



Estácio

Graduação e Pós-graduação

Trabalho Prático | DGT2812 Desenvolvimento de Aplicativos Móveis com Flutter

Nome: Maurício Pereira Campos

Matricula: 2024 0384 3447

Turma: 2025.3

Período: 4º Período

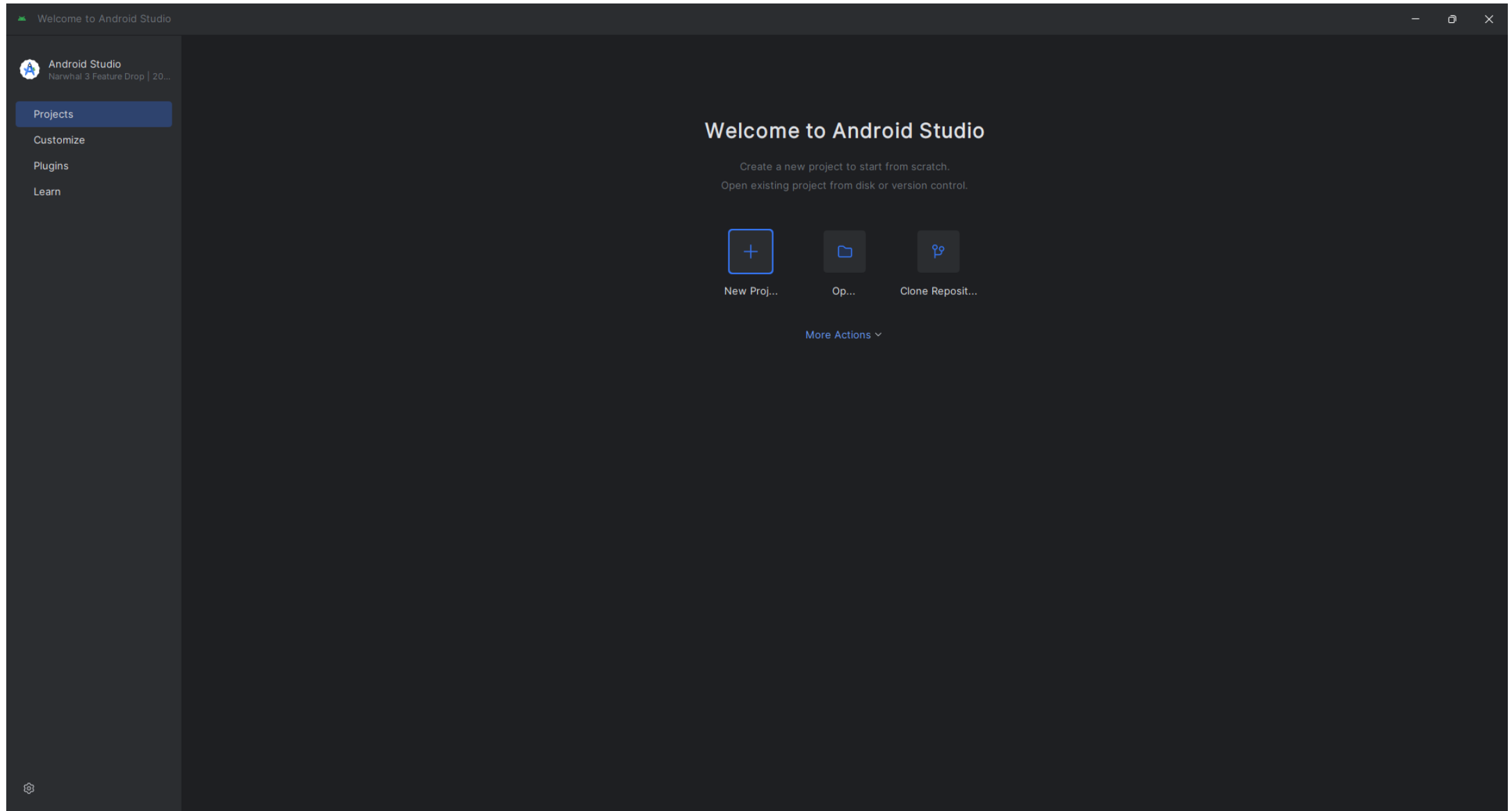
Disciplina: DGT2816 – Interação com sensores de Smartphones e Wearebles

Objetivo do Trabalho:

- Instalação do Android Studio e do emulador (Feito);
- Criar um app para Wear Os;
- Executar um app no emulador;
- Fazer capturas de tela no Android Studio;
- Fazer capturas de tela com app complementar.

