



# DOCUMENTO FUNCIONAL DEL PROYECTO

ALUMNO/GRUPO: Arcides Manuel Mateo Terrero

---

## 1. Introducción y contexto

En la actualidad, muchas personas tienen dificultades para organizar sus tareas, recordar citas importantes o mantener un equilibrio entre el estudio, el trabajo y la vida personal. La falta de planificación puede generar estrés, pérdida de tiempo y olvidos que afectan al rendimiento y la productividad diaria.

El proyecto tiene como objetivo desarrollar una aplicación de gestión de tareas y recordatorios, pensada para estudiantes, trabajadores y cualquier persona que necesite llevar un control eficiente de sus actividades.

El usuario final será una persona que desee planificar su día a día, establecer recordatorios, recibir notificaciones y visualizar sus tareas de forma clara mediante un calendario interactivo. También se busca que el diseño sea sencillo para el usuario.

La solución propuesta consiste en una aplicación moderna que combine las funciones clásicas de un gestor de tareas con características como un calendario interactivo, notificaciones y la posible integración de funciones avanzadas. Con ello se pretende mejorar la organización personal y reducir la carga mental de los usuarios, ofreciendo una herramienta práctica, intuitiva y personalizable.

---

## 2. Análisis de requisitos

### 2.1. Requisitos funcionales (RF)

Los requisitos funcionales describen qué debe hacer el sistema, es decir, las funciones principales que los usuarios podrán realizar.

Código	Descripción del requisito funcional
RF1	El sistema permitirá registrar nuevos usuarios correo electrónico y contraseña segura.
RF2	Los usuarios podrán iniciar sesión con su correo y contraseña registrados.
RF3	El sistema permitirá cerrar sesión en cualquier dispositivo.
RF4	Los usuarios podrán editar su perfil básico (nombre y foto de perfil).
RF5	El sistema permitirá crear nuevas tareas con: título, descripción, fecha límite, hora y prioridad.
RF6	Los usuarios podrán editar cualquier información de las tareas creadas.
RF7	Los usuarios podrán eliminar tareas (con confirmación simple).
RF8	Las tareas tendrán 3 estados posibles: "Pendiente", "En Progreso" y "Completada".
RF9	El sistema mostrará las tareas en vista de lista ordenable por orden de creación.
RF10	Los usuarios podrán filtrar tareas por categoría, prioridad o estado.
RF11	Los usuarios podrán ordenar tareas por 3 niveles de prioridad.
RF12	Los usuarios podrán buscar tareas por palabras clave en título o descripción.
RF13	Los usuarios podrán crear, editar y eliminar categorías personalizadas.

RF14	Las categorías tendrán colores asignables para identificación visual.
RF15	Los usuarios podrán asignar categorías a las tareas.
RF16	Los usuarios podrán cambiar entre tema claro y oscuro.
REQUISITOS OPCIONALES (Fase 2)	
RF17	Los usuarios podrán recuperar su contraseña mediante enlace enviado al correo electrónico.
RF18	El sistema permitirá crear tareas recurrentes (diarias, semanales, mensuales).
RF19	Los usuarios tendrán acceso a un calendario interactivo con vista mensual.
RF20	El sistema permitirá restaurar tareas eliminadas desde la papelera (hasta 30 días).
RF21	Los usuarios podrán exportar sus tareas a formato CSV.
RF22	El sistema enviará notificaciones automáticas según la fecha/hora de la tarea.
RF23	Las tareas se sincronizarán entre todos los dispositivos del usuario.
RF24	El sistema realizará sincronización en la nube
FUNCIONALIDADES AVANZADAS (FASE 3)	
RF25	Los usuarios podrán compartir listas de tareas con otros usuarios
RF26	El sistema permitirá permisos diferenciados: "Solo lectura" o "Lectura y escritura".

