TidyTask

Proyecto: Andrea Catalán Menacho

---

1. Motivación

A lo largo del ciclo formativo de Desarrollo de Aplicaciones Multiplataforma he aprendido a crear soluciones tecnológicas útiles para el día a día. Uno de los problemas que he experimentado personalmente es la dificultad para organizarme y recordar tareas importantes. Por ello, me pareció interesante desarrollar una aplicación de gestión de tareas que no solo me ayudara a mantener el control de mis actividades, sino que también pudiera servir a otras personas con necesidades similares.

Este proyecto es una oportunidad para aplicar todo lo aprendido en los dos años de formación: diseño de interfaces, uso de bases de datos locales, gestión de notificaciones, y programación en Kotlin. Con este proyecto funcional y útil espero demostrar mi progreso y cerrar este ciclo.

---

2. Objetivos

Objetivo general:

Desarrollar una aplicación Android que permita a los usuarios gestionar sus tareas diarias de forma intuitiva y eficiente, incluyendo funcionalidades como creación, edición, categorización y recordatorios mediante notificaciones.

Objetivos específicos:

- Diseñar una interfaz amigable y funcional utilizando XML en Android Studio.
- Implementar una base de datos local con SQLite para guardar las tareas de forma persistente.
- Permitir al usuario crear, editar, eliminar y clasificar tareas por prioridad.
- Incorporar notificaciones push a través de AlarmManager para recordar al usuario sus tareas pendientes.

- Aplicar buenas prácticas en programación y diseño de aplicaciones móviles Android.

---

3. Tecnologías y herramientas utilizadas

En el desarrollo de este proyecto se emplearán las siguientes tecnologías y herramientas:

- **Lenguaje de programación**: **Kotlin**, el lenguaje oficial recomendado por Google para el desarrollo de aplicaciones Android.
- **Diseño de interfaces**: **XML**, utilizado para definir la estructura visual de la aplicación.
- **Entorno de desarrollo**: **Android Studio**, el IDE oficial de Google para Android, ya familiar tras su uso en diversas prácticas a lo largo del segundo curso.
- **Base de datos local**: **SQLite**, para el almacenamiento y gestión de las tareas creadas por el usuario.
- **Gestión de notificaciones y recordatorios**: **AlarmManager**, una clase de Android que permite programar eventos futuros y viene ideal para recordar tareas.
- **Gestión visual de tareas**: **Trello**, como soporte a la organización del proyecto, permitiendo visualizar el progreso por fases.
- **Diseño previo de interfaces**: se utilizará **Figma** para crear los primeros bocetos (mockups) de la aplicación y planificar la experiencia del usuario.

Estas herramientas han sido elegidas en base a su compatibilidad con Android y a la experiencia adquirida durante el ciclo formativo, buscando desarrollar un proyecto realista que consolide los conocimientos que he ido adquiriendo.

---

4. Estimación de recursos y planificación

Como estoy compaginando el desarrollo del proyecto con mis prácticas en empresa, he tenido que organizarme bien para poder avanzar cada semana. He planificado las tareas en base a mi disponibilidad por las tardes y fines de semana, dividiendo el proyecto en fases y asignándolas a semanas concretas.

Me he basado en la metodología Scrum, adaptándola a mi ritmo individual. Utilizaré Trello para gestionar visualmente las tareas y tener una idea clara de en qué punto me encuentro en cada momento.

A continuación, muestro la planificación semanal con las tareas previstas. Este cronograma me ayudará a mantener un ritmo constante y cumplir con los plazos establecidos:

SEMANA	TAREAS
Semana 1 17-30 de Marzo	- Escribir la motivación y los objetivos del proyecto. - Elegir las herramientas que voy a usar: Kotlin. - Preparar los requisitos y el diseño de la base de datos. - Hacer un primer boceto de cómo quiero que se vea la app en Figma. - Terminar la documentación.
Semana 2 31 Marzo- 6 de abril	- Revisar lo entregado por si hay algo que mejorar. - Organizar el proyecto en Android Studio.
Semana 3 7 - 13 de abril	-Registro de usuario Programar lo básico: que se puedan crear tareas. - Persistencia de las tareas en la base de datos - Visualización de los elementos correctamente en la interfaz.
Semana 4 14- 20 de abril	- Añadir opción de editar y borrar tareas. - Cambios visuales para que quede más bonito. - Revisar errores como campos vacíos o cosas que puedan fallar.
Semana 5 21-27 de abril	-Organizar las tareas por categorías o prioridades.