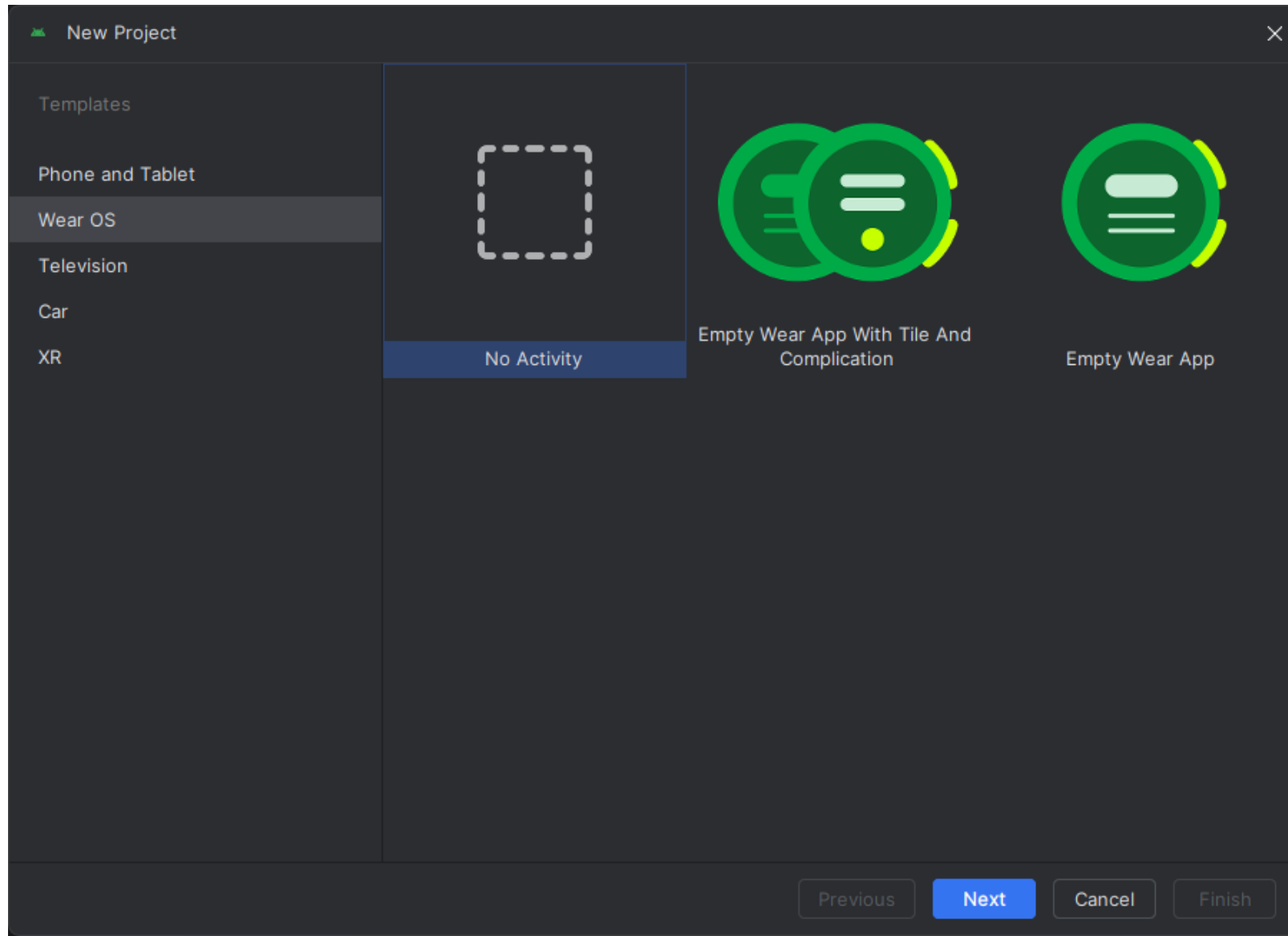
Microatividade 2: Criando um novo Projeto no Android Studio (Wear OS)

Resultado da tela:



