

# Relatório Projeto Final

ATEC HEADQUARTER MOBILE

UCoo609 | 16-02-2026

Hugo Bacalhau To121748 e Tiago Rosa To130956

# Índice

Informações Gerais: .....	2
Decisões Técnicas .....	2
Stacks .....	2
Kotlin: .....	2
Jetpack Compose:.....	2
Retrofit 2 + OkHttp: .....	2
Navigation Compose: .....	3
Arquitetura .....	3
Design .....	3
Material 3:.....	3
Desafios e Soluções .....	4
Conclusao .....	4

## Informações Gerais:

**Aplicação:** ATEC HeadQuarter Mobile

**Objetivo:** Aplicação Android para consulta de dados como cursos ativos, turmas ativas, formandos, formadores e salas disponíveis. A aplicação visa ser usada por formadores da instituição servindo de complemento móvel da plataforma web ATEC HeadQuarter.

**Stacks:** Kotlin, Jetpack Compose, Material 3, Retrofit 2, OkHttp, Gson, Navigation Compose.

**Arquitetura:** MVVM (*Model-View-ViewModel*)

**Funcionalidades complexa:** Fluxo de autenticação JWT, injeção automática de *token* e controlo de acesso por *role*.

**Decisão difícil:** Optar por OkHttp Interceptor + SharedPreferences para gestão de *tokens* devido à dificuldade de integração.

## Decisões Técnicas

### STACKS

#### Kotlin:

A escolha da linguagem Kotlin prendeu-se ao facto de que foi a que abordámos em aula e também a pedida nos critérios de avaliação da UCo609. Caso contrário o grupo teria optado por React Native devido à linguagem escolhida para o projeto Web (React e Node.js) por razões de praticidade e performance.

#### Jetpack Compose:

A decisão de usar Compose em vez do sistema tradicional de Views + XML prendeu-se não só ao facto de o termos abordado em aula, como também a sua longevidade no futuro de Android UI.

#### Retrofit 2 + OkHttp:

A escolha de Retrofit deveu-se ao facto de a ferramenta facilitar integração com o *backend* em Node.js. Além disso, com os Interceptors do OkHttp, conseguimos injetar o *token* JWT automaticamente em todas as chamadas, o que poupa o trabalho manual de adicionar o *header* em cada *request*. Outro ponto forte é o Gson Converter, que faz a conversão de JSON para Kotlin de forma automática, sem que precisássemos de fazer a conversão à mão.

## Navigation Compose:

O uso de Navigation Compose prendeu-se ao simples facto de que é uma biblioteca da Google usada para gerir a navegação entre ecrãs em aplicações Jetpack Compose. Na ATEC HeadQuarter Mobile, toda a navegação está centrada num *NavHost* que define as rotas, login, *dashboard*, cursos, turmas, formadores e salas. A centralização facilita a manutenção, uma vez que a alteração de um ecrã é feita num único ponto do código. A navegação suporta ainda passagem de parâmetros entre ecrãs, como por exemplo ao clicar numa turma para ver os seus alunos, onde o nome da turma é enviado para o ecrã de detalhes. Após o login, o ecrã de autenticação é removido da navegação, garantindo que o botão "Voltar" do dispositivo fecha a aplicação em vez de regressar ao formulário de login.

## ARQUITETURA

A escolha da arquitetura MVVM (*Model-View-ViewModel*) fundamenta-se com um dos critérios de avaliação, contudo o grupo viu também justificações práticas aquando do desenvolvimento do projeto:

1. **Escalabilidade** – Devido à organização que a arquitetura possui, criamos um código limpo e organizado, conseguimos que no futuro seja mais fácil adicionar funcionalidades.
2. **Manutenção** - Caso a API mude o endpoint limitamo-nos a alterar o ficheiro “*ApiService*” para o endpoint atualizado sem causar impacto nos restantes.

## DESIGN

### Material 3:

O Material Design 3 é um sistema de design da Google para aplicações Android. Na ATEC HeadQuarter Mobile, foi usado para garantir que a interface segue as convenções visuais que o sistema Android possui (botões, cartões, campos de texto e ícones), estes comportam-se como o utilizador espera, sem necessidade de explicação.

O tema foi personalizado com as cores oficiais da ATEC HeadQuarter, o que garante coerência visual entre a versão web e a versão móvel da plataforma. A escolha do Material 3 em vez de uma UI totalmente personalizada reduz o tempo de desenvolvimento e assegura acessibilidade.

## DESAFIOS E SOLUÇÕES

Durante o desenvolvimento do projeto *mobile* enfrentamos desafios como o problema o *localhost* que não existia e o da autenticação JWT entre cliente - servidor Node.js.

O primeiro problema veio no seguimento dos testes à aplicação, como a API funciona em *localhost* no computador o emulador por sua vez com este endereço procurava dentro do seu próprio sistema em vez de procurar na máquina hospedeira.

A solução passou por configurar a rede da aplicação para apontar para o IP que queríamos, isto permitiu que o AndroidStudio saísse da sua “caixa” e acedesse ao servidor Node.js do ncomputador.

O segundo problema passou por garantir que o *token* fosse injetado em todos os pedidos, para isto o grupo considerou o uso do *token* manual em cada *ViewModel*, seria simples de implementar, contudo haveria repetição de código. Com alguma pesquisa deparamo-nos com o OkHttpInterceptor + SharedPreferences, esta solução trazia-nos automatização e anulava a repetição de código, contudo seria mais complexa de implementar.

Todavia decidimos usá-la, criamos o ficheiro “TokenManager” que serve apenas para guardar e ler o *token* JWT. Criamos ainda o “AuthInterceptor” que nos serviu como intermediário, ou seja, sempre que há um pedido este ficheiro vai ao “TokenManager”, retira uma cópia do *token* e coloca-a no pedido efetuado e depois envia-o.

## CONCLUSÃO

A aplicação móvel ATEC HeadQuarter Mobile cumpre os requisitos funcionais, entregando uma solução *mobile* robusta para a gestão académica. A utilização de Kotlin e Jetpack Compose garante longevidade e facilidade de manutenção do código.

A arquitetura MVVM assegura uma separação necessária para que a aplicação seja escalável: a lógica de autenticação e comunicação com a API REST está isolada da interface do utilizador, permitindo futuras integrações (como notificações push) sem necessidade de refatorização profunda. Em suma, o produto final não é apenas um protótipo funcional, mas sim uma base sólida, segura e preparada para evoluir num ambiente de produção real.