	- Implementar las notificaciones para recordar
Semana 6 28 abril- 4 de Mayo	- Pruebas de todo lo que llevo hecho. - Corregir errores y dejar la interfaz más pulida.
Semana 7 5-11 de Mayo	- Escribir la memoria del proyecto con capturas y explicaciones. - Asegurar de que esté todo documentado y bien presentado.
Semana 8 12-19 de Mayo	- Revisar todo antes de entregarlo. - Preparación de la defensa del proyecto.

Diagrama de Grantt

En el siguiente diagrama se representa de forma visual la secuenciación y duración de las tareas anteriormente descritas.

La planificación ha sido adaptada a mi ritmo de trabajo, compaginando el desarrollo del proyecto con las prácticas en empresa.

5. Análisis

A continuación, detallo los **requisitos funcionales** y **no funcionales** de la aplicación, que servirán como base para el desarrollo posterior. Estos requisitos me han ayudado a tener una visión más clara de lo que necesito implementar y cómo debe comportarse la app para cumplir su objetivo.

Requisitos funcionales:

1. **Registrar nuevos usuarios** desde un formulario con nombre de usuario y contraseña.
2. **Iniciar sesión** con los datos del usuario registrado.
3. **Crear tareas** con título, descripción, fecha, hora y prioridad.
4. **Guardar las tareas en una base de datos local (SQLite)** para que se mantengan tras cerrar la app.

5. **Visualizar la lista de tareas** de forma ordenada, mostrando sus detalles.
6. **Editar tareas existentes** para cambiar la información cuando se desee.
7. **Eliminar tareas** cuando ya no se necesiten.
8. **Organizar tareas por prioridad** (Alta, Media, Baja).
9. **Recibir notificaciones de recordatorio** mediante AlarmManager, en la fecha y hora indicada para cada tarea.

Requisitos no funcionales:

1. **Interfaz amigable y visualmente clara**, con colores suaves y estructura intuitiva.
2. **Accesibilidad** básica: botones grandes, texto legible y disposición sencilla.
3. **Funcionamiento sin conexión a internet**, al guardar todo localmente.
4. **Rendimiento adecuado**, sin demoras notables al crear o cargar tareas.
5. **Compatibilidad mínima** con versiones recientes de Android.
6. **Seguridad mínima** al no permitir campos vacíos en el registro o al crear tareas.
7. **Notificaciones eficientes**, que se activen aunque la app esté cerrada (mediante AlarmManager).