RF27	Los usuarios podrán asignar tareas a otros usuarios en proyectos compartidos.
RF28	El sistema se integrará con Google Calendar para importar eventos.
RF29	Los usuarios podrán crear tareas mediante comandos de voz.
RF30	El sistema mostrará estadísticas de productividad.

---

## 2.2. Requisitos no funcionales (RNF)

Los requisitos no funcionales indican cómo debe comportarse el sistema, incluyendo aspectos de rendimiento, seguridad, compatibilidad y facilidad de uso.

Código	Descripción del requisito no funcional
RNF1	La interfaz será accesible desde dispositivos móviles.
RNF2	Las páginas deberán cargarse en menos de tres segundos.
RNF3	Los datos de usuarios y tareas estarán cifrados y seguros.
RNF4	La aplicación debe ser intuitiva y fácil de usar, con navegación clara.
RNF5	Las notificaciones deberán entregarse de forma fiable y oportuna según el usuario lo haya programado.
RNF6	El backend no debe sufrir degradación del rendimiento.

---

## 2.3. Restricciones

Condiciones o limitaciones del proyecto.

#### Lenguajes o tecnologías obligatorias:

- Kotlin/Supabase backend.
- Kotlin/xml para fronten.

#### Recursos disponibles:

- Tiempo: duración del proyecto acorde al calendario del curso.
- Equipo: desarrollo individual, sin equipo de trabajo adicional.
- Materiales: ordenador con software necesario (Android studio, Trello, Supabase, GitHub).

#### Dependencias o limitaciones técnicas:

- La aplicación debe funcionar en dispositivos Android.
- La IA para sugerencias o reconocimiento de voz es opcional y puede requerir APIs externas.
- Limitación de rendimiento para no superar la carga máxima de usuarios simultáneos según entornos de pruebas.

---

### 3. Análisis de usuarios y roles

El objetivo de este apartado es identificar los tipos de usuarios que utilizarán el sistema, describir sus necesidades dentro de la aplicación **Memento**, y definir los permisos y acciones que cada uno podrá realizar.

**Memento** está diseñado para ofrecer distintos niveles de acceso según el tipo de usuario, garantizando seguridad, organización y control sobre los datos almacenados.

Rol	Descripción	Permisos principales
Usuario registrado	Persona que utiliza Memento para gestionar su tiempo, crear tareas y compartirlas con otros usuarios. Toda la información se sincronizará en la	<ul style="list-style-type: none"><li>- Crear, editar, eliminar y marcar tareas como completadas.</li><li>- Organizar tareas por categorías, prioridad o proyectos.</li></ul>

	nube, permitiendo acceder desde cualquier dispositivo.	<ul style="list-style-type: none"> <li>- Establecer recordatorios y recibir notificaciones.</li> <li>- Sincronizar sus datos automáticamente en la nube.</li> <li>- Compartir tareas con otros usuarios, otorgando permisos de lector (solo visualiza) o editor (puede modificar).</li> <li>- Consultar su historial y progreso general.</li> <li>- Personalizar su perfil y preferencias.</li> </ul>
--	--	---

### Necesidades principales:

Poder gestionar todas sus tareas desde una misma aplicación.

Acceder a sus datos desde cualquier dispositivo mediante sincronización en la nube.

Compartir tareas con otros usuarios, asignando permisos según el tipo de colaboración(opcional).

Recibir notificaciones y recordatorios de forma puntual.

Contar con una interfaz sencilla e intuitiva.

Disponer de un sistema seguro que proteja su información personal y sus tareas.

## 4. Casos de uso / Escenarios de uso

Mostrar cómo interactúa el usuario registrado con Memento y cuáles son los resultados esperados de sus acciones principales.

Código	Nombre del caso de uso	Actor principal	Descripción	Resultado esperado
CU1	Registrar usuario	Usuario	El usuario introduce su correo, nombre y contraseña para crear una cuenta en Memento.	Cuenta creada correctamente y datos guardados en la nube.

