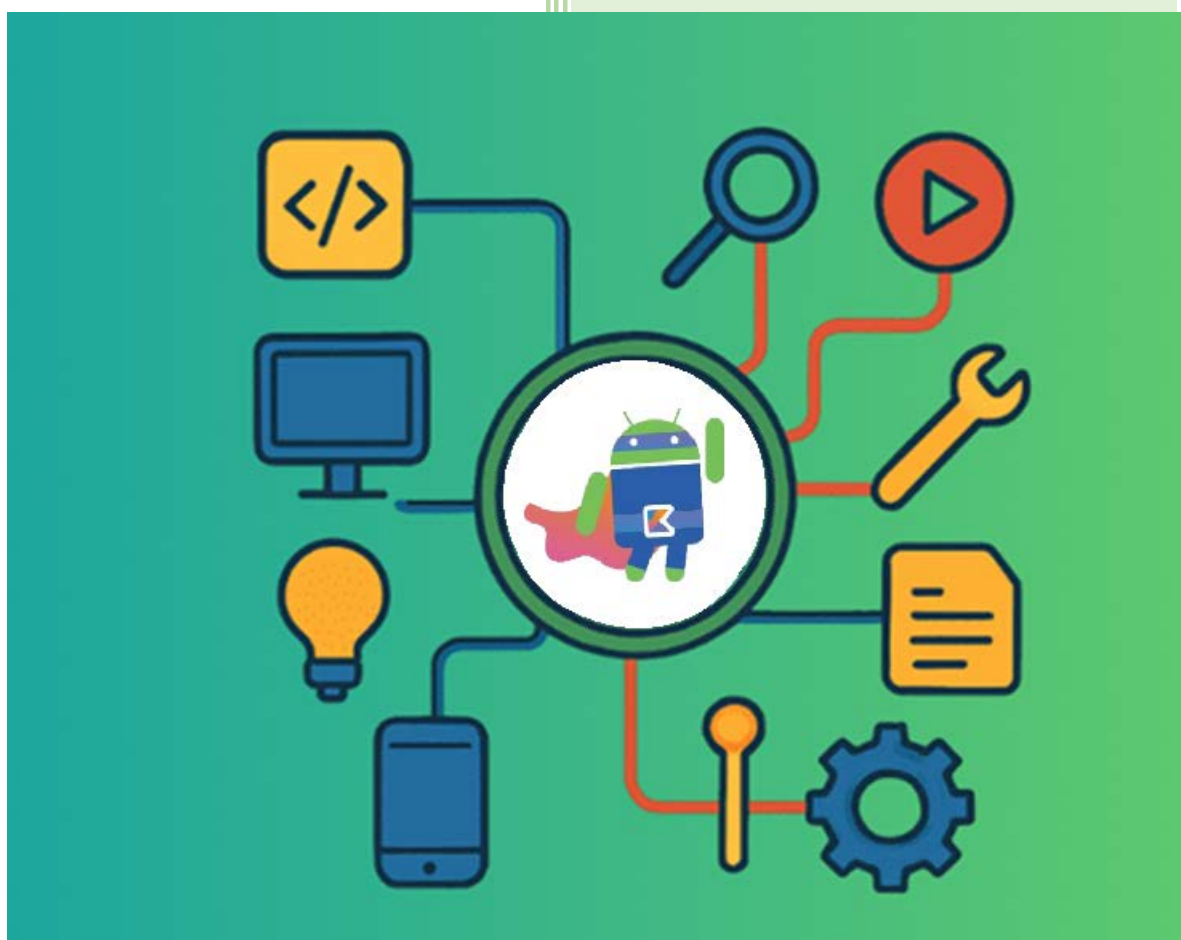


PROYECTO 2ª Evaluación



Programación de móviles y
servicios multimedia

Febrero 2026

El proyecto consiste en diseñar y desarrollar una aplicación nativa en Android utilizando Kotlin. La temática de la aplicación es totalmente libre (gestión de inventario, biblioteca de libros, recetario de cocina, gestión de gastos, lista de tareas, etc.).

Para que el proyecto sea considerado válido, debe implementar como mínimo los siguientes puntos:

1. Base de Datos (Room)

- La base de datos debe contener mínimo 3 tablas (Entidades) relacionadas entre sí.
- Uso de Foreign Keys para mantener la integridad referencial.

2. Gestión de Usuarios y Seguridad

- Pantalla de Login. Debe validar usuario y contraseña contra la base de datos.
- Registro: Posibilidad de dar de alta nuevos usuarios.
- **EXTRA:** Intentar usar cifrado en las contraseñas de los usuarios en vez de que éstas se guarden en texto plano. Podéis tratar de implementar un mecanismo de cifrado básico (ej. Hashing con SHA-256) antes de guardarla en la BD y al compararla en el login.

