

ESTRATEGIA DE PRUEBAS

EQUIPO 29

INDICE

1. APLICACIÓN BAJO PRUEBAS	1
1.1. NOMBRE APLICACIÓN.....	1
1.2. VERSIÓN	1
1.3. DESCRIPCIÓN	1
1.4. FUNCIONALIDADES CORE:	1
1.5. DIAGRAMA DE ARQUITECTURA:	2
<i>Modelo de componentes (vista en capas)</i>	2
1.6. DIAGRAMA DE CONTEXTO:	2
1.7. MODELO DE DATOS	3
<i>Diagrama de clases.</i>	3
<i>Modelo de despliegue</i>	3
1.8. MODELO DE GUI:	4
2. CONTEXTO DE LA ESTRATEGIA DE PRUEBAS	4
2.1. OBJETIVOS:	4
2.2. DURACIÓN DE LA ITERACIÓN DE PRUEBAS:	5
2.3. PRESUPUESTO DE PRUEBAS:	5
<i>Recursos Humanos</i>	5
<i>Recursos Computacionales</i>	5
<i>Recursos Económicos para la contratación de servicios/personal:</i>	5
2.4. TNT (TÉCNICAS, NIVELES Y TIPOS) DE PRUEBAS:	5
2.5. DISTRIBUCIÓN DE ESFUERZO	6

1. Aplicación Bajo Pruebas

1.1. Nombre Aplicación

Vynilsapp

1.2. Versión

1.0

1.3. Descripción

Plataforma para compartir álbumes y canciones

1.4. Funcionalidades Core:

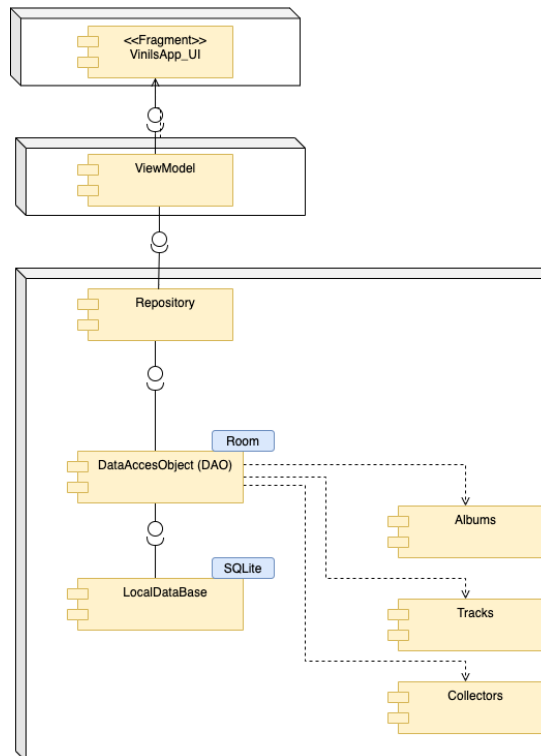
- Despliegue de álbumes por usuario
- Detalle de álbumes

Plantilla elaborada por

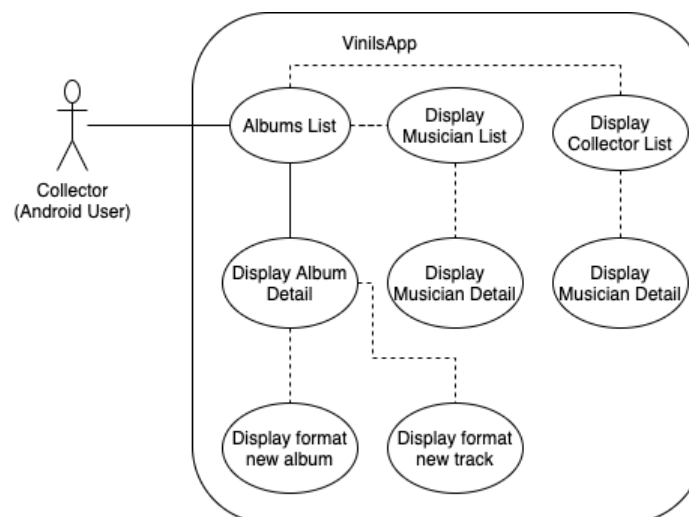
- Listado de artistas
- Listado de coleccionistas
- Información detallada de artistas
- Información detallada de coleccionistas
- Comentarios de álbumes
- Creación de nuevos álbumes
- Vinculación de canciones a álbumes

1.5. Diagrama de Arquitectura:

Modelo de componentes (vista en capas)