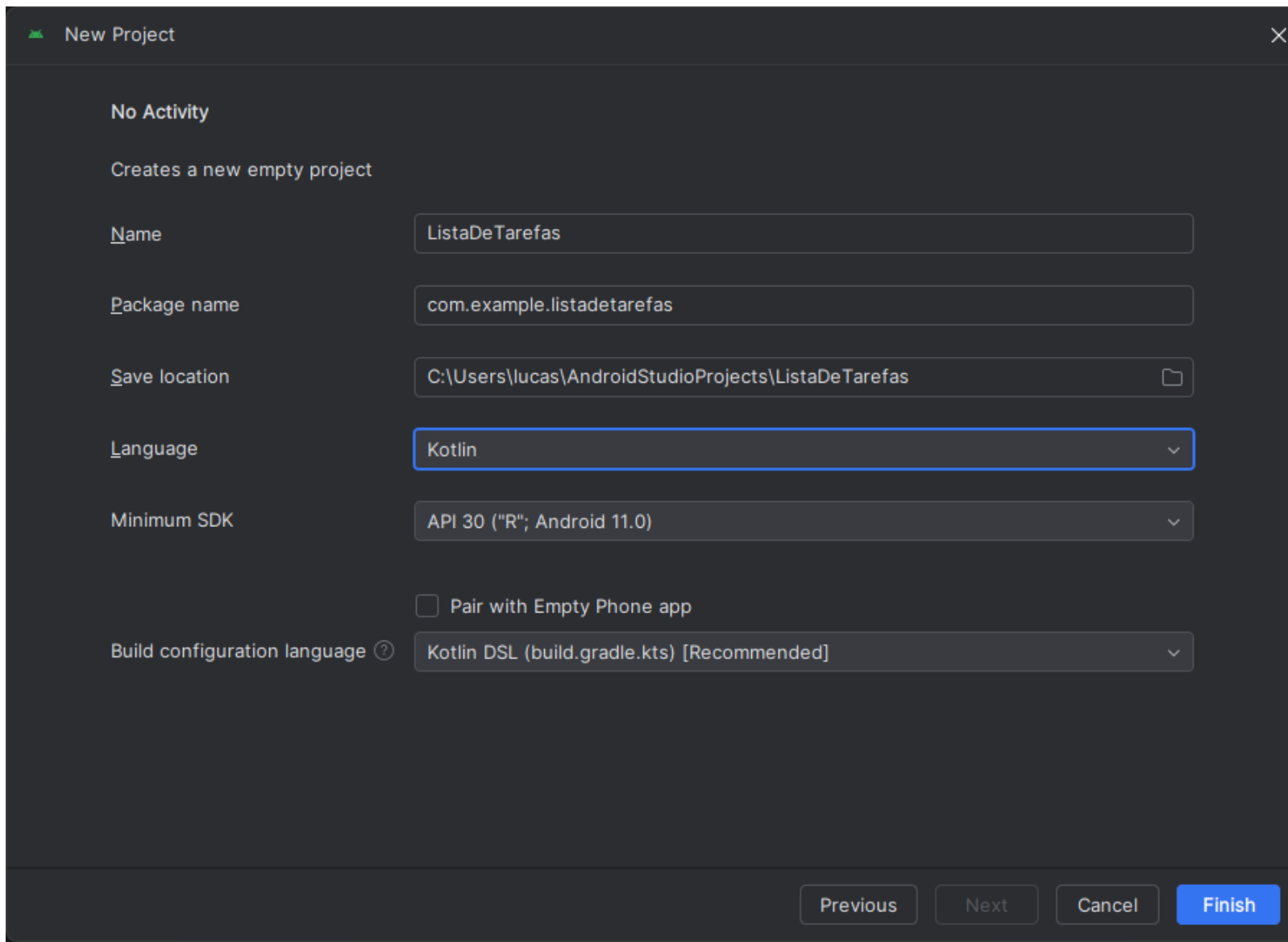
Microatividade 2: Criando um novo Projeto no Android Studio (Wear OS)

Resultado da tela:



Microatividade 2: Criando um novo Projeto no Android Studio (Wear OS)

Resultado da tela:



New Project

No Activity

Creates a new empty project

Name: ListaDeTarefas

Package name: com.example.listadetarefas

Save location: C:\Users\lucas\AndroidStudioProjects\ListaDeTarefas

Language: Kotlin

Minimum SDK: API 30 ("R"; Android 11.0)

