos que pueden editarse. |
| N5 | El usuario confirma los cambios. |
| N6 | El sistema valida los datos. |
| N7 | Se actualizan los datos en Firestore. |
| N8 | Se muestra mensaje de éxito. |
| Postcondición | Datos editados han sido actualizados. | |
| Excepciones | **Paso** | **Acción** |
| N4 | Si los datos ingresados no son válidos, se muestra un mensaje de error. |
| N5 | Si el usuario cancela la edición, se descartan los cambios. |
| N7 | Si no hay conexión con Firebase, se muestra un mensaje de error. |
| Rendimiento | **Paso** | **Acción** |
| N7 | 3 segundos |
| Frecuencia esperada | Pocas veces | |

|  |  |  |
| --- | --- | --- |
| CU-06 | Escribir comentario de un Local | |
| Objetivo asociado | Gestión de comentarios | |
| Descripción | El sistema permite al usuario comentar y valorar sobre un local. | |
| Precondición | El usuario debe de estar autenticado, tiene que existir el Local y se debe poder acceder a Firestore. | |
| Secuencia  Normal  (Flujo de eventos) | **Paso** | **Acción** |
| N1 | Usuario accede a la ventana detalles del local. |
| N2 | El usuario escribe un comentario y hace una valoración. |
| N3 | El usuario envía el comentario. |
| N4 | El sistema almacena el comentario en “ComentariosPendientes” en estado “PENDIENTE”. |
| N5 | Se muestra mensaje de que se ha enviado el comentario. |
| Postcondición | Datos editados han sido actualizados. | |
| Excepciones | **Paso** | **Acción** |
| N3 | Si el comentario está vacío salta un mensaje de error. |
| N2 | Si la valoración no está entre 1 y 5, se muestra un mensaje de error. |
| N3 | Si no hay conexión con Firebase, se muestra un mensaje de error. |
| Rendimiento | **Paso** | **Acción** |
| N3 | 2 segundos |
| Frecuencia esperada | Pocas veces | |

|  |  |  |
| --- | --- | --- |
| CU-07 | Solicitar Agregar un Restaurante | |
| Objetivo asociado | Gestión de Solicitudes | |
| Descripción | El sistema permite al usuario solicitar que se agregue un bar o restaurante nuevos rellenando un formulario. | |
| Precondición | El usuario debe estar autenticado y se debe tener acceso a Firestore y a la API de Geocodificación. | |
| Secuencia  Normal  (Flujo de eventos) | **Paso** | **Acción** |
| N1 | Usuario accede a la ventana de solicitud de un nuevo bar o restaurante. |
| N2 | El usuario ingresa los datos que se le piden. |
| N3 | El usuario solicita obtener las coordenadas. |
| N4 | El sistema usa la Geocodificación para obtener las coordenadas. |
| N5 | El usuario confirma el envío de la solicitud. |
| N6 | El sistema valida los datos. |
| N7 | El sistema guarda la solicitud en la colección “Solicitudes”. |
| N8 | Se muestra mensaje de éxito. |
| Postcondición | Solicitud enviada para ser aprobada o rechazada por el administrador. | |
| Excepciones | **Paso** | **Acción** |
| N2 | Si los campos están vacíos, se muestra un mensaje de error. |
| N4 | Si no se obtienen correctamente las coordenadas, se muestra un mensaje de error. |
| N4 | Si no hay conexión a internet, se muestra un mensaje de error. |
| N7 | Si no hay conexión con Firebase, se muestra un mensaje de error. |
| Rendimiento | **Paso** | **Acción** |
| N7 | 4 segundos |
| Frecuencia esperada | Ocasional. | |

Administrador

|  |  |  |
| --- | --- | --- |
| CU-08 | Aprobar o Rechazar Comentario | |
| Objetivo asociado | Gestión de comentarios | |
| Descripción | El sistema permite al administrador aprobar o rechazar los comentarios proporcionados por los usuarios. | |
| Precondición | El usuario debe ser administrador. | |
| Secuencia  Normal  (Flujo de eventos) | **Paso** | **Acción** |
| N1 | El administrador accede a la ventana de detalles de comentarios. |
| N2 | El administrador revisa el comentario y decide si lo aprueba o rechaza. |
| N3 | Si lo aprueba, el estado del comentario pasa de “PENDIENTE” a “APROBADO”. |
| N4 | El sistema mueve el comentario a la colección “Comentarios”. |
| N5 | Si lo rechaza, el comentario se elimina de la colección. |
| Postcondición | El comentario ha sido aprobado y es visible para los usuarios o el comentario ha sido rechazado y se ha eliminado. | |
| Excepciones | **Paso** | **Acción** |
| N3/N4 | Si no hay conexión con Firebase, se muestra un mensaje de error. |
| Rendimiento | **Paso** | **Acción** |
| N3/N4 | 3 segundos |
| Frecuencia esperada | Ocasional. | |

|  |  |  |
| --- | --- | --- |
| CU-09 | Aprobar o Rechazar Solicitud | |
| Objetivo asociado | Gestión de solicitudes | |
| Descripción | El sistema permite al administrador aprobar o rechazar la solicitud de un nuevo bar o restaurante. | |
| Precondición | El usuario debe ser administrador. | |
| Secuencia  Normal  (Flujo de eventos) | **Paso** | **Acción** |
| N1 | El administrador accede a la ventana de solicitudes. |
| N2 | El administrador revisa el comentario, y decide si lo aprueba o lo rechaza. |
| N3 | Si lo aprueba, el nuevo bar o restaurante el estado “PENDIENTE” se cambia. |
| N4 | El sistema mueve el bar o restaurante a la colección “Locales”. |
| N5 | Si lo rechaza, la solicitud se elimina de la colección. |
| Postcondición | La solicitud ha sido aprobada y es visible para los usuarios o la solicitud ha sido rechazada y se ha eliminado. | |
| Excepciones | **Paso** | **Acción** |
| N3/N5 | Si no hay conexión con Firebase, se muestra un mensaje de error. |
| Rendimiento | **Paso** | **Acción** |
| N3/N5 | 3 segundos |
| Frecuencia esperada | Ocasional. | |

|  |  |  |
| --- | --- | --- |
| CU-10 | Editar Local Aprobado | |
| Objetivo asociado | Gestión de Locales | |
| Descripción | El sistema permite editar los datos de los locales que ya son visibles para todos los usuarios autenticados. | |
| Precondición | El usuario debe ser administrador y se debe tener acceso a Firestore. | |
| Secuencia  Normal  (Flujo de eventos) | **Paso** | **Acción** |
| N1 | El administrador accede a la ventana de gestión de locales. |
| N2 | Selecciona uno de los locales y accede a un formulario con los datos actuales. |
| N3 | Modifica lo que necesite. |
| N4 | El administrador confirma que quiere guardar los cambios hechos. |
| N5 | El sistema valida los datos. |
| N6 | El sistema cambia los datos anteriores por los nuevos en la base de datos. |
| N7 | Se muestra mensaje de éxito. |
| Postcondición | El local ha sido actualizado por los nuevos datos. | |
| Excepciones | **Paso** | **Acción** |
| N4 | Si los datos no son válidos o los campos están vacíos, se muestra mensaje de error. |
| N6 | Si no hay conexión con Firestore, se muestra un mensaje de error. |
| Rendimiento | **Paso** | **Acción** |
| N6 | 2 segundos |
| Frecuencia esperada | Ocasional. | |