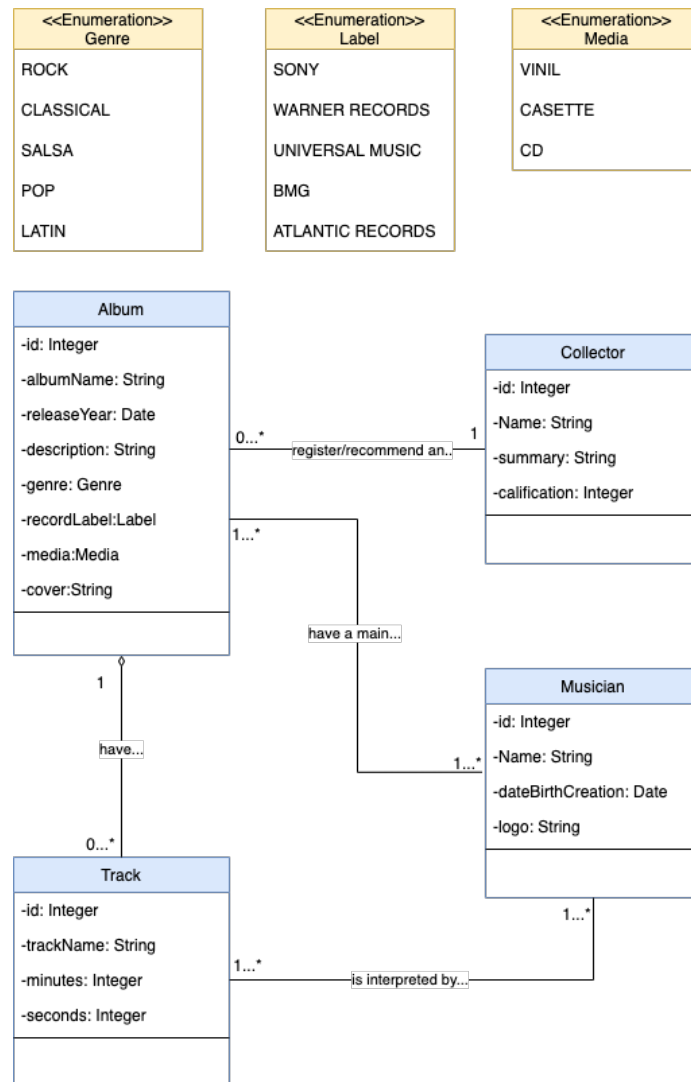


1.6. Diagrama de Contexto:

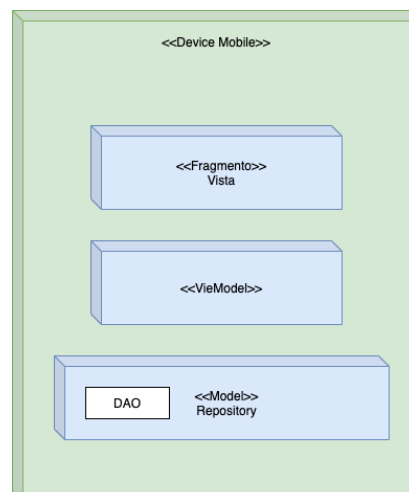


1.7. Modelo de Datos

Diagrama de clases.



Modelo de despliegue



1.8. Modelo de GUI:

En el siguiente link se puede ver el mockup navegable del modelo GUI:

<https://www.figma.com/proto/9IPJk9zU2OdZxMYON7dVbh/DesarrolloMovil?node-id=59%3A3&scaling=scale-down&page-id=0%3A1&starting-point-node-id=59%3A3&show-proto-sidebar=1>