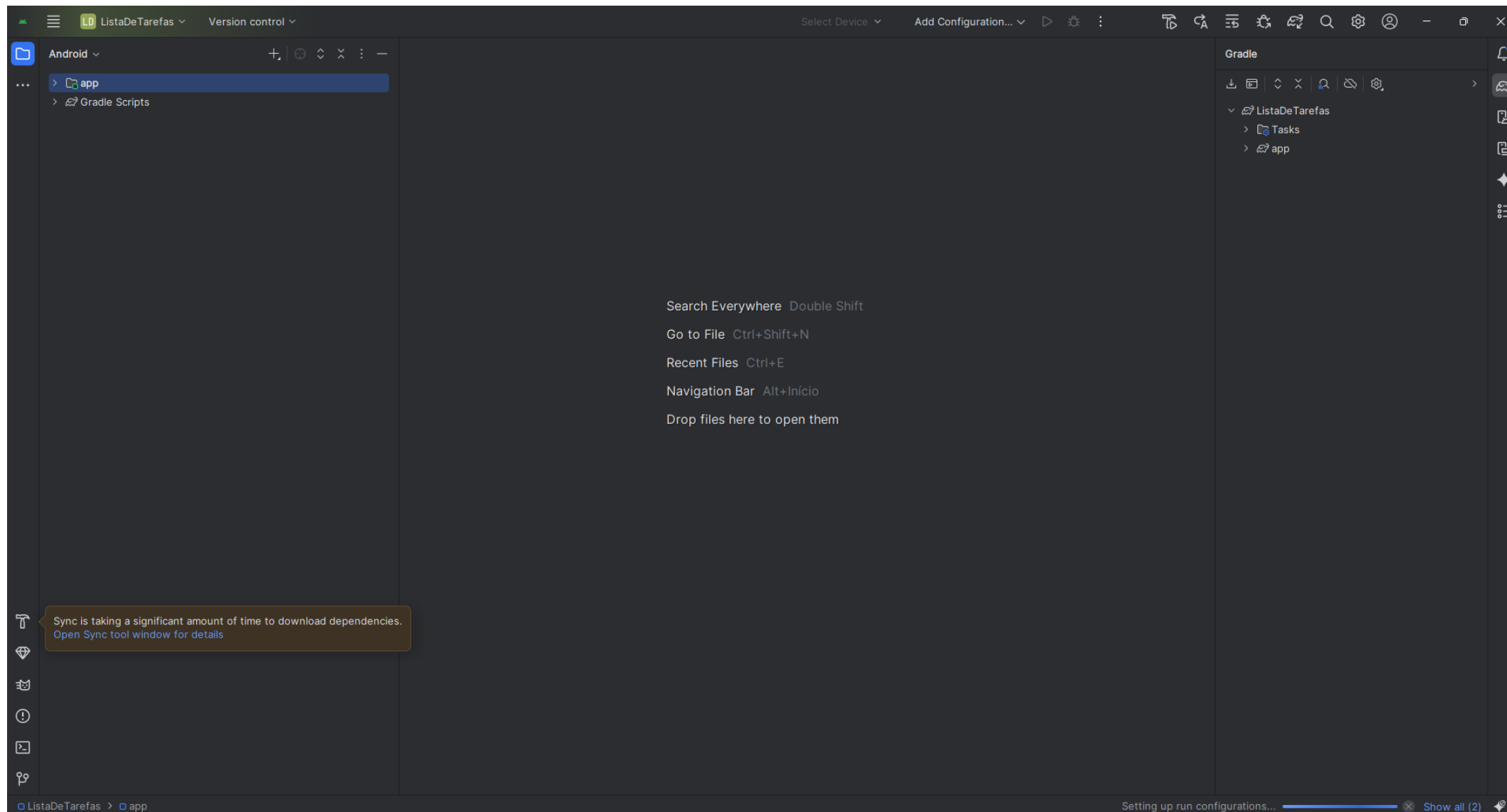
☐ Pair with Empty Phone app

Build configuration language ⓘ: Kotlin DSL (build.gradle.kts) [Recommended]