CU2	Iniciar sesión	Usuario	El usuario introduce su correo y contraseña para acceder a la aplicación.	Acceso concedido y sincronización de datos con la nube realizada.
CU3	Crear tarea	Usuario	El usuario añade una nueva tarea, asignando título, descripción, fecha, hora y categoría.	Tarea registrada correctamente y visible en la lista de tareas y calendario.
CU4	Editar o eliminar tarea	Usuario	El usuario selecciona una tarea existente y la modifica o elimina según sea necesario.	Cambios reflejados en la lista de tareas y sincronizados en la nube.
CU5 (Opcional)	Compartir tarea	Usuario	El usuario comparte una tarea con otro usuario, otorgando permisos de lector o editor.	Otro usuario puede ver o editar la tarea según los permisos asignados.
CU6	Recibir recordatorio	Usuario	El usuario recibe una notificación a la hora establecida para una tarea.	Usuario notificado correctamente y recordatorio cumplido.
CU7	Consultar historial y progreso	Usuario	El usuario revisa tareas completadas, pendientes y estadísticas de progreso.	Visualización de historial y métricas de productividad actualizadas.

## 5. Modelo de datos o estructura de la información

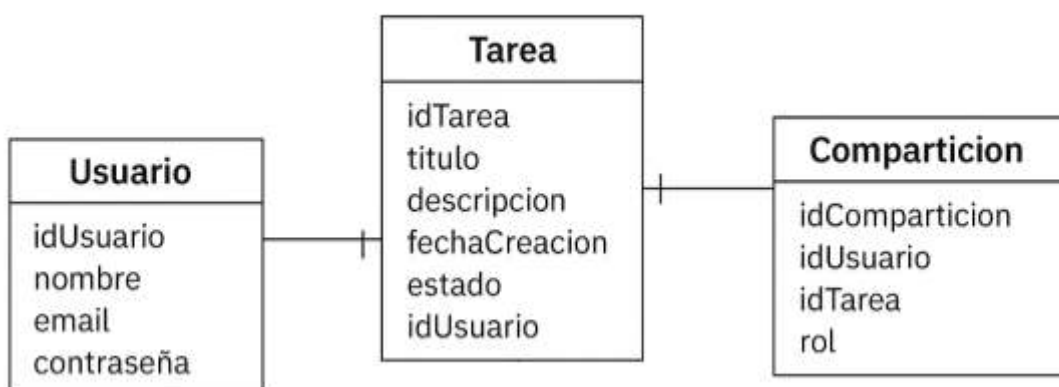
Representar la información que gestionará **Memento**, incluyendo entidades principales, sus atributos y relaciones. Esto servirá para diseñar la base de datos en PostgreSQL y para comprender cómo se interrelacionan los datos.

Entidades principales	Descripción	Campos principales	Relaciones
-----------------------	-------------	--------------------	------------

Usuario	Representa al usuario registrado que accede a la app.	idUsuario, nombre, email, contraseña, fechaRegistro	1 a N con Tarea (un usuario puede crear varias tareas).
Tarea	Actividad o elemento de trabajo que el usuario crea o comparte.	idTarea, titulo, descripcion, fechaCreacion, estado, idUsuario	N a N con Usuario (si se comparte como lector/editor).
Progreso	Guarda el avance o estado de una tarea.	idProgreso, idTarea, porcentaje, ultimaActualizacion	1 a 1 con Tarea.
Comparticion (Opcional)	Define los permisos de acceso a tareas compartidas.	idComparticion, idUsuario, idTarea, rol (lector/editor)	Relaciona Usuario y Tarea.
Nube	Representa la sincronización de datos del usuario.	idNube, idUsuario, ultimaSincronizacion	1 a 1 con Usuario.

“El campo idUsuario representa al creador de la tarea. Los usuarios adicionales con acceso se gestionan mediante la tabla Compartición.”

