

# Proyecto integrado

**Daniel Jiménez Pacheco**

# Idea del proyecto

— — —

- Reproductor de música para dispositivos Android
- Hace uso de su propia base de datos interna
- Interfaz gráfica simple

# Antecedentes

- Existen muchos reproductores multimedia con características únicas:

- mpv
- NewPipe
- mixxx
- audacious

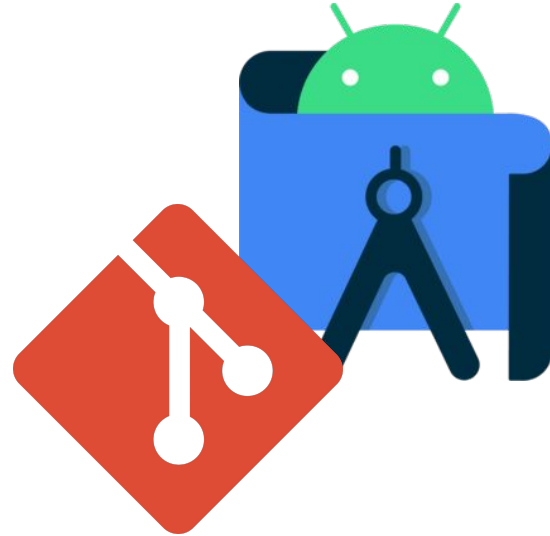


# Tecnologías usadas

— — —

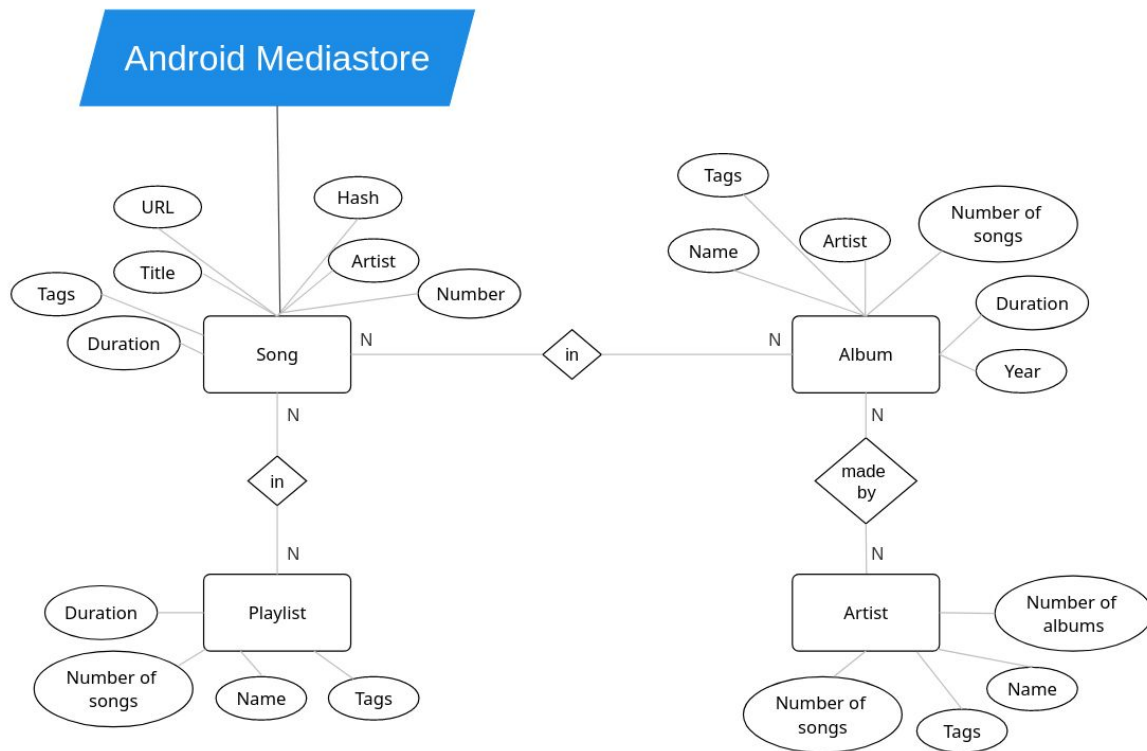
- IDE: Android Studio
- Lenguaje: Java
- Base de datos: sqlite3
- Control de versiones: git

[Repositorio GitHub](#)