Previous Next Cancel Finish

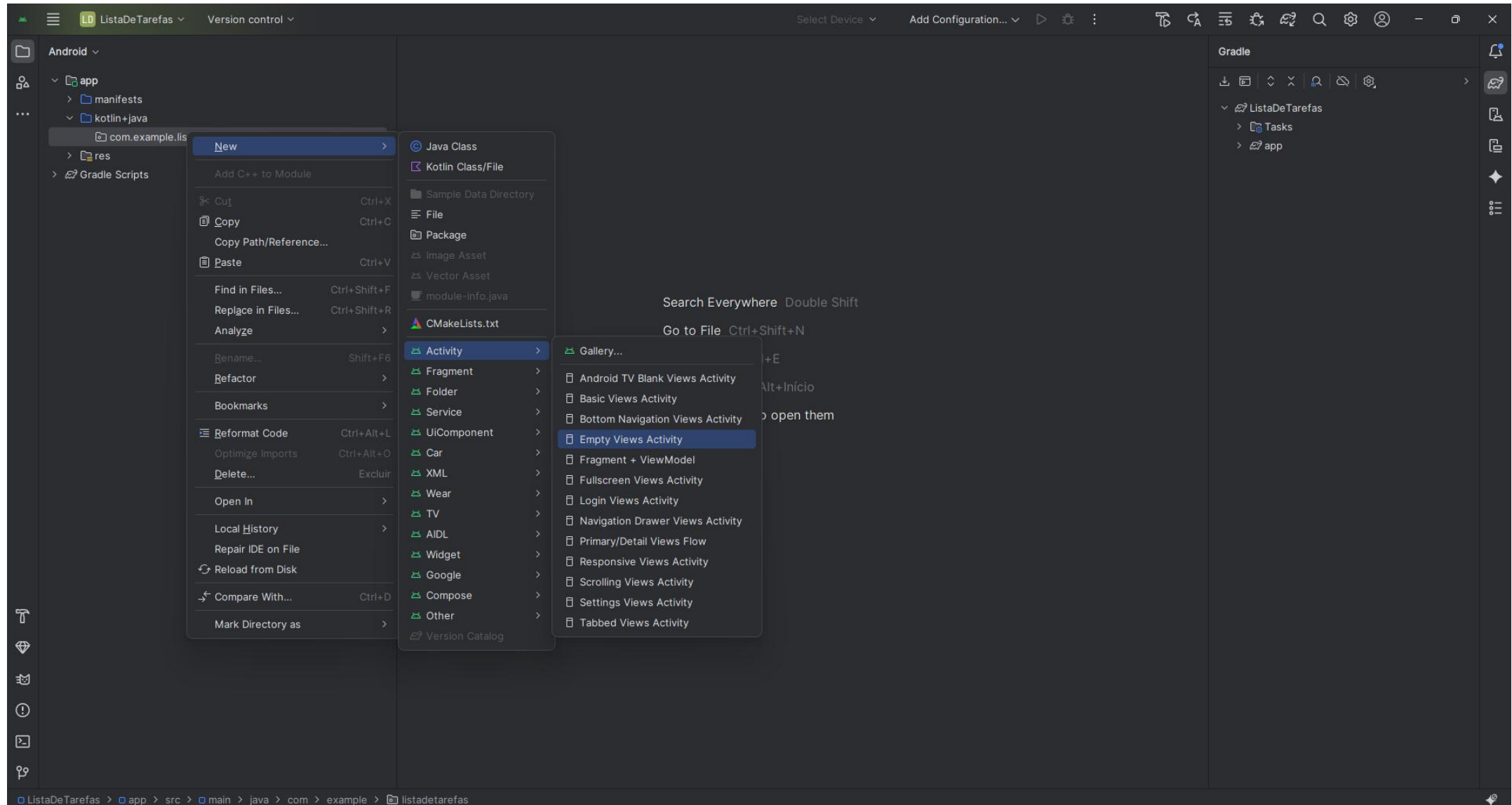
Microatividade 2: Criando um novo Projeto no Android Studio (Wear OS)

Resultado da tela:



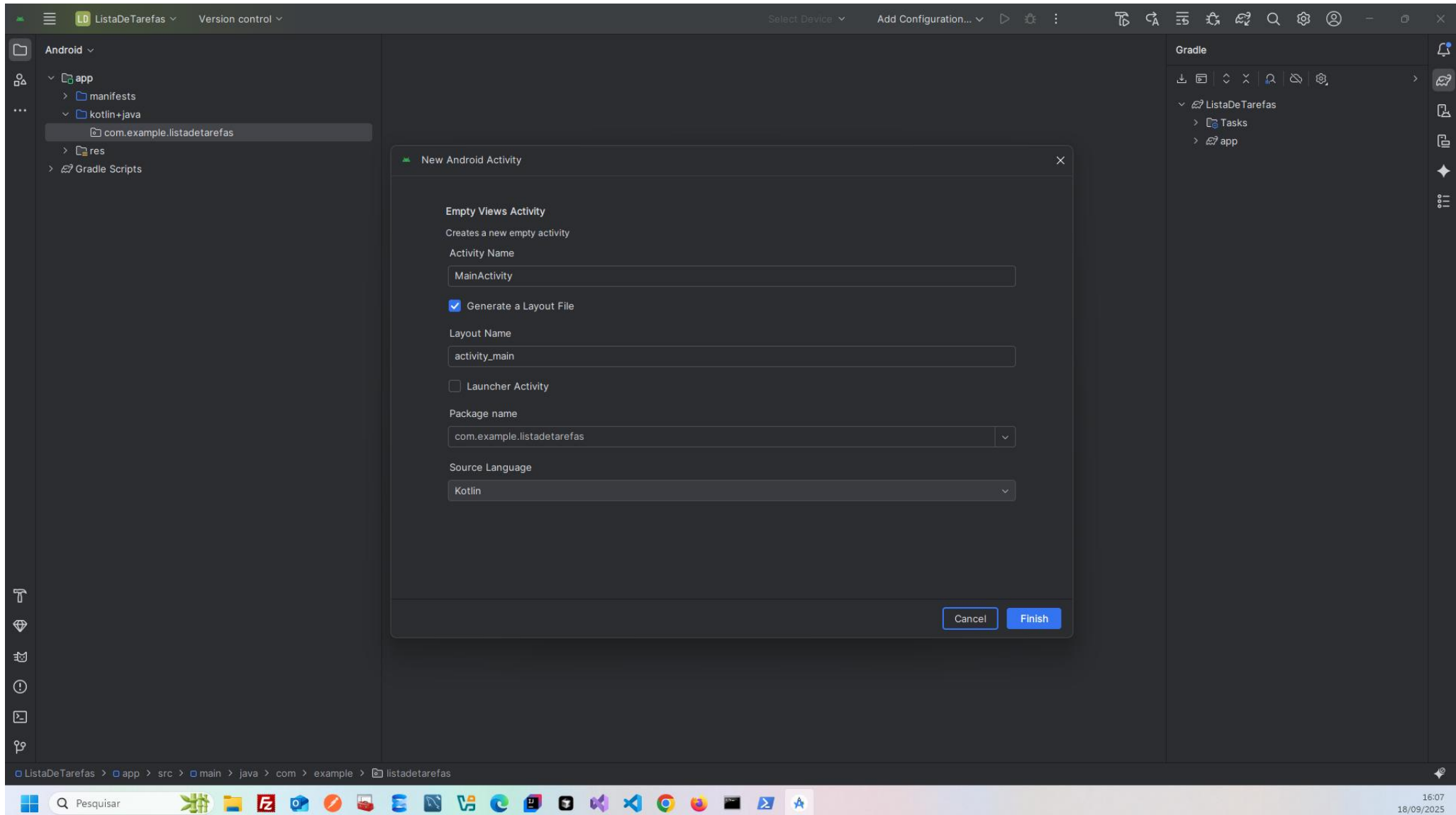
Microatividade 3: Arquivos de lógica e configurações

