

Programação Android com Kotlin

Professor: Celso Brunno Rocha Custódio de Campos

Objetivo Geral

Capacitar os alunos a desenvolver aplicativos Android funcionais utilizando Kotlin e Jetpack Compose, desde fundamentos e lógica de programação até a criação de um aplicativo completo.

Objetivos Específicos

- Dominar fundamentos de Kotlin e lógica de programação no IntelliJ IDEA.
- Configurar o ambiente de desenvolvimento Android com Android Studio.
- Criar interfaces modernas com Jetpack Compose, incluindo personalização e gerenciamento de estados.
- Implementar persistência de dados com Room e lógica de negócios com ViewModel/StateFlow.
- Aplicar navegação, testes unitários, e boas práticas (MVVM, separação de responsabilidades).
- Desenvolver um aplicativo funcional otimizado para performance e usabilidade.

Conteúdo Programático

1. Introdução ao Kotlin e Lógica de Programação

- Importância do Kotlin para Android.
- Configuração do IntelliJ IDEA.
- Variáveis, tipos de dados, funções, estruturas de controle (if, when).
- Lógica de programação: Algoritmos simples (cálculos, validações).
- Prática: Programa para calcular média de notas e verificar aprovação.

2. Kotlin Avançado e Lógica de Programação

- Coleções (List, Set, Map), loops (for, while).
- Programação orientada a objetos: Classes, objetos, herança.
- Funções de ordem superior, lambdas, padrão Adapter.
- Lógica de programação: Problemas complexos (ordenação, validação).
- Prática: Programa para gerenciar lista de tarefas em memória.

3. Introdução ao Android e Jetpack Compose

- Configuração do Android Studio.
- Estrutura de um projeto Android.
- Primeira composição, componentização, importação de ícones/imagens.
- Prática: Projeto "Olá, Mundo" com Jetpack Compose.

4. **Personalização de Componentes**

- Configuração de temas, fontes, e cores.
- Uso de modificadores para espaçamentos e layouts.
- Acessibilidade em componentes.
- Prática: Tela personalizada com tema e componentes customizados.

5. **Fundamentos de Jetpack Compose (2h30min)**

- Estados e recomposições.
- Gerenciamento de estados em formulários.
- Botões, callbacks, e Layout Inspector.
- Prática: Formulário interativo com estado.

6. **Listas e Ações (2h30min)**

Criação de listas com LazyColumn.

- Ações em itens (adicionar, editar, remover).
- Formatação de datas e horas.
- Prática: Lista de tarefas com ações.

7. **Persistência de Dados com Room (2h30min)**

- Configuração do Room: Entidades, DAOs, Database.
- Operações CRUD.
- Integração com ViewModel e StateFlow.
- Prática: Banco de dados para tarefas.

8. **Depuração, Navegação, Testes e Publicação (2h30min)**

- Otimização de listas e layouts.
- Navegação com Navigation Compose.
- Testes unitários com JUnit.
- Assinatura do APK e preparação para publicação.
- Prática: Finalizar e apresentar o aplicativo.

Metodologia

- Aulas expositivas com exemplos práticos.

- Exercícios práticos por aula.
- Acompanhamento individual.

Pré-requisitos

- Lógica de programação (recomendado).
- Computador com IntelliJ IDEA e Android Studio.

Pré-requisitos Técnicos

- Android Studio exige computadores com pelo menos 8 GB de RAM e boa conexão para baixar dependências.

Projeto Final

Desenvolvimento de um aplicativo de Lista de Tarefas(to-do-list) com:

- Interface em Jetpack Compose.
- Persistência com Room.
- Navegação entre telas.
- Operações CRUD.
- Boas práticas (MVVM, StateFlow).

Disponibilização dos Materiais

- https://github.com/CelsoBrunno/kotlin_compose_unintesi