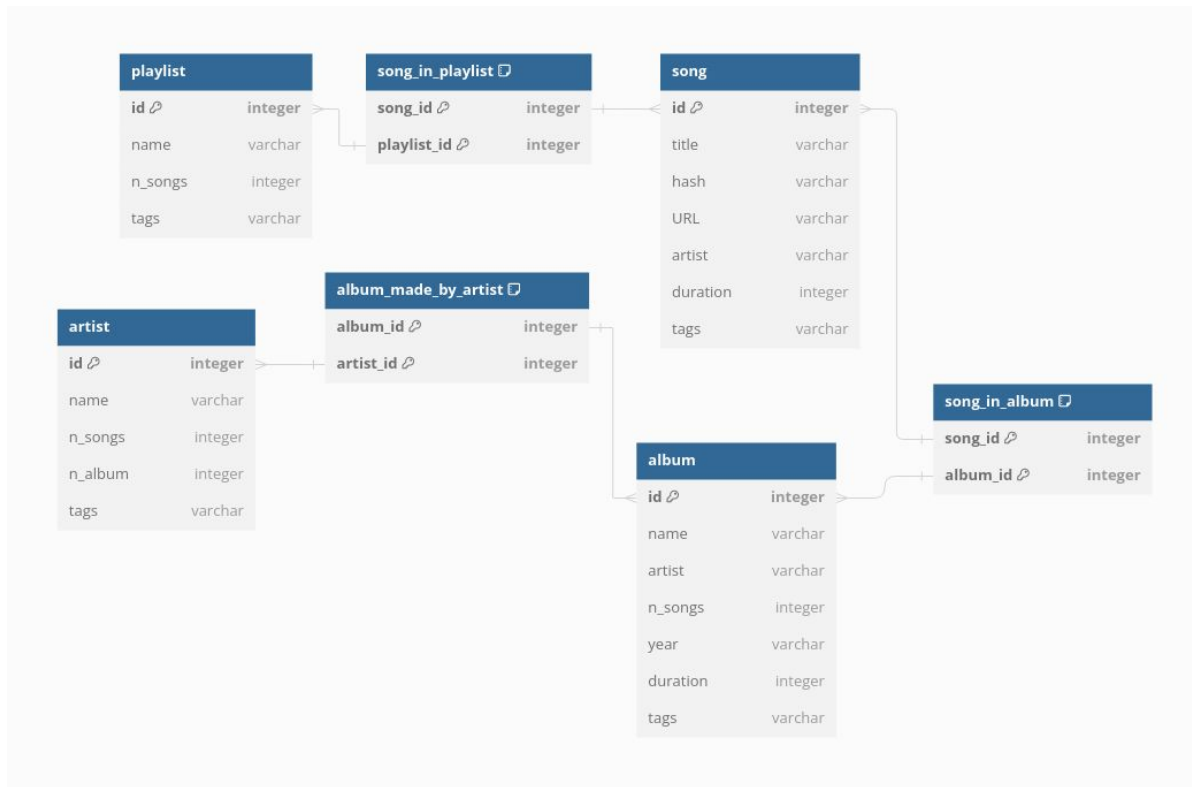


# Base de datos

# Modelo conceptual



# Modelo lógico



# Modelos: Canción

— — —

- ID: INTEGER (rowid)
- ID: INTEGER (MediaStore)
- Título: TEXT
- Artista: TEXT
- ID Artista: INTEGER (MediaStore)
- Álbum: TEXT
- ID Álbum: INTEGER (MediaStore)
- Número: INTEGER
- URL: TEXT
- MIME: TEXT
- Carátula: TEXT



# Modelos: Otros

— — —

- También están los modelos correspondientes a Álbum, Artista, Playlist
- Además de la tabla de relación entre Playlist y Canción
- Dado que se usa sqlite, los tipos de datos son limitados y pueden dar errores

**Diseño**

# Wireframe

Search

Artists

Artist 1

Artist 2

Artist 3

Albums

Album 1  
Artist 1

Album 2  
Artist 2

Songs

Title  
Artist

Title  
Artist

Playlist

Current song

Settings

Artist

2 albums  
10 songs

Albums

Album 1  
Artist

Album 2  
Artist

Songs

Title  
Artist

Title  
Artist

Title  
Artist

Title  
Artist

Title  
Artist

Album name  
Artist  
2024

20:41

1 Title  
Artist2:19

2 Title  
Artist2:19

3 Title  
Artist2:19

4 Title  
Artist2:19

5 Title  
Artist2:19

6 Title  
Artist2:19

7 Title  
Artist2:19

8 Title  
Artist2:19

9 Title  
Artist2:19

Title  
Artist

2:315:23

Playlist


Current song

Settings


# Mockup

ya


Artists



yandere




Dina Ayada




Yameii

Albums




Voyage  
I like to tak...