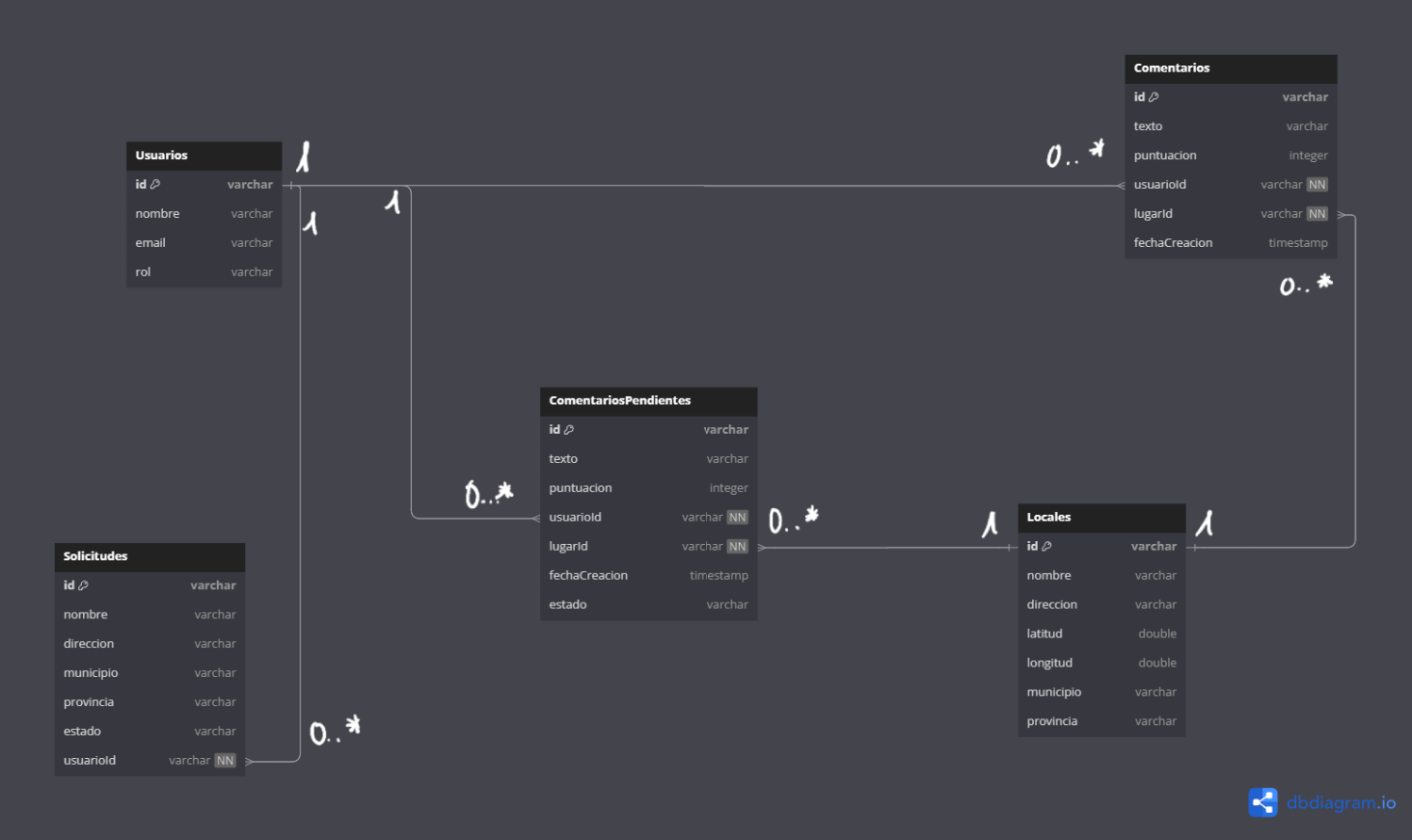
|  |  |  |
| --- | --- | --- |
| CU-11 | Eliminar Local Aprobado | |
| Objetivo asociado | Gestión de Locales | |
| Descripción | El sistema permite eliminar los locales que ya están visibles para los usuarios, en caso de que estos hayan cerrado u otra causa. | |
| Precondición | El usuario debe ser administrador y se debe tener acceso a Firestore. | |
| Secuencia  Normal  (Flujo de eventos) | **Paso** | **Acción** |
| N1 | El administrador accede a la ventana de gestión de locales. |
| N2 | Selecciona uno de los locales y aparece la opción de eliminar. |
| N3 | El administrador pulsa “Eliminar” y confirma que quiere eliminar. |
| N4 | El sistema elimina el local de la colección “Locales” en la base de datos. |
| N5 | Se muestra un mensaje de éxito. |
| Postcondición | El local ha sido eliminado de la base de datos y de la lista de locales en la app. | |
| Excepciones | **Paso** | **Acción** |
| N3 | Si el administrador cancela la eliminación, se cancela el proceso. |
| N4 | Si no hay conexión con Firestore, se muestra un mensaje de error. |
| Rendimiento | **Paso** | **Acción** |
| N4 | 2 segundos |
| Frecuencia esperada | Rara vez. | |

|  |  |  |
| --- | --- | --- |
| CU-12 | Eliminar Comentario Aprobado | |
| Objetivo asociado | Gestión de Comentarios | |
| Descripción | El sistema permite eliminar los comentarios que ya están visibles para los usuarios. | |
| Precondición | El usuario debe ser administrador y se debe tener acceso a Firestore. | |
| Secuencia  Normal  (Flujo de eventos) | **Paso** | **Acción** |
| N1 | El administrador accede a la ventana de gestión de comentarios. |
| N2 | Selecciona uno de los comentarios y aparece la opción de eliminar. |
| N3 | El administrador pulsa “Eliminar” y confirma que quiere eliminar. |
| N4 | El sistema elimina el comentario de la colección “Comentarios” en la base de datos. |
| N5 | Se muestra un mensaje de éxito. |
| Postcondición | El comentario ha sido eliminado de la base de datos y de la lista de comentarios en la app. | |
| Excepciones | **Paso** | **Acción** |
| N3 | Si el administrador cancela la eliminación, se cancela el proceso. |
| N4 | Si no hay conexión con Firestore, se muestra un mensaje de error. |
| Rendimiento | **Paso** | **Acción** |
| N4 | 2 segundos |
| Frecuencia esperada | Rara vez. | |

|  |  |  |
| --- | --- | --- |
| CU-13 | Eliminar Usuario | |
| Objetivo asociado | Gestión de Usuarios | |
| Descripción | El sistema permite eliminar los usuarios cuando esto sea necesario. | |
| Precondición | El usuario debe ser administrador y se debe tener acceso a Firestore. | |
| Secuencia  Normal  (Flujo de eventos) | **Paso** | **Acción** |
| N1 | El administrador accede a la ventana de gestión de usuarios. |
| N2 | Selecciona un usuario y aparece la opción de “Eliminar”. |
| N3 | El administrador pulsa “Eliminar” y confirma que quiere eliminar al usuario. |
| N4 | El sistema elimina el usuario de la colección “Usuarios” en la base de datos. |
| N5 | Se muestra un mensaje de éxito. |
| Postcondición | El usuario ha sido eliminado de la base de datos y de la lista de usuarios en la app. | |
| Excepciones | **Paso** | **Acción** |
| N3 | Si el administrador cancela la eliminación, se cancela el proceso. |
| N4 | Si no hay conexión con Firestore, se muestra un mensaje de error. |
| Rendimiento | **Paso** | **Acción** |
| N4 | 2 segundos |
| Frecuencia esperada | Rara vez. | |

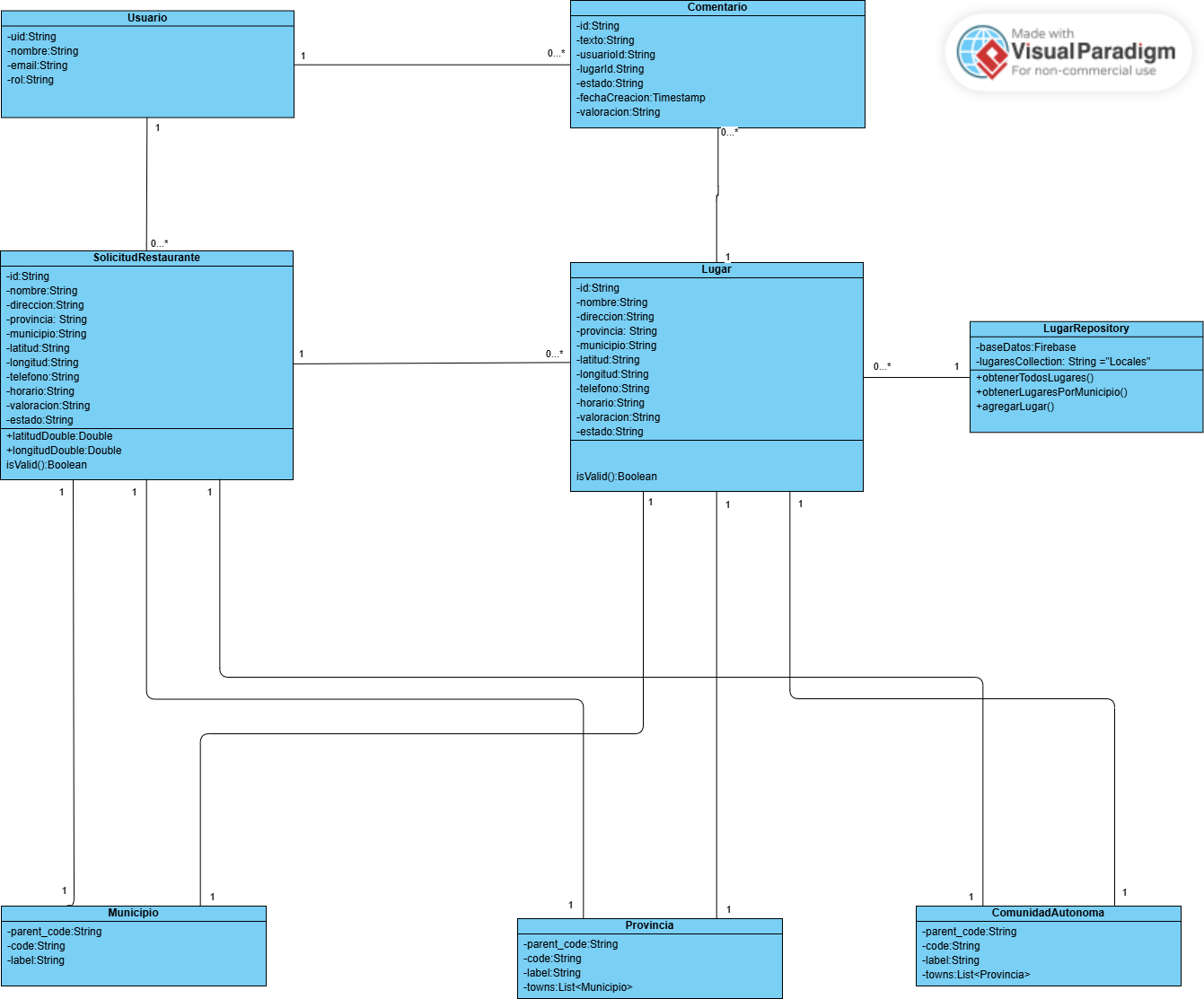
|  |  |  |
| --- | --- | --- |
| CU-14 | Editar Usuario | |
| Objetivo asociado | Gestión de Usuarios | |
| Descripción | El sistema permite editar la información de los usuarios cuando esto sea necesario. | |
| Precondición | El usuario debe ser administrador y se debe tener acceso a Firestore. | |
| Secuencia  Normal  (Flujo de eventos) | **Paso** | **Acción** |
| N1 | El administrador accede a la ventana de gestión de usuarios. |
| N2 | Selecciona un usuario y aparecen los campos que se pueden editar. |
| N3 | El administrador edita algún dato y confirma que quiere guardar los cambios. |
| N4 | El sistema actualiza los datos del usuario de la colección “Usuarios” en la base de datos. |
| N5 | Se muestra un mensaje de éxito. |
| Postcondición | Los cambios se han actualizado correctamente. | |
| Excepciones | **Paso** | **Acción** |
| N3 | Si el administrador cancela que quiere editar datos, se cancela el proceso. |
| N4 | Si no hay conexión con Firestore, se muestra un mensaje de error. |
| Rendimiento | **Paso** | **Acción** |
| N4 | 2 segundos |
| Frecuencia esperada | Rara vez. | |