Diagrama de ER:


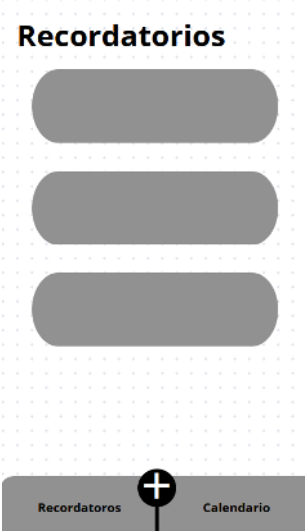


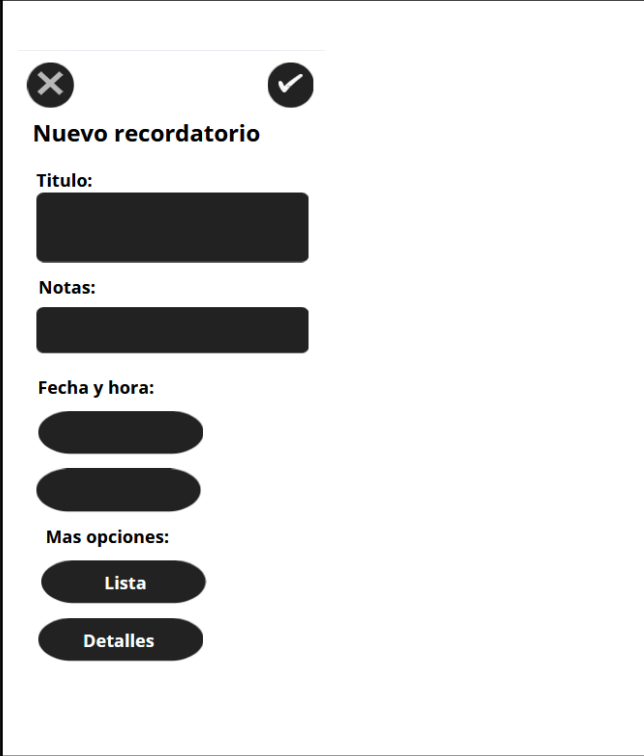
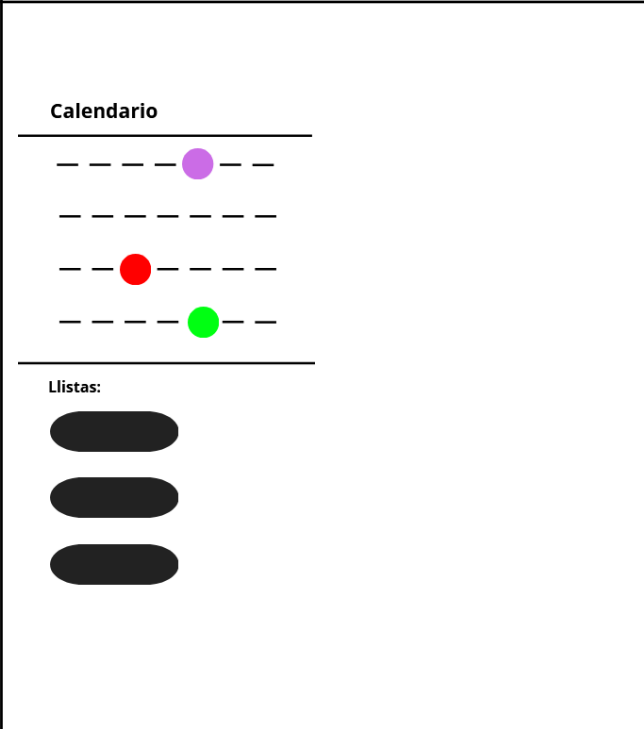
Compartición: En caso de que se llegue a completar dicha funcionalidad\*

---

## 6. Diseño de la interfaz

Visualizar la estructura y navegación del sistema **Memento** antes de su desarrollo. Este apartado muestra las pantallas principales, su funcionalidad y los casos de uso asociados, facilitando la comprensión del flujo de la aplicación.

