



MediaVault

Trabalho Prático CM

Parte 2 Entrega

Grupo F Tema 2

Licenciatura de Engenharia da Computação Gráfica e
Multimédia

Unidade Curricular: Computação Móvel
IPVC-ESTG

2024/2025

Realizado por:

Marco Fernandes, N° 30566

Francisco Pimenta, N° 25418

Duarte Pires, N° 29999

Índice

Introdução.....	4
Requisitos Funcionais	5
Requisitos Não Funcionais	7
Plataforma de Gestão de Projeto	8
Plataforma de Protótipo e Desenho da Navegação.....	9
GitHub	9
Diagramas de Arquitetura	10
Conclusão	13
Referências	14

Índice de Figuras

Figura 1 – Plano de Trabalho no Asana.....	8
Figura 2 - Gestão de Dependências	10

Introdução

O presente relatório corresponde à Entrega 2 do projeto da unidade curricular de Computação Móvel, no âmbito da Licenciatura em Engenharia da Computação Gráfica e Multimédia.

A aplicação MediaVault é uma ferramenta móvel desenvolvida para a plataforma Android, com o objetivo de permitir aos utilizadores a organização, edição e gestão eficiente dos seus conteúdos multimédia pessoais, incluindo fotografias, vídeos e ficheiros de áudio. Através de um sistema de autenticação seguro, a aplicação oferece perfis personalizados, permitindo que os utilizadores criem álbuns ou playlists, apliquem tags e categorias, realizem pesquisas avançadas e partilhem os seus conteúdos com outros utilizadores.

A sincronização com a cloud é efetuada através do Firebase, assegurando que os dados sejam armazenados de forma segura e acessível. A aplicação foi também concebida para funcionar offline, com a possibilidade de sincronização automática dos dados quando uma conexão à internet estiver disponível.

Requisitos Funcionais

Os requisitos funcionais da aplicação serão:

Sistema de contas de utilizadores:

- Utilizador tem de criar conta e iniciar sessão para utilizar a aplicação;
- O utilizador pode depois gerir o seu perfil e alterar dados como nome, username, email, fotografia e password.

Gestão de multimédia:

- Criar álbuns ou playlists de fotografias, vídeos e áudios que podem ser adicionados e substituídos escolhendo do dispositivo diretamente, ou gravando tirando uma foto/vídeo com a câmara do telemóvel;
- Adicionar, alterar ou remover conteúdos multimédia, ambos das playlists/álbuns e da base de dados do utilizador;
- Visualização de multimédia (através da biblioteca MediaPlayer);
- As multimédias e os álbuns/playlists podem ter tags ou categorias descritivas dos seus conteúdos;
- O utilizador pode procurar através de uma barra de pesquisa, podendo complementar a pesquisa filtrando por data de upload, tipo de conteúdo, tags e categorias;
- O utilizador poderá editar os conteúdos multimédia de uma forma rudimentar:
 - Fotografias:
 - Usar a biblioteca uCrop para rodar, cortar e ajustar posição das fotografias;
 - Vídeos e Áudios:
 - Usar a biblioteca Mobile-FFmpeg para recortar tempos e ajustar as posições de partes no vídeo.
 - Partilhar conteúdos ou álbuns/playlists com outros utilizadores.

Sistema de base de dados:

- Backup dos dados dos utilizadores serão postas em armazenamento Cloud (serviço escolhido foi Firebase), onde enquanto que o utilizador ficar off-line o backup fica somente localmente (através da biblioteca Room).

Requisitos Não Funcionais

A arquitetura que se vai usar é MVVM (Model-View-ViewModel), onde se vai ter vários Fragments a comunicarem com uma ViewModel que vai partilhar dados com todos eles, bem como cada Fragment vai comunicar com os dados que estarão guardados na Room (assim a partilha de informação entre ecrãs da aplicação fica de forma modular e facilmente repetível).

A Room vai atualizar com os dados do utilizador guardados na base de dados cloud Firebase cada vez que houver um início de sessão, onde a partir desse ponto todas as ações que o utilizador fizer na aplicação serão feitas na base de dados Room e quando a Activity é terminada, caso o user esteja on-line, vai-se sincronizar os dados da cloud de Firebase com os dados da Room (desta forma poupa-se em recursos do dispositivo pois não vai ser preciso esperar pelas comunicações com a cloud Firebase em todos os processos).

Plataforma de Gestão de Projeto

A plataforma de gestão de progresso do projeto desenvolvido será a Asana, pois já existe familiaridade com os elementos do grupo na mesma.

▼ Primeira Entrega	
✓ Asana	MF Marco Ferna...
▼ ✓ Figma 4	
✓ Landscape	MF Marco Ferna...
✓ Portrait	Francisco Pi...
✓ Tradução - Landscape	MF Marco Ferna...
✓ Tradução - Portrait	José, Engravi...
✓ Relatório	José, Engravi...
Adicionar tarefa...	
▼ Segunda Entrega	
✓ App	
✓ Relatório	
Adicionar tarefa...	
▼ Frontend	
✓ Intro Sliders	
▶ ✓ Gestão User 2	
▶ ✓ Gestão Multimédia 3	
Adicionar tarefa...	
▼ Backend	
▶ ✓ Base de Dados (Firebase) 3	Francisco Pi...
▶ ✓ User 3	
▶ ✓ Media 10	
✓ Update de Offline para Online Sync	
✓ Backup Automatico em armazenamento Cloud (Firebase)	Francisco Pi...