Resultado da tela:



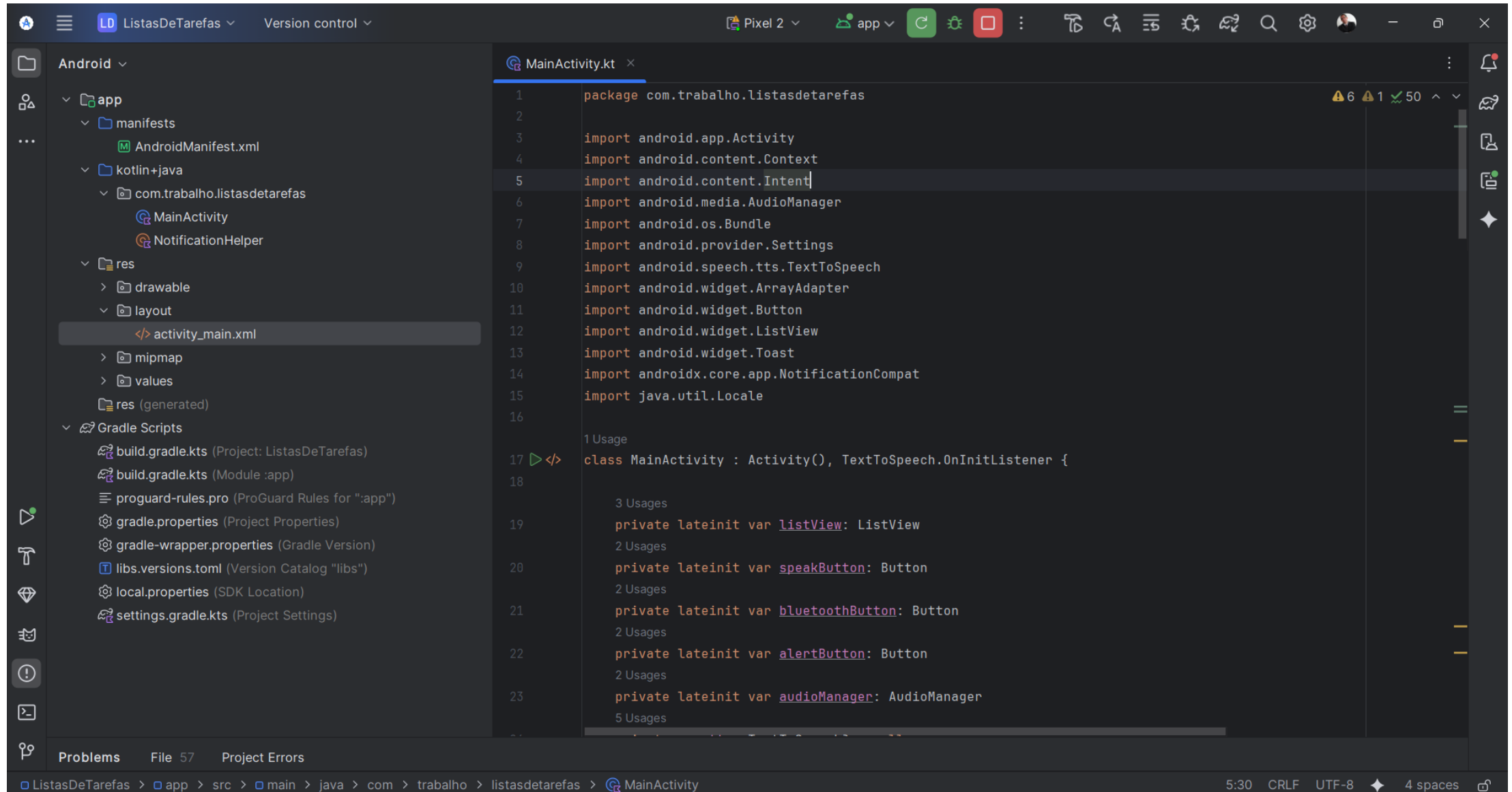
Microatividade 3: Arquivos de lógica e configurações