Mockups	Descripción
	<p>Permite al usuario acceder a la aplicación mediante su correo electrónico y contraseña. También incluye la opción de registrarse o recuperar la contraseña. Presenta un diseño limpio y un fondo minimalista.</p> <p><b>Funcionalidad principal:</b> Autenticación segura del usuario y acceso al panel principal.</p> <p><b>Casos de uso relacionados:</b></p> <p>Iniciar sesión.</p> <p>Crear una nueva cuenta.</p> <p>Recuperar contraseña.</p>
	<p>Es el centro de control del usuario, donde puede ver un resumen de sus tareas activas, progreso general y accesos rápidos al calendario, perfil y configuración. Incluye una barra de navegación inferior para moverse entre secciones.</p> <p><b>Funcionalidad principal:</b> Visualizar y gestionar las tareas de forma general.</p> <p><b>Casos de uso relacionados:</b></p> <p>Ver listado de tareas.</p> <p>Acceder a la gestión o creación de una tarea.</p>

 <p><b>Nuevo recordatorio</b></p> <p>Título: [Redacted]</p> <p>Notas: [Redacted]</p> <p>Fecha y hora: [Redacted]</p> <p>Mas opciones:  <input type="button" value="Lista"/>  <input type="button" value="Detalles"/></p>	<p>Permite crear nuevas tareas, asignarles nombre, descripción, fecha, prioridad y colaboradores. También ofrece opciones para editar o eliminar tareas existentes, y compartirlas con otros usuarios (lector o editor).</p> <p><b>Funcionalidad principal:</b></p> <p>Crear, modificar y compartir tareas.</p> <p><b>Casos de uso relacionados:</b></p> <p>Crear tarea, Editar tarea, Eliminar tarea. Compartir tarea con otros usuarios.</p>
 <p><b>Calendario</b></p> <p>--- -- ● ---  --- -- ---  --- ● ---  --- -- ● ---</p> <p>Listas:  [Redacted]  [Redacted]  [Redacted]</p>	<p>Muestra las tareas organizadas por fecha en un calendario mensual o semanal. Los usuarios pueden hacer clic en un día específico para ver las tareas pendientes o finalizadas.</p> <p><b>Funcionalidad principal:</b></p> <p>Visualizar las tareas según su fecha y gestionar la planificación.</p> <p><b>Casos de uso relacionados:</b></p> <p>Consultar tareas por día o semana.  Marcar tareas como completadas.  Revisar progreso temporal de proyectos.</p>

## 7. Planificación técnica

**Objetivo:** planificar el desarrollo del proyecto.

### Lenguajes y frameworks:

- **Kotlin** (Android Studio): lenguaje principal para el desarrollo de la aplicación Android.
- **XML**: para el diseño de las interfaces y disposición visual.
- **Supabase**: backend como servicio (PostgreSQL, autenticación, sincronización en tiempo real).
- **Room**: persistencia local offline.
- **Hilt**: inyección de dependencias.

### Base de datos:

- **Supabase (PostgreSQL)** como sistema principal de gestión de datos en la nube.
- **Room** como base de datos local para modo offline.
- Tablas principales: Usuario, Tarea, Compartición (opcional), Estadísticas, Nube.

### Herramientas de diseño o edición:

- Figma / Canva: para crear los mockups y pantallas de la app.
- [Draw.io](https://draw.io) / Lucidchart: para elaborar diagramas ER y de flujo.

### Reparto de tareas:

(Proyecto individual)

- **Diseño y planificación**: definición de estructura, diseño visual y modelo de datos.
- **Desarrollo**: implementación de pantallas, navegación y lógica de tareas.
- **Integración**: conexión con base de datos y sincronización en la nube.
- **Pruebas y revisión**: detección y corrección de errores antes del despliegue.