Bon voyage  
The KOXX

Songs




Isobeyan Gadget #435 Self...  
**Tenchio**




Clairvoyant  
**Toronto is broken**

Playlist Current song Settings




**Marietta**  
2 albums  
10 songs

Albums




As it were  
**Marietta**




Couples th...  
**Marietta / ...**


Songs




Yeah yeah, Utah!  
**Marietta**




Green call her sims  
**Marietta**




Hawaii 5-0 noseblunt  
**Marietta**



Pony up!!  
**Marietta**



United away  
**Marietta**



As it were  
**Marietta**  
2015  
34:42

11 Pony up!!  
**Marietta** 2:47

2 Beater  
**Marietta** 3:42

3 Horse tattoo  
**Marietta** 2:33

4 Are you afraid of god?...  
**Marietta** 4:50


5 5 minute raw vegan (a...  
**Marietta** 3:33

6 Ilai, Eli, a lie  
**Marietta** 2:06

7 No god in New Jersey  
**Marietta** 3:01

8 United away  
**Marietta** 2:41

9 Destination unknown (...  
**Marietta** 2:29



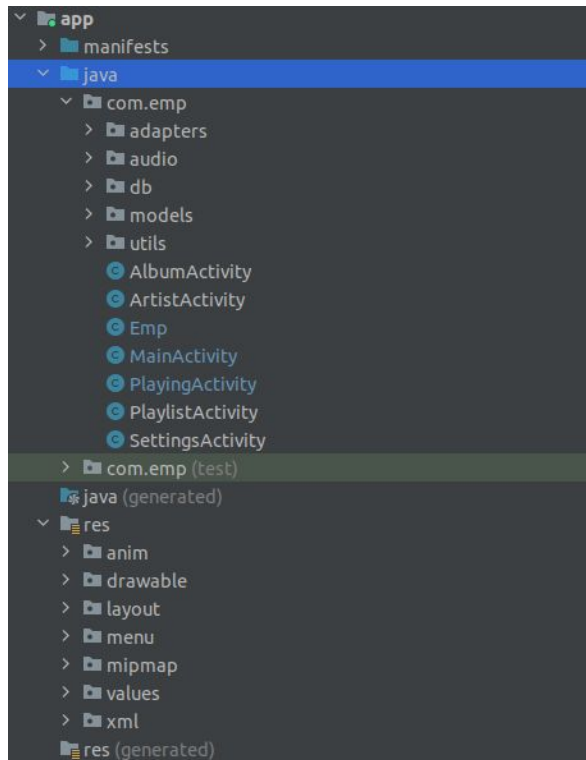
Pony up!!  
**Marietta**

0:45 2:47

Playlist Current song Settings

# Implementación

# Organización del proyecto



- El proyecto se organiza como cualquier otro proyecto de Android Studio
- Backend (paquetes):
  - Adapters: para las RecyclerViews
  - Audio: reproductor y servicio
  - DB: acceso a base de datos interna y handler
  - Models: modelos de la base de datos
  - Utils: formateo de timestamp
  - [Actividades]: actividades asociadas a cada vista
  - [Emp]: estado global
- Frontend:
  - Drawable: iconos por defecto
  - Layout: vistas
  - Menu: menus secundarios (popups, navbar)
  - Values: strings

# Backend

— — —

- **Emp:**
  - Hereda de Application
  - Guarda el estado global de la aplicación
  - Scope global
  - Se encarga de inicializar el reproductor y la base de datos
- **Connector:**
  - Sincroniza las bases de datos internas y MediaStore
  - Contiene métodos para extraer los datos de la base de datos interna
- **Player:**
  - Reproductor de música
  - Soporta los comandos básicos: pausa, siguiente, anterior
  - Contiene una queue y una referencia a la playlist

# Backend

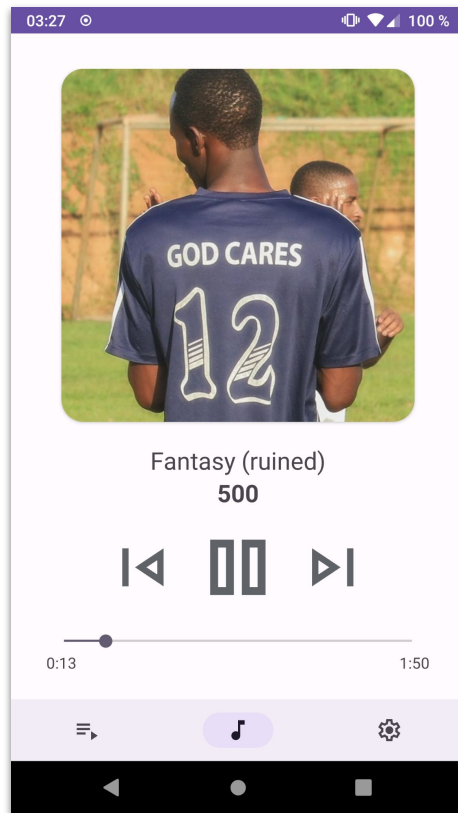
---

- Actividades:
  - Clases asociadas a las vistas del frontend
  - Contienen referencias a los distintos elementos visuales
  - Interactúan con el reproductor de música mediante PlayerCallback
- PlayerCallback:
  - Interfaz de la que heredan algunas actividades
  - Las actividades se “registran” al reproductor
  - Cuando hay cambios en el estado del reproductor, éste notifica a las actividades para modificar los elementos en pantalla