## Diseño e implementación de la persistencia de los datos



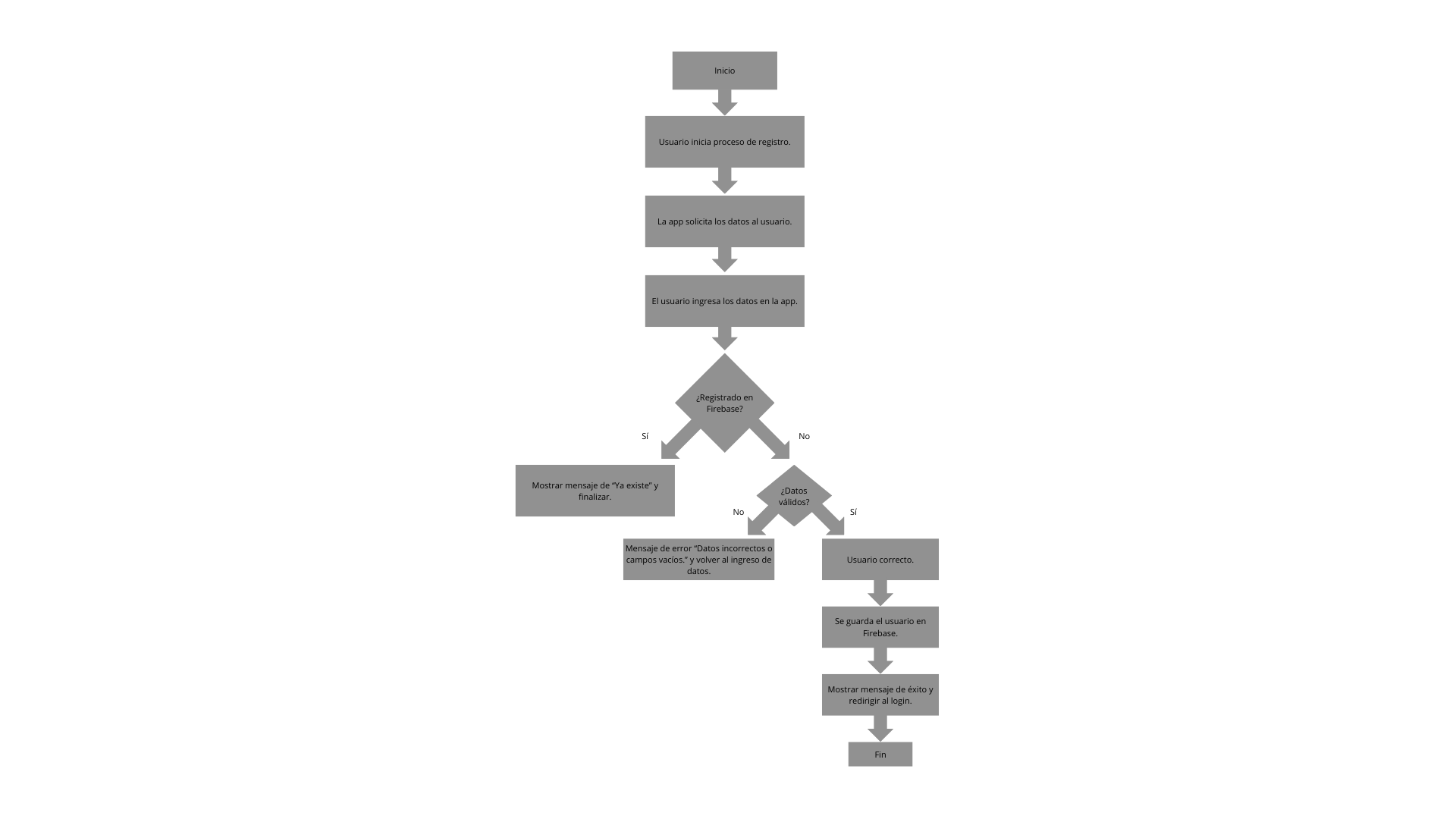
## Diagramas: clases y diagrama de flujo