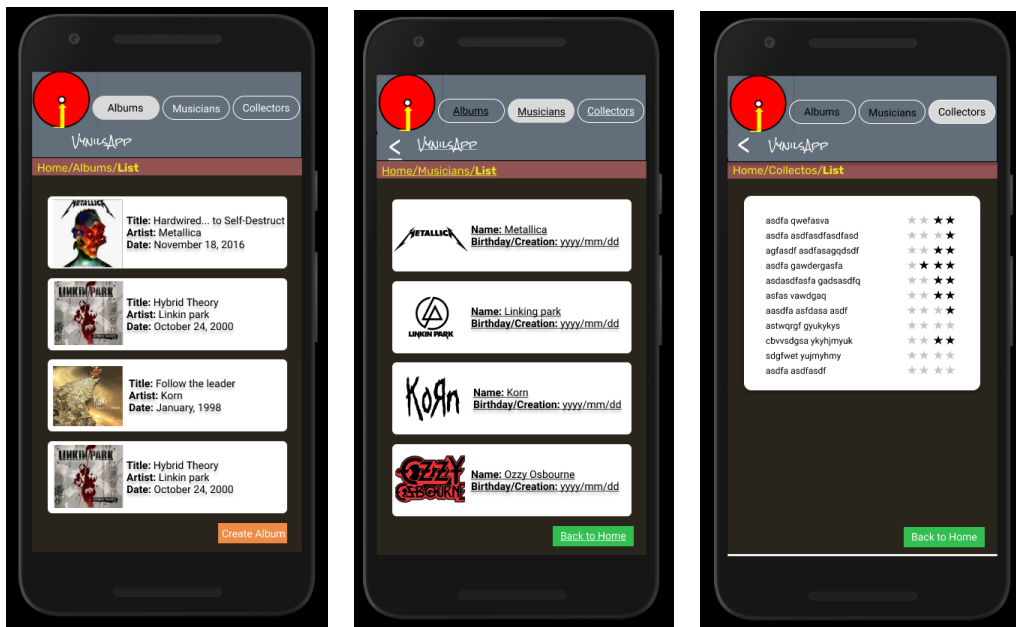


Imagen 1 – Vistas de las pantallas (activities) para Albums/Musicians/Collectors

2. Contexto de la estrategia de pruebas

2.1. Objetivos:

OBJETIVO
Validar el correcto funcionamiento de la interfaz grafica de la app móvil
Reconocer posibles debilidades que pudiera tener la interfaz grafica de la app móvil
Basado en un oráculo de pruebas, identificar que componentes gráficos pudieran tener acciones inesperadas

Validar el correcto funcionamiento de la app y sus funcionalidades -> API's de automatización (pruebas funcionales) y GUI Ripper (Pruebas no funcionales)

Plantilla elaborada por

2.2. Duración de la iteración de pruebas:

Pruebas manuales: 8 horas hombre

Pruebas automatizadas: (4 personas x 5 horas) 20 Horas hombre + 40 horas maquina

Las pruebas automatizadas se harán con los recursos cargados localmente (BDs, Back, etc.)

2.3. Presupuesto de pruebas:

Presupuesto: Recursos de Juniors de pruebas 28 horas hombre, con estimado de 40.000 pesos/hora, para un total de \$1.120.000

Recursos Humanos

Se consideran cuatro (4) ingenieros juniors para pruebas para un total de 28 horas/hombre

Dentro de las horas descritas arriba se requiere que el personal conozca las herramientas de pruebas disponibles para el proyecto, la preparación de los escenarios de pruebas y el alistamiento y planeación de los componentes de pruebas y del proyecto en general.

Cálculo del presupuesto basado en estimaciones de HH

Para la implementación y funcionamiento de pruebas se usarán recursos computacionales locales por lo tanto no hay un costo asociado a esto más allá de las horas hombre usadas.

Recursos Computacionales

Los recursos del outsourcing dependen de este servicio contratado, por lo que no se ofrece detalle de esto. El contrato debe establecer que los recursos sean suficientes para lograr los objetivos requeridos en el tiempo determinado.

- 1 PCs (Intel) con los recursos de software para pruebas (software libre y gratuito). 2 Maquinas Windows 10 y 2 MAC (M1)
- Como dispositivos de grabación de videos de pantalla se tiene la aplicación OBS Studio instalados en los PCs yy Zoom
- Github como parte de la data de prueba y repositorios de información

Restricciones:

- Solo sistemas Android (dispositivos físicos y virtuales). No Apple.
- Kotlin + Android Studio
- Pruebas con Espresso

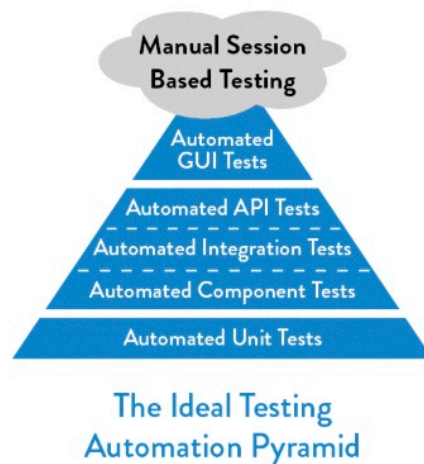
Recursos Económicos para la contratación de servicios/personal:

Para el caso particular de este presupuesto no se encuentra destinado recurso para contratación de outsourcing u otros servicios aparte de los ya especificados

2.4. TNT (Técnicas, Niveles y Tipos) de pruebas:

PRESUPUESTO	ESTRATEGIA	PATRON	TIPO	NIVEL	TECNICA	OBJETIVO
1	Automatizadas	Pirámide de pruebas	Pruebas funcionales, de caja negra, positivas y negativas	Sistema	Pruebas manuales Exploratorias	Validar el correcto funcionamiento de la app y sus funcionalidades, errores no detectados por otras técnicas
			Pruebas no funcionales, de caja negra positivas y negativas	Aceptación	GUI Ripper	Validar correspondencia de la GUI
			Pruebas funcionales, de caja negra, positivas	Aceptación	API's de automatización E2E	Validar la correcta integración entre las diferentes funcionalidades de la app móvil

2.5. Distribución de Esfuerzo



Como característica principal se propone que la metodología de pruebas siga los patrones relacionados con pirámide de pruebas. Sin embargo, de acuerdo con la información que poseemos, no se harán pruebas unitarias, pero se harán pruebas automatizadas con API Espresso y/o similares tipo rippers que aportan una manera ordenada de explorar el GUI de la aplicación.

En proporción se destinará un 70% de las pruebas a Espresso y el otro 30% restante a Pruebas exploratorias manuales para la primera iteración.