Resultado da tela:



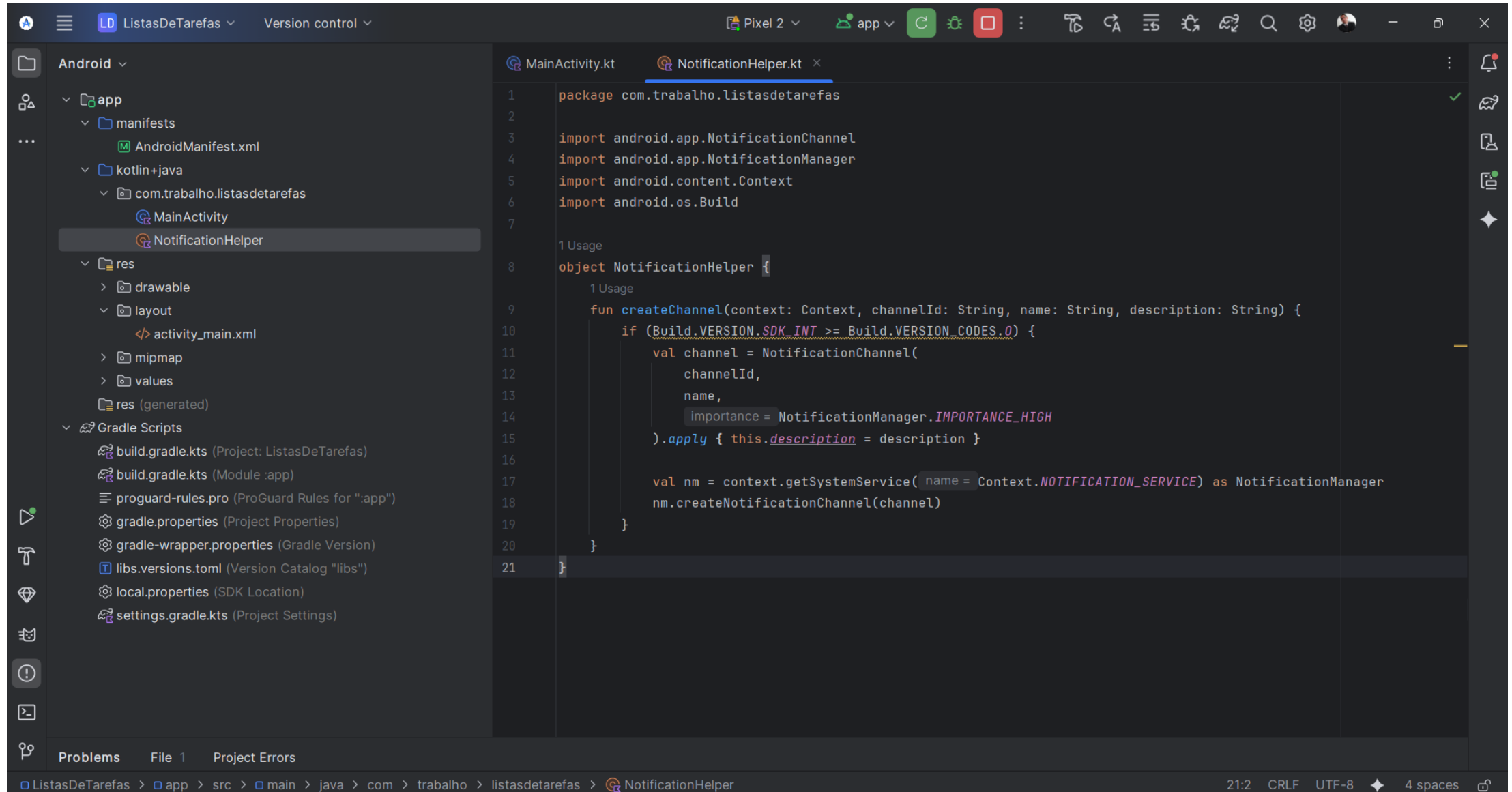
Microatividade 3: Arquivos de lógica e configurações