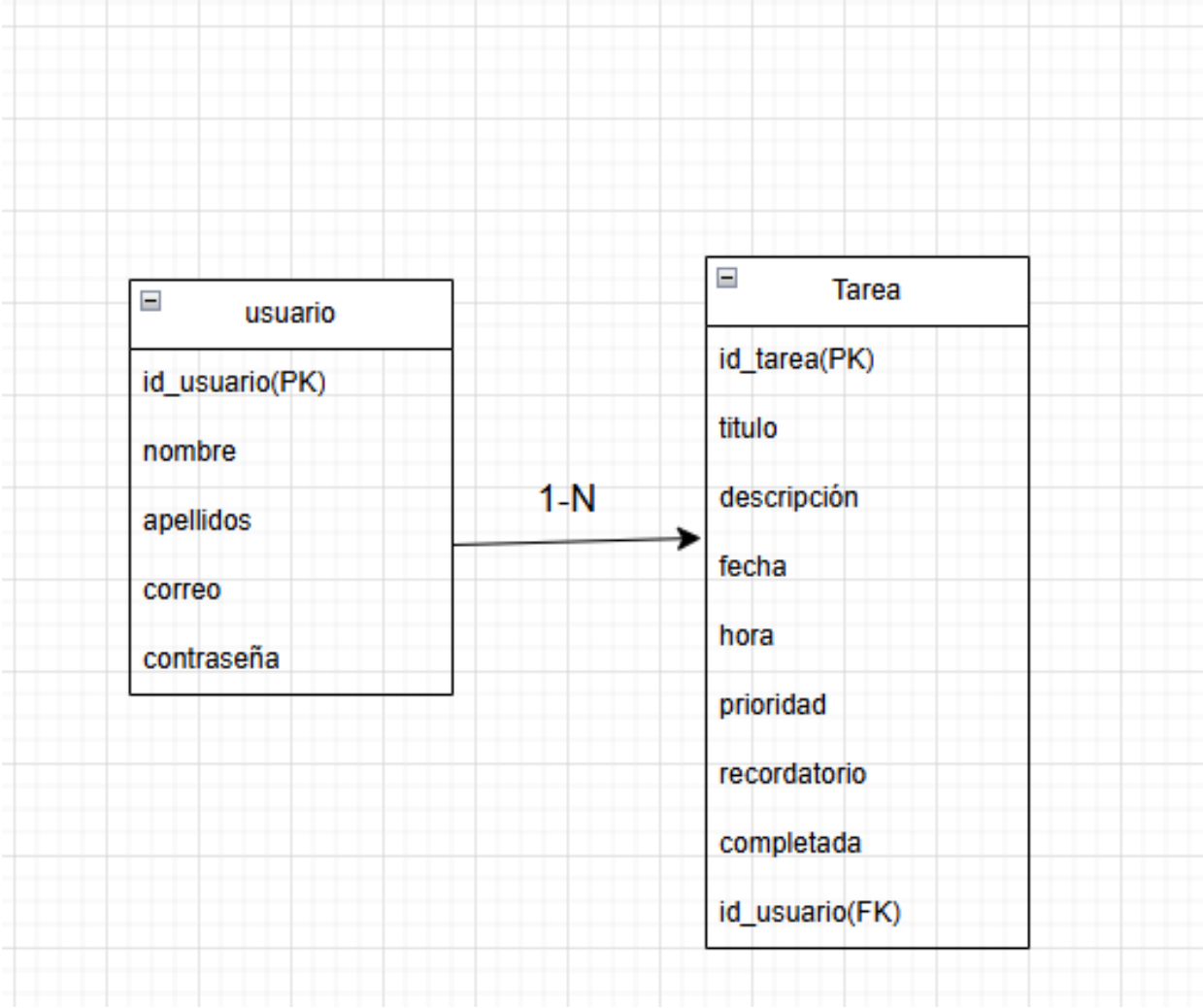
Modelo entidad-Relación:

El modelo Entidad-Relación de la aplicación se compone de dos entidades principales: **Usuario y Tarea**. Cada usuario puede tener varias tareas asociadas, por lo tanto, la relación entre ambas entidades es de uno a muchos (1:N).

En la entidad Usuario se almacenan los datos necesarios para iniciar sesión y gestionar las tareas de forma personalizada.

La entidad Tarea contiene toda la información relativa a cada actividad: título, descripción, fecha, hora, prioridad, si está completada, si tiene recordatorio, y un campo que la vincula con el usuario correspondiente.