### Diagrama de clases

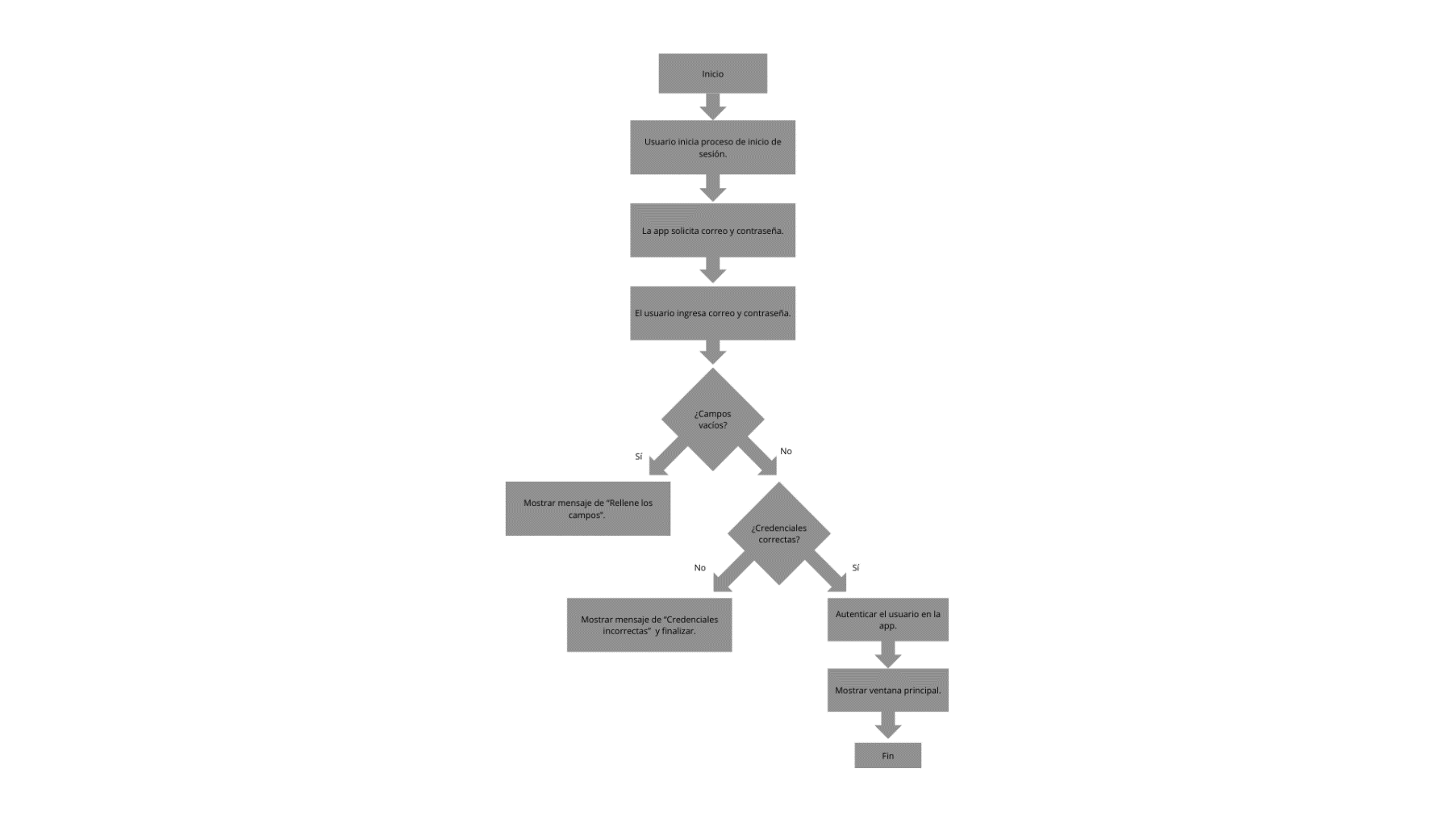


### Diagramas de Flujo

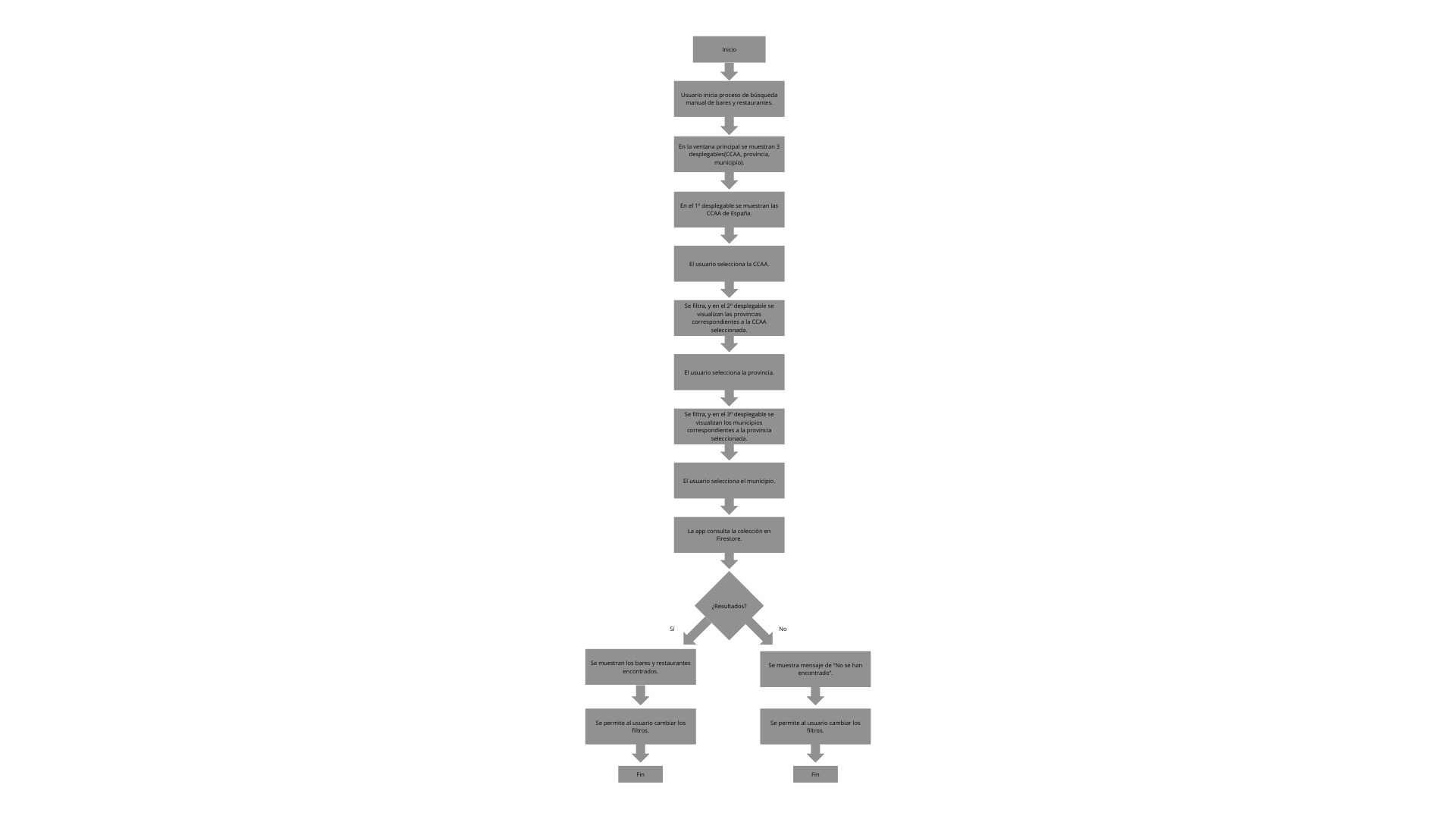
Registro del usuario:



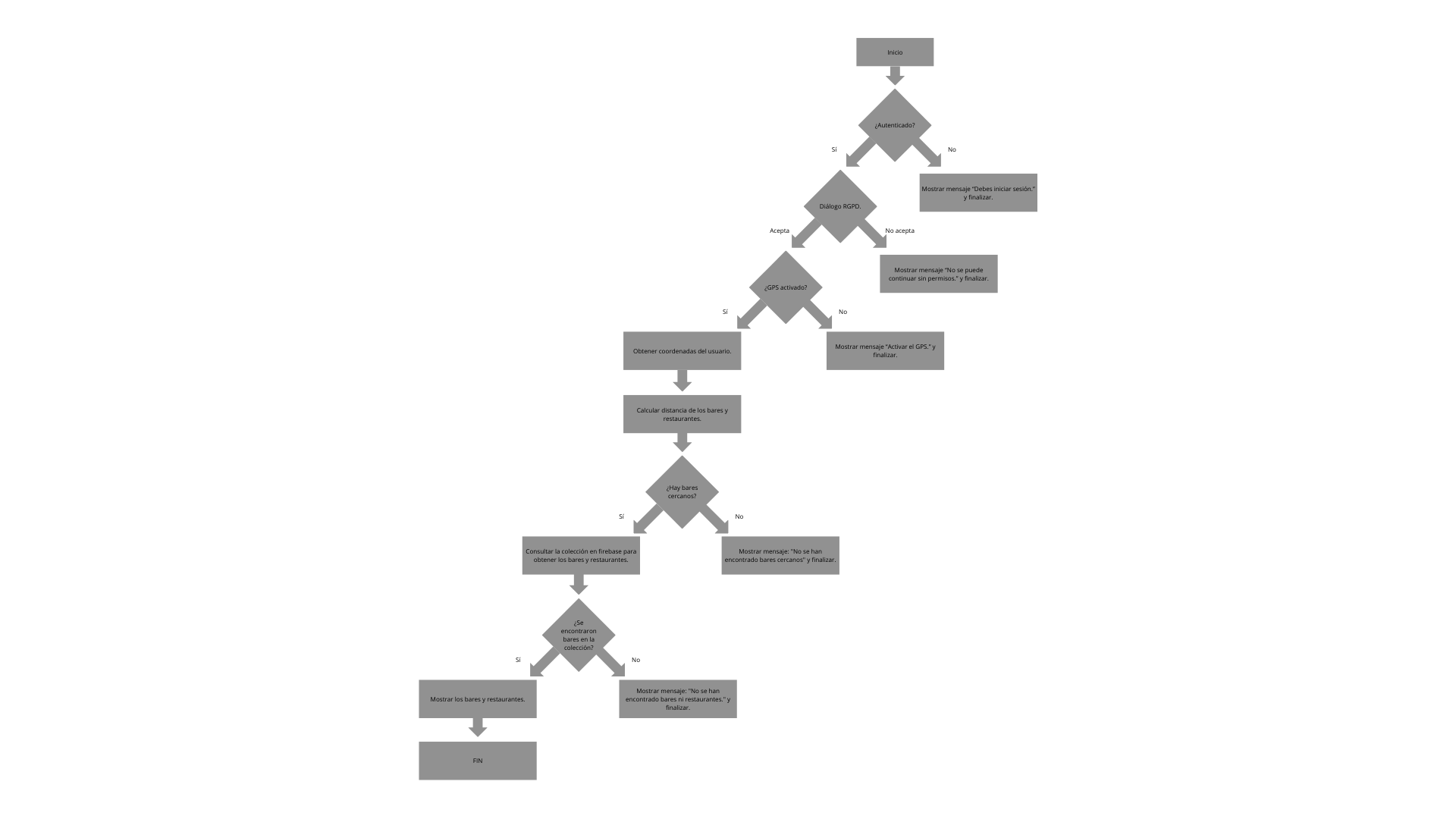
Inicio de sesión:



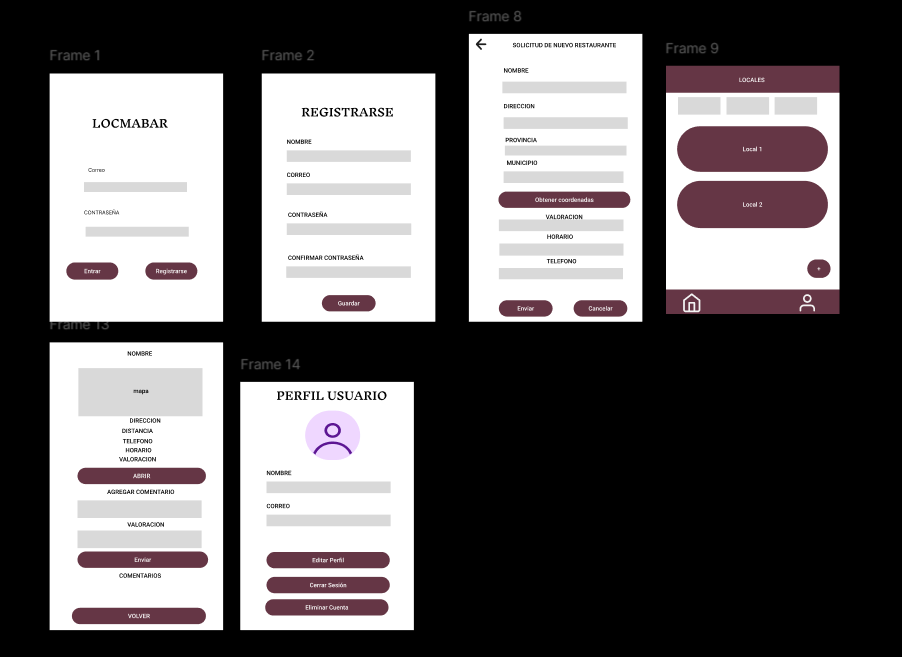
Búsqueda de bares y restaurantes manual:

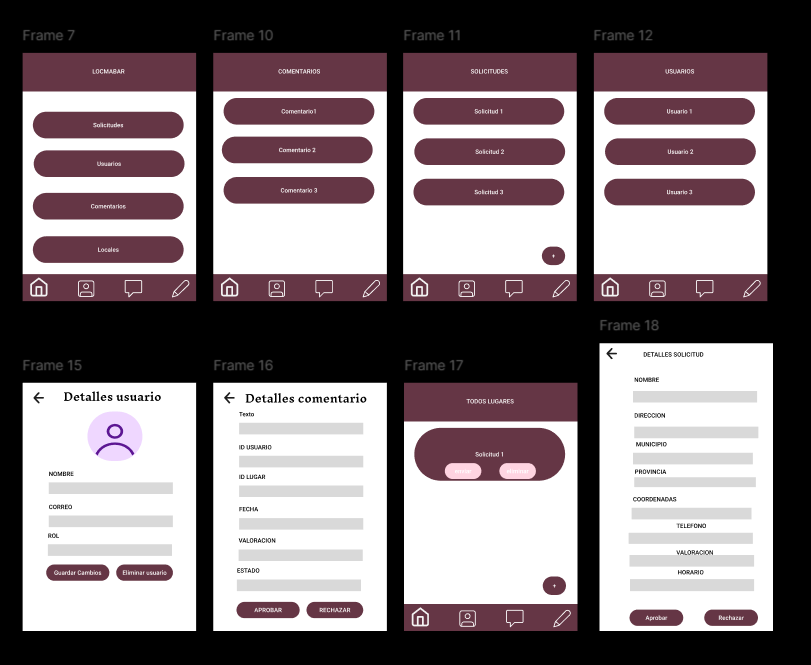


-Búsqueda de bares y restaurantes por geolocalización:



## Interfaces (desarrollados o mocks)





## Planes de pruebas

El objetivo del plan de pruebas es asegurarse de que la aplicación LocMaBar funcione bien y cumpla con todo lo que se espera de ella, tanto en lo que hace (funcionalidades) como en cómo lo hace (por ejemplo, que sea rápida y segura). Se quiere que los usuarios puedan usarla sin problemas, que sea fácil de entender, que no tenga fallos y que proteja sus datos.

**Ámbitos cubiertos por las pruebas**

>>Interfaz: garantizar que la app sea fácil e intuitiva para que la puedan usar todos los usuarios.

>>Rendimiento: Asegurar tiempos de respuesta rápidos y estabilidad bajo carga.

>>Seguridad: Proteger los datos mediante autenticación y reglas de Firestore,

evitando accesos no autorizados.

>>Compatibilidad: Comprobar el funcionamiento en diferentes versiones de

Android y dispositivos.

>>Funcionalidad: Comprobar que las funcionalidades principales funcionen

correctamente.

**Tipos de pruebas**

-Pruebas unitarias: validar funciones específicas, como la validación del formato del correo en el registro de los usuarios o en el inicio de sesión.

-Pruebas de rendimiento: comprobar los tiempos de respuesta a la hora de editar, borrar,

solicitar o agregar.

-Pruebas de seguridad: verificación de la seguridad de los datos del usuario y de la integridad

del sistema.

-Pruebas de usabilidad: se estudiará la interfaz y experiencia del usuario para garantizar que

la aplicación sea cómoda y atractiva.

-Pruebas de compatibilidad: probar la app en diferentes dispositivos Android para

comprobar su correcto funcionamiento en otros dispositivos.

**Estrategia de pruebas**

•Definición de casos de prueba: crear casos de prueba detallados para cada funcionalidad de la aplicación.

•Prioridad de las pruebas: se les da prioridad a las pruebas dando mayor prioridad a las pruebas de funcionalidad y de seguridad, que son las que afectan directamente la experiencia del usuario y la protección de datos.

•Ciclos de pruebas: en distintos ciclos se aplicarán unas pruebas u otras, por ejemplo, en un ciclo 1 las pruebas unitarias, en un ciclo 2 las pruebas de usabilidad y funcionalidad y en un ciclo 3 las pruebas de rendimiento, compatibilidad y seguridad.

**Criterios de aceptación**

•Criterios funcionales: todas las funcionalidades deben funcionar a la perfección.

•Criterios de rendimiento: la respuesta de las funciones de la aplicación debe ser aceptable

y firebase tiene que soportar el volumen de datos.

•Criterios de compatibilidad: la aplicación tiene que ser compatible en casi todos los

dispositivos Android de la gente a la que se quiere llegar.

•Criterios de seguridad: debe ser seguro ante los posibles accesos de usuarios no

autorizados.

•Criterios de usabilidad: navegación fluida, con mensajes de error claros.

**Documentación de pruebas**

•Informes de pruebas: informar de cada prueba con su descripción, resultados y fallos detectados.

•Reportes: se usará Trello para documentar los defectos que vayan apareciendo para darles una prioridad y, por último, investigar como enfrentarlos.

•Seguimiento: generar reportes semanales en Trello para revisar el estado de las pruebas y

asegurar que no queden errores críticos antes del lanzamiento.

**Gestión de defectos**

•Registro de defectos: cada defecto identificado se registrará con todo y detalles, con una descripción, los pasos que llevan a que se produzca y la gravedad.

•Priorización de defectos: los defectos detectados serán resueltos en base a la prioridad que se les haya marcado.

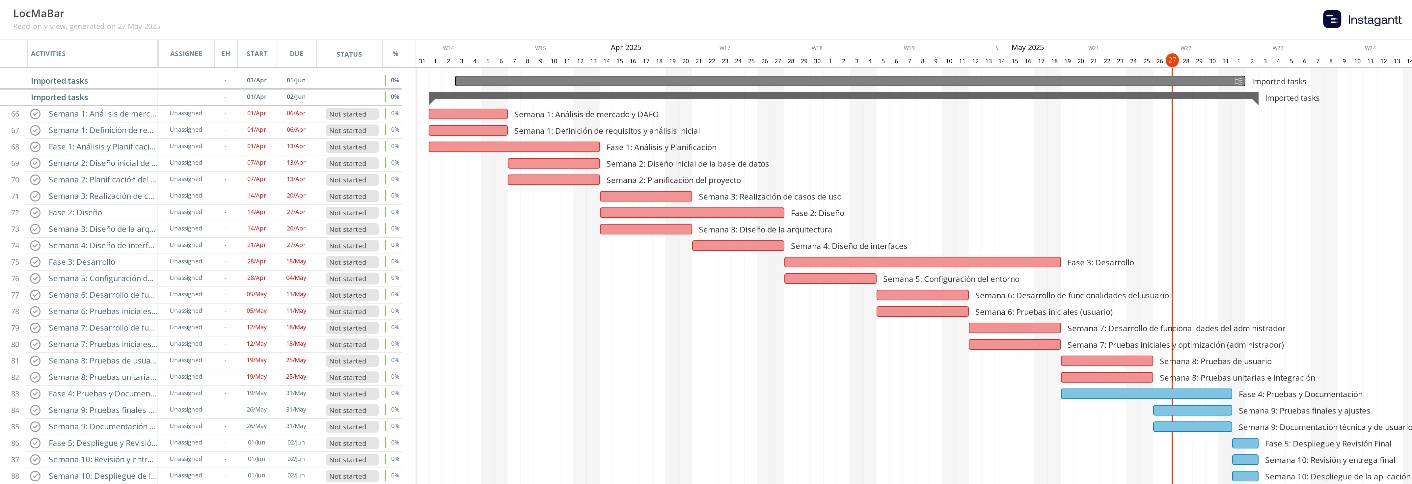
•Corrección y nuevas pruebas: se corregirán los errores que hayan aparecido y seguidamente, se realizarán nuevas pruebas para verificar que las correcciones no introduzcan nuevos errores en la aplicación.