3. Interfaz Gráfica y Navegación

- La app debe contar con al menos 4 Activities distintas (sin incluir los posibles fragmentos que creéis).
- Manejo de Imágenes: Uso de imágenes estáticas desde la carpeta Drawables (ej: logotipos, iconos de menú).
- **EXTRA:** Uso de imágenes dinámicas NO almacenadas en Drawables. Deben permitir al usuario o bien seleccionar una foto de la galería (URI), hacer una foto con la cámara, o cargar una imagen desde una URL de internet (usando librerías como Glide o Coil si se desea).

4. Consultas y lógica de los datos

- Implementar consultas DAO que vayan más allá del simple "Select All".
- La app debe permitir ordenar los listados (ej: por fecha, por nombre A-Z) o filtrar por categoría/propiedad.

5. Diseño y Estilo

- La aplicación debe tener una identidad visual propia (colores coherentes, uso de estilos y temas en styles.xml o Material Design). No serán válidos los diseños "por defecto" sin personalizar.

6. Flujo de navegación.

El flujo mínimo esperado es:

- ↓ Login: Entrada a la app.
- ↓ Home: Pantalla principal donde se ve el resumen o listado principal.
- ↓ Alta/Edición: Pantalla para introducir nuevos registros (con selectores de fecha, texto, desplegables, etc.).
- ↓ Detalle: Pantalla para ver toda la información de un elemento específico.

De forma general, se valorará:

- Limpieza del Código. Código bien indentado, nombres de variables descriptivos y **comentarios** explicando las partes complejas de la lógica.
- Estructura del Proyecto. Separación correcta de paquetes
- Implementación de elementos no explicados en clase, como exportar datos a PDF o CSV, uso de los sensores del móvil, ...

Entrega

- Fichero comprimido (.zip) del proyecto (recordad limpiar la carpeta build y .gradle antes de comprimir).
- Breve documento PDF explicando:
 - Diagrama de las tablas de la base de datos.
 - Instrucciones de usuario de prueba (User/Pass) para corregir la práctica.

Presentación

Al terminar la app, tendréis 15 minutos para convencer a la clase (y al profesor) de que vuestra app es útil, amigable y robusta. En la exposición debéis presentar vuestra aplicación, realizar una demostración de su funcionamiento y responder a las preguntas que surjan sobre la implementación técnica.

Guía de evaluación

| CATEGORÍA | CRITERIOS DETALLADOS | PUNTOS MÁX. | NOTA |
|--------------------|---|-------------|------|
| 1. BASE DE DATOS | <ul style="list-style-type: none"> • Implementación Room: Uso correcto de la librería, DAOs y Database. • Estructura: Mínimo 3 Tablas (Entidades) relacionadas. • Integridad: Uso correcto de claves foráneas (Foreign Keys) y persistencia real de datos. | 2.0 | |
| 2. FUNCIONALIDAD | <ul style="list-style-type: none"> • Login/Registro: Funcional y validado contra la BD. • Consultas: Existen filtros u ordenación de listas (no solo "Select All"). • Estabilidad: La app no se cierra inesperadamente (nose produce un crash). | 2.5 | |
| 3. INTERFAZ (UI) | <ul style="list-style-type: none"> • Navegación: Mínimo 4 Activities con flujo lógico (Intents). • Imágenes: Uso de recursos estáticos (Drawables). • Diseño: Estilo propio y cuidado (colores, temas), evitando el diseño por defecto. | 2.0 | |
| 4. CÓDIGO | <ul style="list-style-type: none"> • Limpieza: Código estructurado (paquetes data/ui), indentado y legible. • Comentarios: Explicación clara de las partes complejas de la lógica. • Buenas Prácticas: Uso de Corrutinas para no bloquear la UI. | 1.5 | |
| 5. EXPOSICIÓN ORAL | <ul style="list-style-type: none"> • Comunicación: Claridad, vocabulario técnico y seguridad al hablar. • Demo: Demostración en vivo fluida y sin errores. • Defensa: Capacidad de respuesta ante preguntas técnicas sobre el código. • Tiempo: Ajuste al límite de 15 minutos. | 2.0 | |
| TOTAL | SUMA DE PUNTOS (hasta 10) | 10.0 | |
| EXTRA (BONUS) | <ul style="list-style-type: none"> • Seguridad: Las contraseñas se guardan cifradas (Hash) y no en texto plano. • Imágenes: carga dinámica (Cámara, Galería o URL). • Innovación: Sensores, Exportación (PDF/CSV) o UI avanzada (Animaciones/Dark Mode). | +1.0 | |
| NOTA FINAL | (Suma del Total + Bonus) | | |
| OBSERVACIONES: | | | |