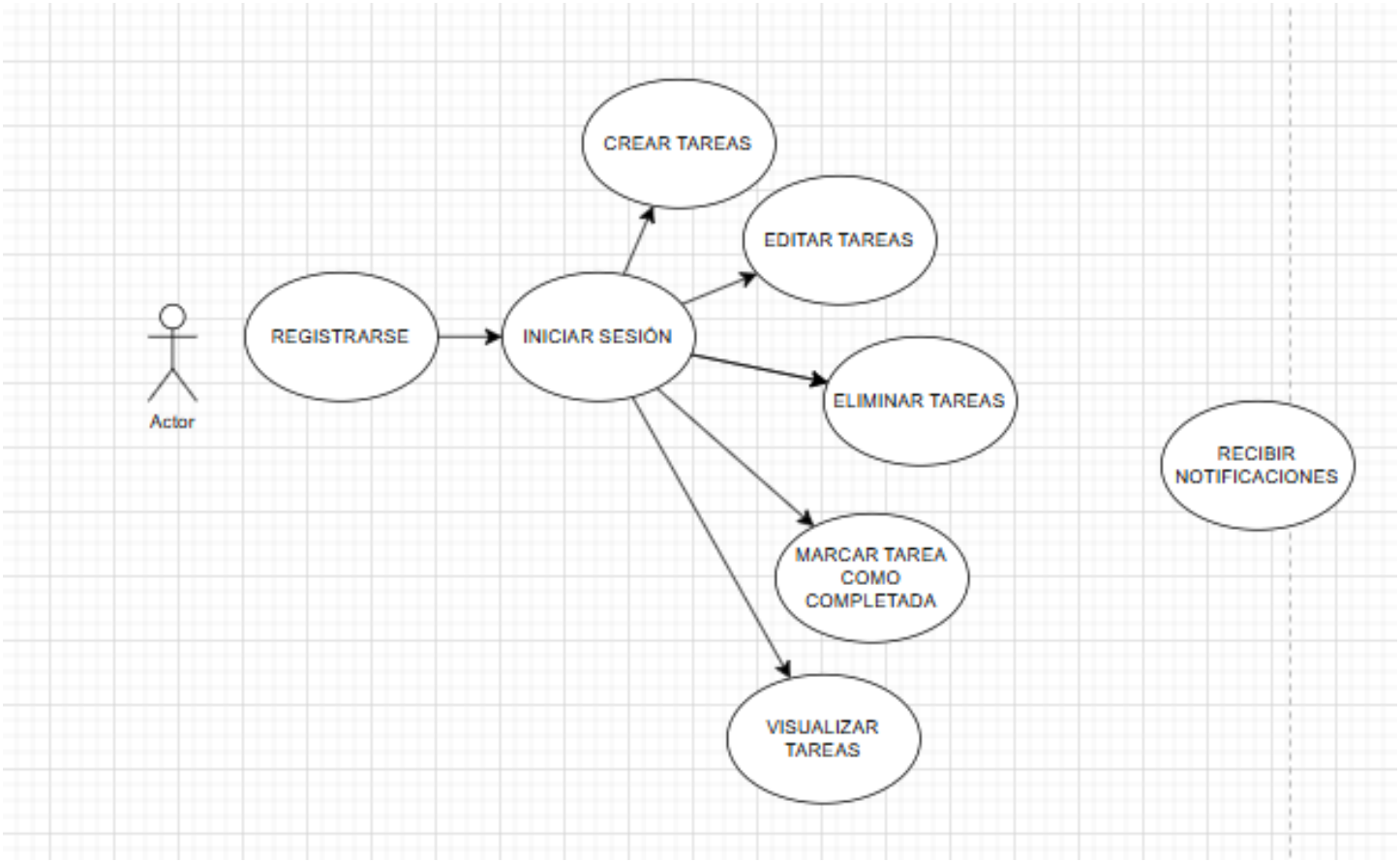
Con esta estructura se cubre todas las funcionalidades de la aplicación, desde el registro de usuarios hasta la gestión de las tareas con persistencia en base de datos.



CASOS DE USO:

El siguiente diagrama representa los principales casos de uso de la aplicación **TidyTask**. El usuario puede registrarse, iniciar sesión y gestionar sus tareas.

Entre las funcionalidades disponibles se encuentran la creación, visualización, edición, eliminación y marcado de tareas como completadas. Además, el sistema se encarga de generar notificaciones mediante AlarmManager para recordar al usuario las tareas pendientes.