Fase	Descripción	Duración estimada
<b>Planificación y análisis</b>	Estudiar apps similares, definir funcionalidades originales, especificar requisitos técnicos y de usuario.	1,5 meses
<b>Diseño (arquitectura + interfaz de usuario)</b>	Diseño de la base de datos, diagramas ER/UML, mockups y estructura de navegación.	1 mes

<b>Desarrollo (programación de la app)</b>	Implementación de funcionalidades principales: login, gestión de tareas, calendario interactivo, estadísticas, recordatorios, recuperación de contraseña.	3,5 meses
<b>Pruebas y correcciones</b>	Pruebas de usabilidad, corrección de errores, optimización del rendimiento y validación en distintos dispositivos Android.	1 mes
<b>Memoria y documentación</b>	Redacción de la memoria final, manual de usuario, anexos técnicos y preparación de la presentación.	1 mes
<b>Defensa del proyecto</b>	Preparar la exposición oral, diapositivas y demostración práctica de la aplicación.	1 mes

## 8. Análisis de riesgos

Durante el desarrollo del proyecto **Memento** pueden aparecer diversos riesgos que afecten al cumplimiento de los objetivos, la calidad del producto o los plazos establecidos. A continuación se describen los principales riesgos identificados y las medidas previstas para reducir su impacto.

### 8.1. Identificación de riesgos

- **Falta de tiempo o mala planificación:**  
Existe el riesgo de que el desarrollo se retrase por una mala organización o acumulación de tareas. Para evitarlo, se seguirá el cronograma definido y se priorizarán las funcionalidades más importantes, revisando los avances semanalmente.
- **Problemas técnicos o incompatibilidades:**  
Pueden surgir errores al integrar diferentes tecnologías, como Android, PostgreSQL o librerías externas. Se reducirá este riesgo utilizando versiones estables, realizando pruebas por módulos y documentando correctamente las dependencias y configuraciones.
- **Pérdida de datos o errores en la base de datos:**  
Un fallo durante la sincronización con la nube o en las operaciones de guardado podría provocar pérdida de información. Para prevenirlo, se realizarán sincronizaciones frecuentes con la nube y se implementarán mecanismos de validación y control de errores en la base de datos.

- **Falta de conocimientos en alguna tecnología:**

El uso de tecnologías o herramientas nuevas, pueden generar dificultades técnicas. Para afrontarlo, se dedicará tiempo a la formación previa, se consultará la documentación oficial y se buscará apoyo en comunidades técnicas.

- **Fallo en la sincronización con la nube:**

Existe la posibilidad de que los datos no se actualicen correctamente entre dispositivos. Se implementarán comprobaciones automáticas y copias locales de respaldo hasta que se confirme la sincronización completa.

## 8.2. Valoración y respuesta

Clasifica cada riesgo según su probabilidad e impacto, e indica cómo se mitigará.

Riesgo	Descripción	Probabilidad	Impacto	Plan de acción
Falta de tiempo o mala planificación	No cumplir con las fechas de entrega por mala organización o imprevistos.	Media	Alta	Crear un calendario con tareas semanales y revisar el progreso regularmente.
Problemas técnicos o incompatibilidades	Fallos en el software, errores de código o versiones incompatibles.	Media	Media	Probar frecuentemente el código y usar control de versiones.
Pérdida de datos	Borrado accidental o fallo.	Baja	Alta	Mantener sincronizaciones frecuentes y verificar la integridad.
Cambios en los requisitos	Modificaciones en las funciones o en el diseño.	Media	Media	Mantener buena comunicación con el profesor/cliente y registrar los cambios

## 9. Validación y criterios de éxito

**Objetivo:** definir cómo sabremos que el proyecto funciona correctamente.

### Criterios de aceptación:

- La aplicación permite crear, editar y eliminar recordatorios sin errores.
- Los datos se guardan correctamente y se sincronizan con la nube.
- La interfaz es clara, fácil de entender y accesible desde distintos dispositivos.
- Las notificaciones se muestran en el momento adecuado.
- No se producen bloqueos ni pérdida de información.