A continuación, se redacta una tabla con uno de cada uno de los tipos nombrados anteriormente y que se han realizado o se realizarán en LocMaBar:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre Prueba** | **Medio de uso** | **Resultado esperado** | **Rendimiento** | **Resultado Obtenido** |
| Unitaria: Validación correo | Emulador de Android. | Que se valide correctamente el formato del correo | Se espera que sea en un tiempo menor de 1 segundo | Mensaje de error al no cumplir el formato; Correo válido aceptado. |
| Rendimiento: Aprobación de comentario | Emulador Android | Comprobar que el administrador puede aprobar un comentario. | Se espera que sea en menos de 3 segundos. | Comentario aprobado en menos de 2,5 segundos. |
| Seguridad: Protección de los datos | Emulador Android | Comprobar que solo usuarios autenticados pueden acceder a los datos. | Acceso no autorizado bloqueado. | Acceso denegado y se muestra mensaje de error de “Acceso denegado”. |
| Usabilidad: Navegación intuitiva | 2 participantes | Comprobar que los usuarios pueden realizar las funciones disponibles. | Un porcentaje de éxito bastante elevado. | 2 de 2 usuarios han conseguido hacer las funciones. |
| Compatibilidad: Funcionamiento diferentes versiones | Emulador android | Verificar el uso de la aplicación con distintas versiones de API. | Sin errores. | App ejecutada con éxito, es decir, sin errores. |

# Metodología y estimación de costes

## Diagrama de Gantt



## SW, HW, horas, amortización

Para la estimación del coste del proyecto, se han considerado los siguientes elementos necesarios para su desarrollo, implementación y fase inicial durante el primer año. A continuación, los costes de Software y Hardware:

**Software(SW):**

>>IDE (Android Studio): El entorno en el que se desarrolla la aplicación es gratuito, 0 €.

>>Firebase: Se utilizan Authentication y Firestore, para la fase inicial de LocMaBar se utilizará el plan gratuito que ofrece 10GB de almacenamiento y unas 50 mil operaciones de lectura diarias. En caso de expansión, se recurriría a el plan de 50€/año en el que se ofrecen 20GB de datos.

>>Sistema de Gestión de Código-Git(GitHub): Se emplea para el control de versiones y la gestión del código fuente, es gratuito, 0€.

>>Google AdMob: Integrar anuncios desde Google AdMob es gratuito, y se estima un ingreso de 100€/año con 1000 usuarios activos.

>>Figma: se emplea para diseñar la interfaz y en LocMaBar se usa la versión gratuita que permite hasta 3 proyectos de diseño, 0€.

>>Google Cloud Geocoding API: el servicio que se utiliza para convertir las direcciones físicas en coordenadas geográficas. Se usa la versión de pago con una media de 1000 usuarios activos al mes, el coste es de aproximadamente 5€.

>>Google Geolocalización: es gratuito, por ello se usa en LocMaBar, 0€.

**Hardware(HW):**

•**Equipo de desarrollo**: Portátil ASUS VivoBook Go 14/15 con procesador AMD Ryzen 5 7535HS (6 núcleos, 12 hilos, hasta 4.55 GHz), gráficos AMD Radeon 660M, 16 GB de RAM DDR5 y SSD de 512 GB. Este equipo es adecuado para programar en Android Studio, gestionar compilaciones y ejecutar emuladores, cumpliendo con los requisitos de rendimiento para el desarrollo de LocMaBar. Costo estimado: 700€, amortizado en 3 años (233€/año).

•**Dispositivo de prueba**: Xiaomi Redmi 12C 3/64GB Gris, con Android 12, procesador MediaTek Helio G85, y pantalla de 6.71 pulgadas. Este dispositivo representa un equipo de gama media-baja, ideal para probar la compatibilidad de LocMaBar con dispositivos Android de uso común (API 21+). Costo: 170€, amortizado en 3 años (56,60€/año).

**Horas de Trabajo:**

•**Desarrolladores**: 2 desarrolladores a 19€/h, con un total de 200 horas (100 horas cada uno). Las tareas incluyen el desarrollo de las actividades principales, la integración con Firebase, y la implementación de geolocalización y geocoding. Coste total: 3.800€. La tarifa de 19€/h se basa en el promedio para desarrolladores junior en España

•Diseñador: 1 diseñador a 16€/hora, con 10 horas para crear las interfaces en Figma. Coste total de 160€.

•**Administrador**: 1 administrador a 20€/h, con 30 horas para gestionar tareas de validación manual y coordinar el equipo mediante Trello y Discord. Coste total: 600€.

**Amortización:**

Los costes del **hardware** se amortizarán linealmente en 3 años. El coste amortizado del portátil, 233€/año, y el dispositivo móvil es de 56,60€/año, lo que suman un total de 289,60€ anuales en hardware. Para **software**, el único coste directo es del Geocoding, con aproximadamente unos 60€/año para 1000 solicitudes mensuales. Las horas de trabajo suman un total de 3960€. Sumando todos los costes, tanto de hardware como software para el primer año se obtiene un total de **4599.20€**.

## Otros detalles de planificación

Además de los costes directos, se han considerado otros gastos relacionados con el proyecto, como:

**Equipo de Proyecto**

El equipo está formado por:

2 desarrolladores: Encargados de programar la app, incluyendo las funciones de búsqueda y gestión de datos.

1 diseñador: Responsable de crear los prototipos de la interfaz en Figma.

1 administrador: Coordina al equipo y revisa el contenido subido por los usuarios, como las propuestas de nuevos bares.

**Herramientas de gestión**

GitHub: Para gestionar el código y controlar las versiones.

Trello: Para organizar las tareas y seguir el progreso del proyecto.

Discord: Para comunicación y reuniones del equipo.

## Seguimiento y control del desarrollo

El seguimiento y control del desarrollo de LocMaBar se llevó a cabo utilizando Trello y Discord, enfocándose en la gestión del proyecto y la calidad del resultado final:

>Gestión de Tareas: Trello ha sido utilizado para asignar tareas, como el diseño de pantallas o la programación de las distintas funciones de la aplicación.

>Colaboración en Equipo: Discord permite comunicación constante en tiempo real, añadir comentarios, notificar cambios.

>Informes y Análisis: Se han generado informes de progreso y se ha analizado el rendimiento para identificar áreas donde se puede mejorar e identificar retrasos, asegurando así que el proyecto cumple con los plazos establecidos.

>Control de Calidad: Se han ralizado pruebas básicas en cada etapa (como probar el inicio de sesión o la búsqueda de bares) para garantizar que la app funcionara correctamente y cumpliera con los requisitos definidos, como tiempos de respuesta rápidos y una interfaz fácil de usar.

# Aplicación desarrollada

## Manuales

### i. Manual de usuario

Este manual está destinado a los usuarios finales de LocMaBar, proporcionando instrucciones claras sobre cómo utilizar las funciones y características de la aplicación.

**Contenido:**

**Introducción a la aplicación:**

“LocMaBar” es una aplicación móvil diseñada para facilitar el descubrimiento de bares y restaurantes locales, especialmente en pequeños municipios de España. Permite buscar bares por provincia y municipio, encontrar opciones cercanas mediante geolocalización, consultar información detallada, dejar valoraciones y proponer nuevos locales.

**Requisitos del sistema:**

●**Hardware**: Dispositivo móvil Android con conexión a internet.

●**Software**: Versión de Android 5.0 (API 21) o superior.

**Guía de instalación y configuración inicial:**

●**Acceso a la aplicación**: LocMaBar estará disponible en Google Play Store (pendiente de publicación). Busca "LocMaBar", descarga e instala la app.

●**Registro de usuario**: Abre la app, selecciona "Registrarse" e introduce tu nombre, correo y contraseña para crear una cuenta.

●**Configuración inicial**: Inicia sesión con tu email y contraseña. La app pedirá permisos de geolocalización la primera vez que busques bares cercanos; selecciona "Aceptar".

**Navegación por la interfaz de usuario:**

●**Interfaz principal:** La pantalla inicial muestra en primer lugar, un mensaje de autorización de permisos de ubicación. Pero también muestra varios menús desplegables de búsqueda para seleccionar provincia y municipio, un botón para buscar bares cercanos, y una barra de navegación en la parte inferior.

●**Secciones principales:** Búsqueda de locales, lista de los locales cercanos, propuestas para nuevos locales y gestión de cuenta e información personal.

**Descripción de funcionalidades clave:**

●**Registro e inicio de sesión:** Selecciona "Registrarse", ingresa tus datos (nombre, correo, contraseña) y confirma. Para iniciar sesión, usa tu email y contraseña.

●**Búsqueda por provincia y municipio:** Selecciona provincia y municipio en los menús desplegables y pulsa "Buscar" para ver una lista de bares.

**●Búsqueda por geolocalización:** Acepta los permisos de ubicación y visualiza los bares cercanos, cargando estos automáticamente.

●**Dejar valoraciones:** En los detalles de un bar, ingresa una puntuación (1-5 estrellas) y un comentario, y envía.

●**Proponer nuevos locales:** En "Propuestas", completa el formulario y pulsa "Enviar".

●**Ver información detallada de un bar:** Selecciona un bar de los resultados para ver su información más detalladamente.

●**Editar perfil:** En "Perfil", pulsa "Editar perfil", modifica los campos disponibles y confirma los cambios.

**Resolución de problemas comunes:**

●**Problemas de inicio de sesión:** Soluciones para contraseñas olvidadas, verifica tus credenciales y recuperación de cuentas, contacta con el soporte de la aplicación.

●**Errores de conexión:** Asegúrate de tener internet; reinicia la app si el problema persiste.