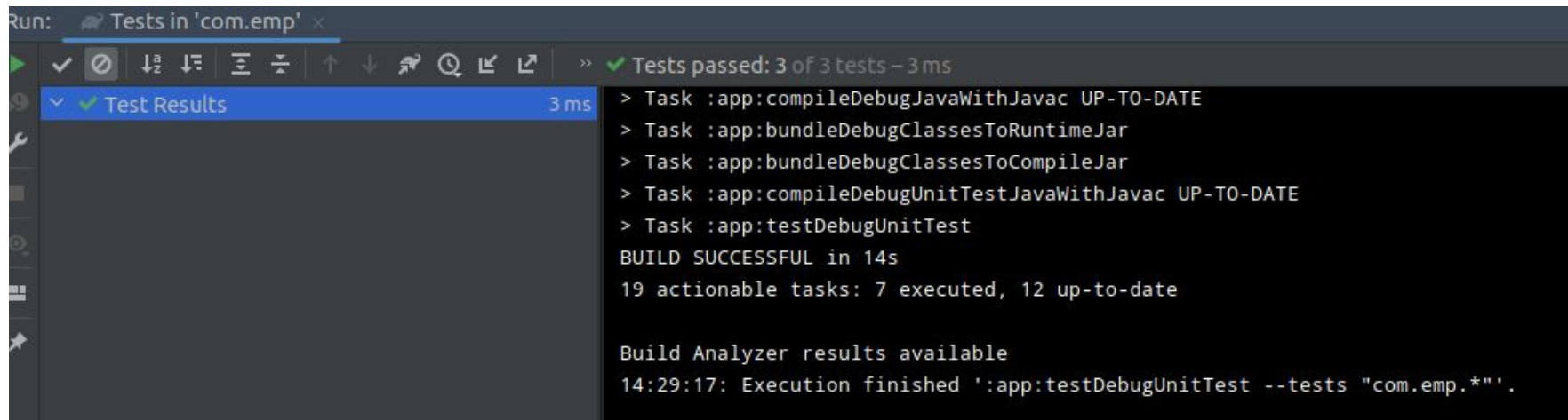
# Frontend

- Esquema típico de aplicación de Android
- Uso de RecyclerViews para representar los datos
- Interfaz del reproductor simple
- Barra de navegación es vistas principales



# Pruebas

- Las pruebas se realizan con JUnit4
- Se realizan sobre la clase Time, que formatea los timestamps de las canciones



The screenshot shows the 'Run' console of an IDE. The top bar indicates 'Tests in 'com.emp' x'. Below it, a toolbar contains icons for running, debugging, and other actions. The main area displays the following output:

```
Run: Tests in 'com.emp' x
> Task :app:compileDebugJavaWithJavac UP-TO-DATE
> Task :app:bundleDebugClassesToRuntimeJar
> Task :app:bundleDebugClassesToCompileJar
> Task :app:compileDebugUnitTestJavaWithJavac UP-TO-DATE
> Task :app:testDebugUnitTest
BUILD SUCCESSFUL in 14s
19 actionable tasks: 7 executed, 12 up-to-date

Build Analyzer results available
14:29:17: Execution finished ':app:testDebugUnitTest --tests "com.emp.*"'.
3 ms
```

Demo

# Conclusiones

# Complicaciones

— — —

- Problemas con el reproductor MediaPlayer
- Problemas con la versión de SDK de mi móvil
  - No compatible con NewPipe Extractor
  - Campo ID de artista en álbum no disponible en MediaStore (???)
- Problemas con Android Studio
  - DEMASIADO consumo de memoria RAM
- MediaPlayer es muy “básico”
  - No permite acceder al stream de datos
  - Otras clases alternativas son confusas
- La documentación de Android y NewPipe Extractor
- Dificultad para compilar ffmpeg con Android Studio
- . . .

# Lecciones aprendidas

— — —

- Aunque no sea “digno de subir” hacer siempre push al repositorio
- No hacer estimaciones sobre el tiempo que ocuparán determinadas partes del proyecto
- Ser más realista a la hora de elaborar los requisitos, o elegir las tecnologías y lenguajes a usar
- Verificar de antemano que las librerías sean compatibles
- Usar un ordenador mejor...

Fin