Figura 1 – Plano de Trabalho no Asana

[Link para a página de Asana deste projeto](#)

Plataforma de Protótipo e Desenho da Navegação

A navegação para a nossa aplicação foi desenhada em conjunto com o nosso protótipo do *layout*, que inclui ambos formatos *Portrait* e *Landscape*, bem como a implementação para as línguas portuguesa e inglesa.

[Link para a página com os vários layouts e línguas](#)

[Link para o teste do protótipo e sua navegação](#)

GitHub

Para facilitar o acesso e acompanhamento do desenvolvimento deste projeto, disponibilizamos o código fonte e todos os recursos relevantes no nosso repositório GitHub. No final do projeto, o repositório terá todo o esforço e dedicação de cada elemento do grupo.

O repositório será atualizado de forma contínua com novas funcionalidades e correções, garantindo que todos tenham acesso à versão mais recente do projeto.

[Link para o repositório do github](#)

Diagramas de Arquitetura

1. Camadas Principais

- **UI Layer:** Activities, Fragments, ViewModels
- **Data Layer:** Repositórios, Firebase e SQLite/Room
- **Domain Layer:** Lógica de negócio

2. Fluxo Geral

diagrama visual

3. Gestão de Dependências

```
dependencies {  
  
    implementation(libs.androidx.core.ktx)  
    implementation(libs.androidx.appcompat)  
    implementation(libs.material)  
    implementation(libs.androidx.activity)  
    implementation(libs.androidx.constraintlayout)  
    implementation(libs.androidx.navigation.fragment.ktx)  
    implementation(libs.androidx.navigation.ui.ktx)  
    testImplementation(libs.junit)  
    androidTestImplementation(libs.androidx.junit)  
    androidTestImplementation(libs.androidx.espresso.core)  
  
    //Firebase dependencies  
    implementation(platform("com.google.firebase:firebase-bom:33.13.0"))  
    implementation("com.google.firebase:firebase-analytics")  
    implementation("androidx.credentials:credentials:1.3.0")  
    implementation("com.google.firebase:firebase-auth")  
    implementation("androidx.credentials:credentials-play-services-auth:1.3.0")  
    implementation("com.google.android.libraries.identity.googleid:googleid:1.1.1")  
    implementation("com.google.firebase:firebase-firestore-ktx")  
  
    //Exoplayer dependencies  
    implementation("androidx.media3:media3-exoplayer:1.6.1")  
    implementation("androidx.media3:media3-ui:1.6.1")  
    implementation("androidx.media3:media3-exoplayer-dash:1.6.1")  
}
```

Figura 2 - Gestão de Dependências

4. Instruções de Instalação

Pré-requisitos

- Android Studio
- SDK 33 ou superior
- Conta Google configurada (para Firebase)

Clonar o repositório

- “git clone <https://github.com/FranciscoPPimenta/MediaVault.git>”

Abrir o projeto no Android Studio

Configurar o Firebase

-

Executar o projeto

- Selecionar dispositivo virtual ou físico
- Clicar em “Run”

5. Instruções de Uso

- **Intro Slides:** Ao iniciar, são mostrados tutoriais com descrição da app.
- **Registo/Login:** O utilizador pode criar conta ou iniciar sessão.
- **Perfil:** Pode gerir os seus dados pessoais e foto de perfil.
- **Multimédia:**
 - Criar álbuns ou playlists
 - Adicionar conteúdos com tags/categorias
 - Visualizar conteúdos em galeria/lista
 - Usar ferramentas básicas de edição
- **Pesquisa e Partilha:** Utilizar filtros e partilhar conteúdos com outros utilizadores.

- **Backup & Offline:** Todos os dados são guardados no Firebase e sincronizados automaticamente quando a ligação é restaurada.

Adicionar fotos em cada ponto...

Conclusão

A aplicação MediaVault demonstra a capacidade de integrar diversas funcionalidades essenciais para a gestão e organização de conteúdos multimédia pessoais, proporcionando aos utilizadores uma experiência intuitiva e eficiente. Através da sua interface amigável e recursos como a criação de álbuns e playlists, edição básica de conteúdos e pesquisa avançada, a aplicação facilita a organização e personalização dos ficheiros multimédia, atendendo às necessidades dos utilizadores.

A implementação de autenticação segura e perfis personalizados garante uma experiência única para cada utilizador, enquanto a sincronização com a cloud, juntamente com a funcionalidade offline, assegura que os dados estão sempre acessíveis e protegidos. A utilização do Firebase como plataforma para a gestão e armazenamento de dados oferece segurança e escalabilidade.

Em suma, a MediaVault não só cumpre os requisitos propostos, mas também proporciona uma solução prática e robusta para a gestão de conteúdos multimédia pessoais, com potencial para evoluir e incorporar novas funcionalidades que atendam ainda melhor às necessidades dos utilizadores no futuro.

Referências

1. Figma. (2024). *Figma: the Collaborative Interface Design tool*. Figma. <https://www.figma.com/>
2. Asana. (n.d.). *Gestão on-line dos trabalhos, projetos e tarefas da sua equipe* • Asana. Asana. <https://asana.com/pt>
3. GitHub. (2025). *GitHub*. GitHub. <https://github.com/>