●**Búsqueda sin resultados:** La causa puede ser que sea que no existen locales cercanos o introduce bien los filtros de búsqueda.

●**Geolocalización no funciona:** Habilita los permisos de ubicación en ajustes del dispositivo.

### ii. Manual del desarrollador

Este manual está destinado a los desarrolladores que trabajarán en el mantenimiento y la ampliación de LocMaBar, proporcionando información sobre la arquitectura, el código y las prácticas de desarrollo.

**Contenido:**

**Descripción de la arquitectura del sistema:**

●**Modelo Cliente-Servidor:** LocMaBar sigue un modelo cliente-servidor con tres capas: la presentación con la interfaz de usuario, la lógica de negocio y la persistencia con el almacenamiento en firestore.

●**Diagrama de componentes:** Visualización de los componentes principales y sus interacciones.

**Instalación y configuración del entorno de desarrollo:**

●**Requisitos del entorno:** Hardware (Portátil con al menos 8 GB de RAM y conexión a internet.) y Software (Android Studio) necesarios para configurar el entorno de desarrollo.

●**Pasos de configuración:** Instrucciones para la instalación de Android Studio, configuración de un emulador o el uso de un dispositivo Android, la configuración de Firebase y sus reglas y la configuración de otras dependencias.

**Estructura del código fuente y organización del proyecto:**

●**Directorios principales:** Explicación de la estructura de carpetas y archivos dentro del proyecto.

●**Organización del código:** Descripción de cómo se organiza el código (MVC en el caso de LocMaBar) en módulos y clases.

**Explicación de los componentes principales:**

●Frontend: Descripción del uso de Jetpack Compose para la interfaz de usuario.

●Backend: Firebase Authentication para autenticación y Firestore para almacenamiento.

●Base de datos: Firestore con colecciones y relaciones entre las tablas.

**Patrones de diseño utilizados:**

●MVC (Modelo-Vista-Controlador): Explicación de cómo se implementa el patrón MVC dentro de la aplicación LocMaBar.

**Guía de desarrollo y contribución:**

●Agregar nuevas características: Crear nuevas pantallas en vista, añadir rutas de navegación, y actualiza Firestore si es necesario.

●Corrección de errores: Procedimiento para identificar, rastrear y corregir bugs.

●Control de versiones: Uso de herramientas como Git para el control de las versiones y colaboración.

**Pruebas y depuración:**

●Pruebas unitarias y de integración: Métodos y herramientas para la realización de pruebas.

●Depuración: Técnicas y herramientas para depurar el código y resolver problemas.

### iii. Manual de administrador

Este manual está destinado a los administradores del sistema que gestionarán la infraestructura de LocMaBar, proporcionando instrucciones sobre configuración y mantenimiento.

**Contenido:**

**Requisitos del servidor y del software:**

●Hardware: Dispositivo Android para pruebas (mínimo API 21).

●Software: Acceso a Firebase Console, navegador web.

**Proceso de instalación del servidor y configuración del entorno de producción:**

●Configuración de Firebase: Crear un proyecto en Firebase.google.com, habilitar Authentication y Firestore, y configurar las reglas de seguridad.

●Despliegue de la aplicación: Generar un APK en Android Studio y subir a Google Play Store (pendiente de publicación).

●Configuración de servicios externos: Habilitar Google Location Services, Geocoding API y AdMob en Google Cloud Console y Firebase.

**Gestión de usuarios y roles:**

●Creación y gestión de usuarios: Cómo crear, editar y eliminar cuentas de usuario.

●Asignación de roles: Procedimiento que se debe seguir para asignar roles y permisos a los usuarios.

**Mantenimiento de la base de datos:**

● Backups y restauración: Exportar datos de Firestore desde Firebase Console y realizar copias de seguridad semanales.

● Optimización: Métodos para optimizar el rendimiento de la base de datos.

**Seguridad del sistema:**

● Configuración de permisos: Configurar las reglas de seguridad en Firestore para los usuarios autenticados.

● Protección contra ataques: Medidas para proteger la aplicación contra ataques comunes como “SQL injection”.

**Supervisión del rendimiento y registro de actividades:**

● Monitoreo del servidor: Revisión de Firebase Authentication y Firebase Store.

● Registro de actividades: Cómo habilitar y revisar logs en Android Studio para el seguimiento

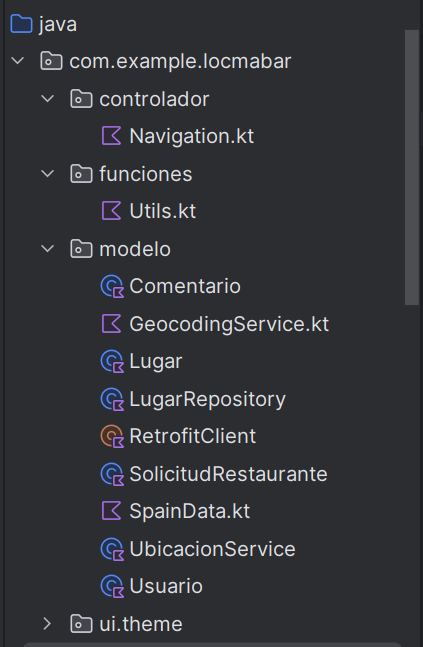
de actividades y resolución de problemas.

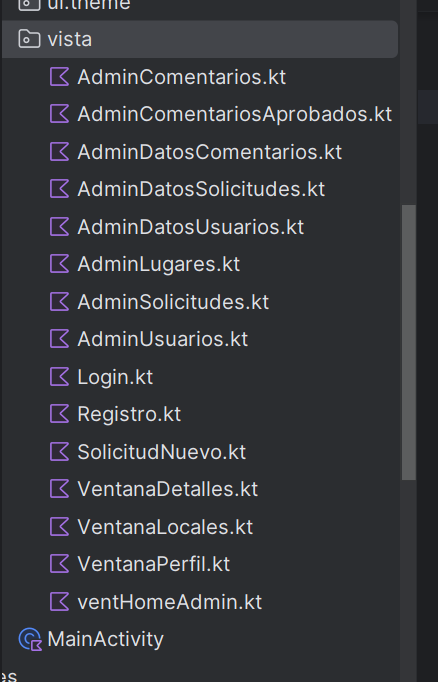
**Resolución de problemas y soporte técnico:**

● Problemas comunes: Soluciones para problemas frecuentes relacionados con el servidor y la aplicación.

● Soporte técnico: Cómo contactar y colaborar con el soporte técnico para resolver problemas más complejos.

# Archivo/s de la aplicación.





**Archivos de configuración:** Configuraciones necesarias para el funcionamiento de la aplicación LocMaBar.

●**AndroidManifest.xml:**

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />  
<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />  
<uses-permission android:name="android.permission.INTERNET" />

●**build.gradle.kts(app):**

*implementation*(platform("com.google.firebase:firebase-bom:33.12.0"))  
*implementation*("com.google.firebase:firebase-auth-ktx")

*implementation*("com.google.firebase:firebase-firestore-ktx")

*implementation*("com.google.android.gms:play-services-auth:21.3.0")  
*implementation*("com.google.android.gms:play-services-location:21.3.0")

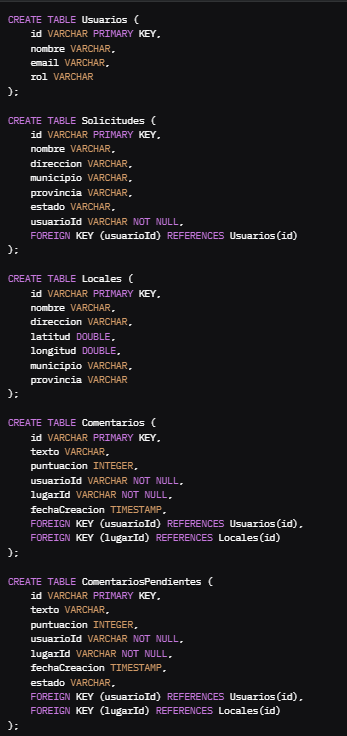
*implementation*("com.google.maps.android:maps-compose:4.3.3")

*implementation*("androidx.navigation:navigation-compose:2.7.3")

*implementation*("com.google.accompanist:accompanist-permissions:0.34.0")

*implementation*("org.jetbrains.kotlinx:kotlinx-coroutines-android:1.6.4")

Consultas la creación de las tablas necesarias (aunque en Firebase se hace de manera manual):



# Conclusiones

El desarrollo de LocMaBar ha sido un proceso completo y enriquecedor, abarcando desde la planificación inicial hasta la implementación final de una aplicación móvil que facilita el descubrimiento de bares y restaurantes locales. Este proyecto ha cumplido con los objetivos establecidos, demostrando su viabilidad y utilidad para usuarios y administradores en el contexto de pequeños municipios en España. A continuación, se detallan las principales conclusiones obtenidas:

**Facilitación del descubrimiento de bares locales:** LocMaBar ha demostrado ser una herramienta eficaz para descubrir bares y restaurantes, especialmente en pequeños municipios donde la visibilidad es mucho menor. Las funcionalidades de búsqueda por provincia y municipio, así como la búsqueda por geolocalización, permiten a los usuarios encontrar opciones relevantes de manera rápida y accesible.