### Pruebas previstas:

- **Pruebas funcionales:** verificar que cada función (crear, editar, borrar, sincronizar, notificar) funciona correctamente.
- **Pruebas de usuario:** realizar pruebas con usuarios reales para comprobar la facilidad de uso y la comprensión de la interfaz.
- **Pruebas de rendimiento:** asegurar que la app carga rápido y responde sin retrasos.
- **Pruebas de sincronización (opcional):** confirmar que los datos se actualizan correctamente entre dispositivos.

### Indicadores de calidad:

- Más del 90% de las pruebas funcionales superadas sin errores.
- Tiempos de carga inferiores a 2 segundos.
- Valoración de los usuarios superior a 8/10 en facilidad de uso.
- Sin pérdida de datos en las pruebas de sincronización.

## 10. Conclusión

*Objetivo:* cerrar el análisis y preparar la siguiente fase.

El análisis funcional del proyecto **Memento** ha permitido definir de forma detallada los objetivos, funcionalidades y la estructura técnica de la aplicación.

El sistema se centrará en ofrecer una herramienta práctica y moderna para la **gestión personal de tareas**, orientada a usuarios que buscan una forma sencilla de organizar su tiempo, planificar actividades y mejorar su productividad desde un dispositivo Android.

### Funciones principales del sistema

La aplicación contará con varias funciones clave que marcarán su funcionamiento general:

- **Pantalla de inicio y login:** permitirá al usuario acceder de forma segura mediante autenticación, garantizando la privacidad de sus datos.
- **Gestión de tareas:** ofrecerá la posibilidad de crear, editar, eliminar y marcar tareas como completadas, añadiendo detalles como fechas límite o prioridades.
- **Calendario de tareas:** mostrará una vista ordenada de las tareas por días o semanas, facilitando la planificación y el seguimiento del progreso.
- **Compartición de tareas:** permitirá enviar o compartir tareas con otros usuarios, fomentando la colaboración en actividades comunes (opcional).
- **Panel principal:** servirá como centro de control, mostrando un resumen de las tareas activas, próximas y completadas.

### Tecnologías utilizadas

Para el desarrollo se empleará **Android Studio** como entorno principal, utilizando **Kotlin** para la lógica del programa y **XML** para el diseño de las interfaces.

La información se almacenará en la nube mediante **Supabase** (PostgreSQL) y en local con **Room**.

Además, se emplearán herramientas de apoyo como Figma o Canva para el diseño de los mockups y la interfaz de usuario, y **GitHub** para el control de versiones y organización del código.

### Valor y aportación del proyecto

El proyecto **Memento** busca combinar **simplicidad y utilidad** en una interfaz atractiva y accesible. Su principal valor radica en ofrecer a los usuarios una experiencia fluida para **organizar tareas diarias y mejorar la gestión del tiempo** sin necesidad de herramientas complejas.

Además, su diseño modular permitirá futuras ampliaciones, como estadísticas de productividad, recordatorios inteligentes o integración con asistentes virtuales.

### Próximos pasos

Con el análisis completado, se inicia la transición hacia la fase de desarrollo. Los siguientes pasos serán:

1. **Preparar el entorno de desarrollo** en Android Studio y configurar los emuladores necesarios.

2. **Diseñar la estructura inicial del repositorio** en GitHub y establecer la base del proyecto.
3. **Configurar la base de datos y las conexiones** con la app.
4. **Implementar los casos de uso prioritarios**, empezando por el inicio de sesión y la creación de tareas.
5. **Realizar pruebas iniciales** de funcionamiento y diseño para garantizar una buena experiencia de usuario.

### **Reflexión final**

El análisis funcional ha proporcionado una visión clara, realista y bien estructurada del proyecto. Gracias a esta planificación previa, el desarrollo podrá realizarse de forma organizada y eficiente, reduciendo riesgos y asegurando la calidad del resultado final.

**“Con una base sólida y un enfoque definido, el proyecto Memento está preparado para pasar a la fase de desarrollo y convertir las ideas en una aplicación funcional.”**