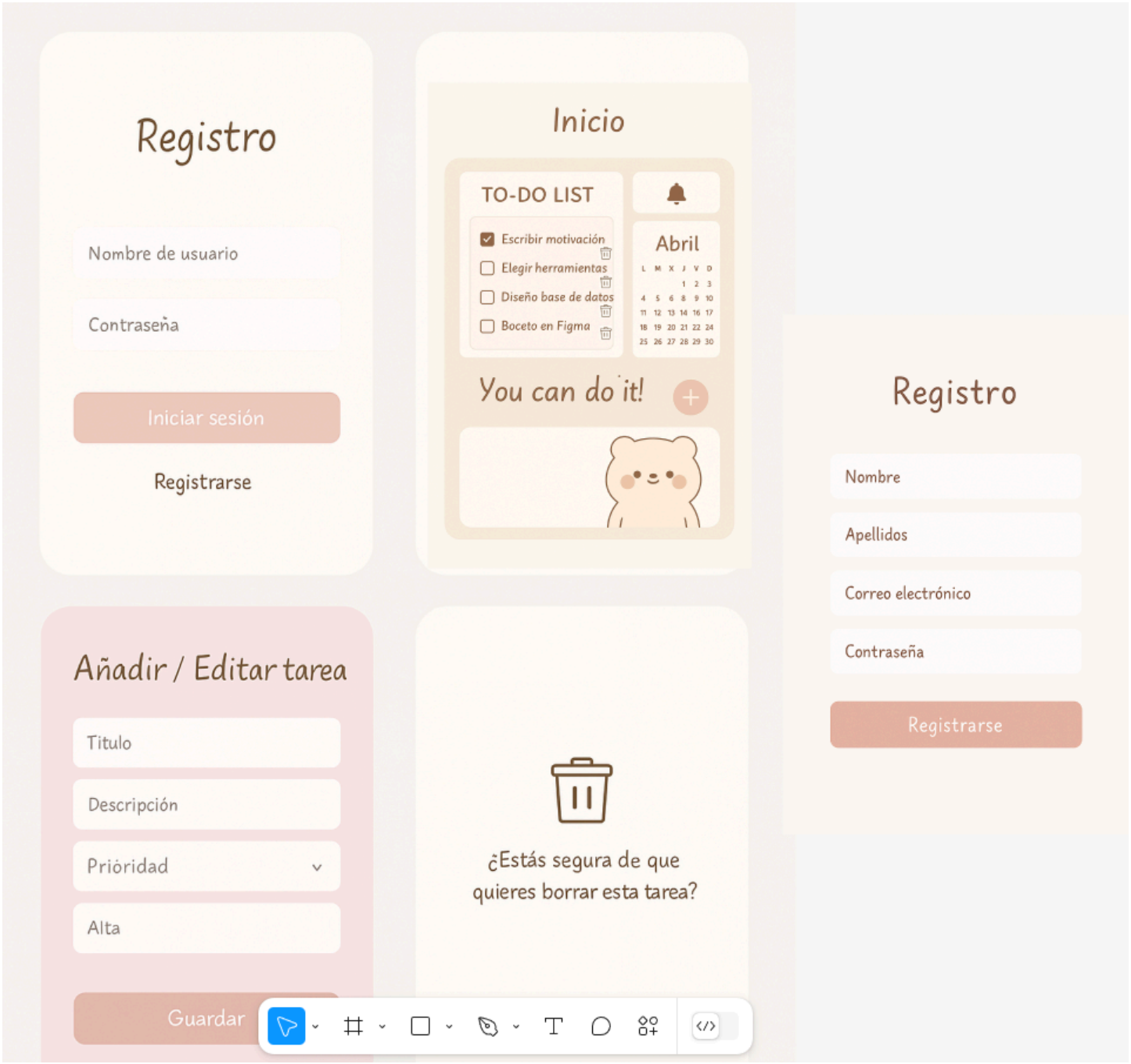


---

6. Diseño

Para el diseño de mi aplicación he querido centrarme en que sea visualmente clara, agradable y fácil de usar. Me inspiré en estilos tipo planner digital con colores suaves y estructura limpia, porque me transmiten calma y organización, que es justo lo que quiero que el usuario sienta al usar la app.

Antes de empezar a programar, elaboré varios mockups en Figma que representan las pantallas principales: el registro, el inicio con la lista de tareas y el calendario, el formulario para crear o editar una tarea, y la confirmación de borrado.



Pantalla de registro de usuario

Esta es la pantalla inicial donde el usuario puede registrarse. He incluido campos básicos como nombre, apellidos, correo y contraseña. El diseño sigue la línea estética del resto de la app, con colores suaves y un estilo limpio.

Pantalla de inicio / lista de tareas

Esta pantalla es el centro principal de la aplicación. Aquí el usuario puede ver sus tareas, organizadas por fecha, con opción para marcarlas como completadas (check) o eliminarlas (icono de papelera). También se muestra un calendario y una campanita para representar los recordatorios. Todo está integrado dentro de un contenedor tipo “planner visual” que refuerza la idea de organización y armonía.

Pantalla de añadir / editar tarea

Esta pantalla permite crear una nueva tarea o modificar una existente. El usuario podrá introducir el título, descripción, fecha, hora y prioridad. He querido que esta vista sea lo más intuitiva posible, dejando el foco en los campos importantes botones que llevan a cabo la acción.

Pantalla de confirmación de borrado

Esta pequeña ventana aparece cuando el usuario intenta borrar una tarea. Sirve como medida de seguridad para evitar eliminaciones accidentales.