**Mejora en la participación de los usuarios:** La implementación de funciones como la posibilidad de proponer nuevos locales y dejar valoraciones ha fomentado la participación activa de los usuarios, creando una comunidad donde los datos de los bares se enriquecen con las contribuciones de los propios usuarios. Esto ha mejorado la calidad y cantidad de información disponible, beneficiando tanto a los usuarios como a los negocios publicados en la aplicación.

**Accesibilidad y usabilidad:** La interfaz de usuario, diseñada con Jetpack Compose y prototipada en Figma, ofrece una experiencia intuitiva y fácil de usar para todo tipo de usuarios. La barra de navegación inferior y las pantallas principales aseguran una navegación fluida, mientras que el registro e inicio de sesión garantizan una experiencia personalizada y segura.

**Seguridad y privacidad:** Se han implementado medidas robustas para proteger los datos de los usuarios, cumpliendo con las normativas. Firebase Authentication asegura la gestión segura de cuentas con cifrado, y las reglas de seguridad de Firestore restringen el acceso a usuarios autenticados. Además, los permisos de geolocalización se solicitan de forma transparente, respetando la privacidad del usuario.

**Flexibilidad y escalabilidad:** La arquitectura basada en el patrón MVC y el uso de tecnologías modernas como Firebase y Kotlin permiten que LocMaBar sea flexible y adaptable a futuras mejoras. La estructura modular del código (con paquetes como vista y modelo) y la base de datos en Firestore facilitan la incorporación de nuevas funcionalidades.

# Propuestas de mejora

Aunque el objetivo principal de LocMaBar se ha cumplido con éxito, siempre se puede mejorar y evolucionar. A continuación, se presentan algunas propuestas de mejora que podrían considerarse para futuras mejoras del proyecto:

**Implementación de nuevas funcionalidades para usuarios:** Añadir opciones como reservas directas de mesas, notificaciones de ofertas y recomendaciones personalizadas basadas en preferencias (como "bares con música en vivo" o "bares con terraza"). Estas funcionalidades aumentarían el valor de la app para los usuarios, mejorarían la experiencia y podrían incrementar la visibilidad.

**Incorporación de imágenes de los locales:** Permitir que los usuarios y los locales bares suban imágenes de los locales y sus productos, mejorando la experiencia visual de la app. Las imágenes harían que la app sea más atractiva y ayudaría a los usuarios a tomar mejores decisiones a la hora de ir a un local u otro, especialmente en pequeños municipios donde la información visual suele ser limitada.

**Implementación de IA en la publicación de los contenidos:** Optimizar el proceso de validación manual de comentarios y solicitudes mediante herramientas de validación automática más avanzadas, como la inteligencia artificial. Reduciendo así la carga de trabajo a los administradores y mejorando la velocidad de aprobación del contenido.

**Colaboraciones con bares locales para incluir menús y promociones:** Llegar a un acuerdo con bares y restaurantes para que se puedan incluir menús diarios y promociones especiales directamente en la app. Esto beneficiaría a los usuarios al proporcionar información más detallada y actualizada, y a los negocios al aumentar su visibilidad. Por ejemplo, los bares podrían pagar una pequeña tarifa para destacar sus promociones, creando un modelo de ingresos adicional y consiguiendo así un beneficio ambas partes.

**Promoción de la sostenibilidad:** Incluir una sección en los detalles de los bares que resalte el uso de productos locales o prácticas sostenibles, ya que según el análisis de demanda realizado el 66% de los consumidores prefieren marcas sostenibles. Esto podría atraer un número mayor de usuarios y diferenciar a LocMaBar de sus competidores.

# Bibliografía

- [Agencia Española de Protección de Datos | AEPD](https://www.aepd.es/)

- [BOE.es - Agencia Estatal Boletín Oficial del Estado](https://www.boe.es/)

-https://www.bing.com/ck/a?!&&p=e0603e44dac48b58357507cef84b747aca18cc374e6feaf064a1d43f1022c82bJmltdHM9MTc0NDY3NTIwMA&ptn=3&ver=2&hsh=4&fclid=00551e7b-5f15-6bfb-2e7b-0b765efe6ac8&psq=cuanto+cobra+un+desarrollador+de+aplicaciones+moviles+por+hora+en+espa%c3%b1a&u=a1aHR0cHM6Ly93d3cua2l3aXJlbW90by5jb20vc3VlbGRvL2Rlc2Fycm9sbGFkb3ItZGUtYXBsaWNhY2lvbmVzLW1vdmlsZXMvIzp-OnRleHQ9VW4lMkZhJTIwRGVzYXJyb2xsYWRvciUyRmElMjBkZSUyMEFwbGljYWNpb25lcyUyME0lQzMlQjN2aWxlcyUyMGNvYnJhJTIwZGUlMjBtZWRpYSxlbXBsZW8lMjBwdWJsaWNhZGFzJTIwZW4lMjBLaXdpJTIwUmVtb3RvLiUyMCVDMyU5QWx0aW1hJTIwYWN0dWFsaXphY2klQzMlQjNuJTNBJTIwMjIlMkYwMSUyRjIwMjUu&ntb=1

- [HOSTELERÍA DE ESPAÑA | A tu servicio](https://cehe.es/)

- <https://www.huleymantel.com/barras-estrellas/suspenso-digitalizacion-bares-restaurantes-cafeterias-espana_100962_102.html>

- [Privacidad y seguridad en Firebase](https://firebase.google.com/support/privacy?hl=es-419)

- [Récord de turismo en España 2024: éxito para la hostelería | Last.app](https://www.last.app/recursos/blog/record-turismo-espana-2024-exito-bares-hoteles-restaurantes#:~:text=Espa%C3%B1a%20ha%20logrado%20cifras%20r%C3%A9cord%20en%20turismo%20en,significativo%20de%20visitantes%20internacionales%20y%20el%20gasto%20tur%C3%ADstico.)

- <https://www.bing.com/ck/a?!&&p=5f97f07e9f18f9933a17210e296b32d68a501e2061de2cef6d4751247441000aJmltdHM9MTc0NzAwODAwMA&ptn=3&ver=2&hsh=4&fclid=00551e7b-5f15-6bfb-2e7b-0b765efe6ac8&psq=uso+del+movil+en+espa%c3%b1a&u=a1aHR0cHM6Ly9ta3RlZmEuZGl0cmVuZGlhLmVzL2Jsb2cvaW5mb3JtZS1tb2JpbGUtMjAyMA&ntb=1>

-https://www.bing.com/ck/a?!&&p=3dd4432ed30672b79147309f22d6256f918053a137c6de85ba7a72ef4221a6ffJmltdHM9MTc0Nzg3MjAwMA&ptn=3&ver=2&hsh=4&fclid=00551e7b-5f15-6bfb-2e7b-0b765efe6ac8&psq=kiwi+remoto+sueldo+programadores+espa%c3%b1a+hora&u=a1aHR0cHM6Ly93d3cua2l3aXJlbW90by5jb20vc3VlbGRvL2FuYWxpc3RhLXByb2dyYW1hZG9yLyM6fjp0ZXh0PSVDMiVCRkN1JUMzJUExbnRvJTIwZ2FuYSUyMHBvciUyMGhvcmElM0YlMjBVbiUyRmElMjBBbmFsaXN0YSUyMFByb2dyYW1hZG9yJTJGYSUyMGNvYnJhLGVtcGxlbyUyMHB1YmxpY2FkYXMlMjBlbiUyMEtpd2klMjBSZW1vdG8uJTIwJUMzJTlBbHRpbWElMjBhY3R1YWxpemFjaSVDMyVCM24lM0ElMjAwMyUyRjA1JTJGMjAyNS4&ntb=1

-https://www.bing.com/ck/a?!&&p=8364dc7d45d4f086f045d2b6ccf6260accc6dc3d62135c504cc750638223af7bJmltdHM9MTc0ODEzMTIwMA&ptn=3&ver=2&hsh=4&fclid=00551e7b-5f15-6bfb-2e7b-0b765efe6ac8&psq=que+porcentaje+de+personas+en+espa%c3%b1a+prefieren+marcas+sostenibles&u=a1aHR0cHM6Ly9tYXJjYXMuZWxlY29ub21pc3RhLmVzL29yYW5nZS9ub3RpY2lhcy8xMTM5NDY4OS8wOS8yMS9Mb3MtY29uc3VtaWRvcmVzLWVzdGFuLWRpc3B1ZXN0b3MtYS1wYWdhci1tYXMtcG9yLXByb2R1Y3Rvcy1zb3N0ZW5pYmxlcy5odG1sIzp-OnRleHQ9QXNpbWlzbW8lMkMlMjB1biUyMGluZm9ybWUlMjBkZWwlMjBJbnN0aXR1dG8lMjBkZSUyMEludmVzdGlnYWNpJUMzJUIzbiUyMGRlLGElMjBtYXJjYXMlMjBtZW5vcyUyMGNvbm9jaWRhcyUyMHBlcm8lMjBxdWUlMjBzb24lMjBzb3N0ZW5pYmxlcy4&ntb=1

# Anexos

## Presentación PowerPoint

## Video de presentación aplicación