Resultado da tela:



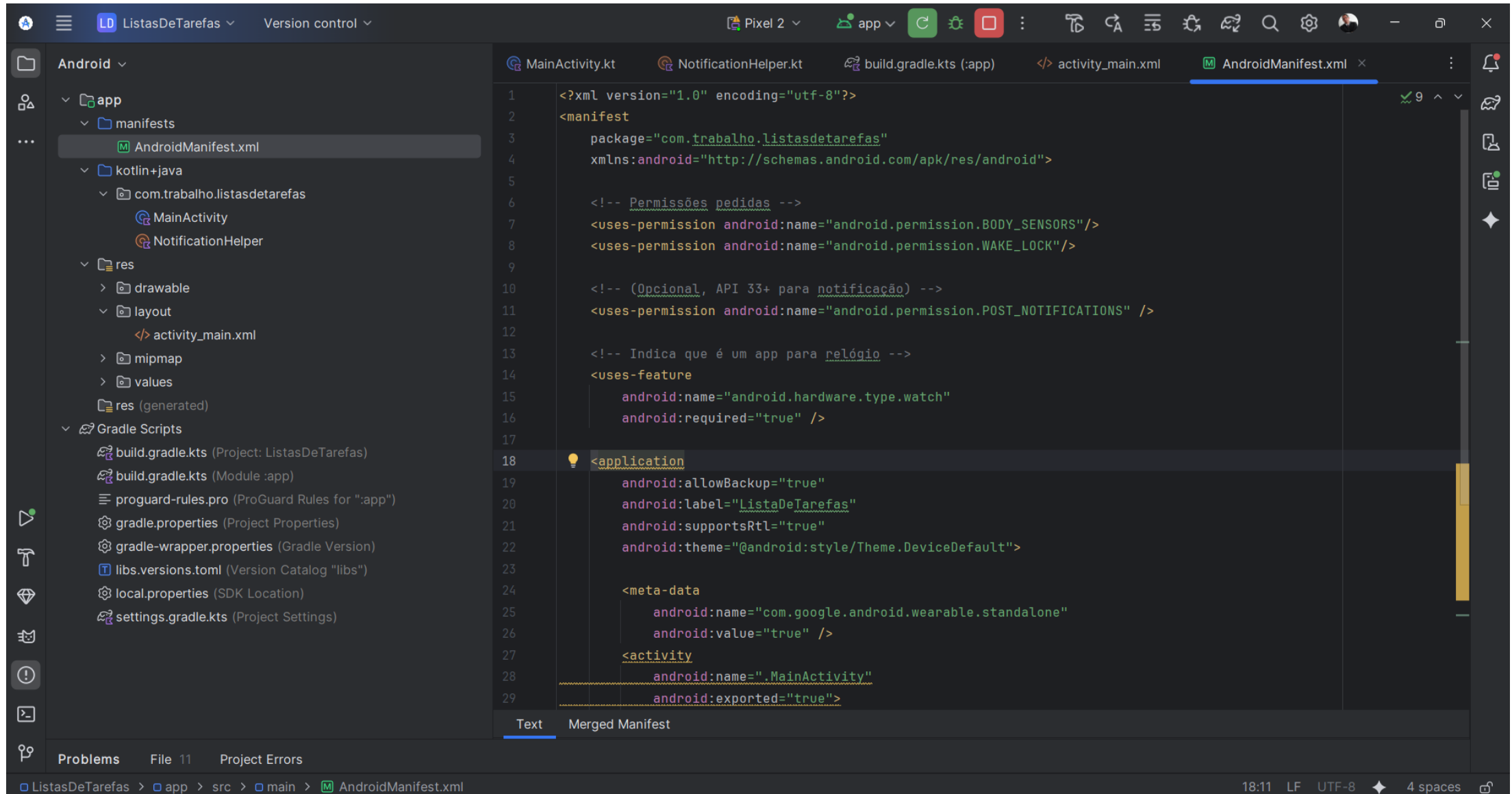
Microatividade 3: Arquivos de lógica e configurações

Resultado da tela:



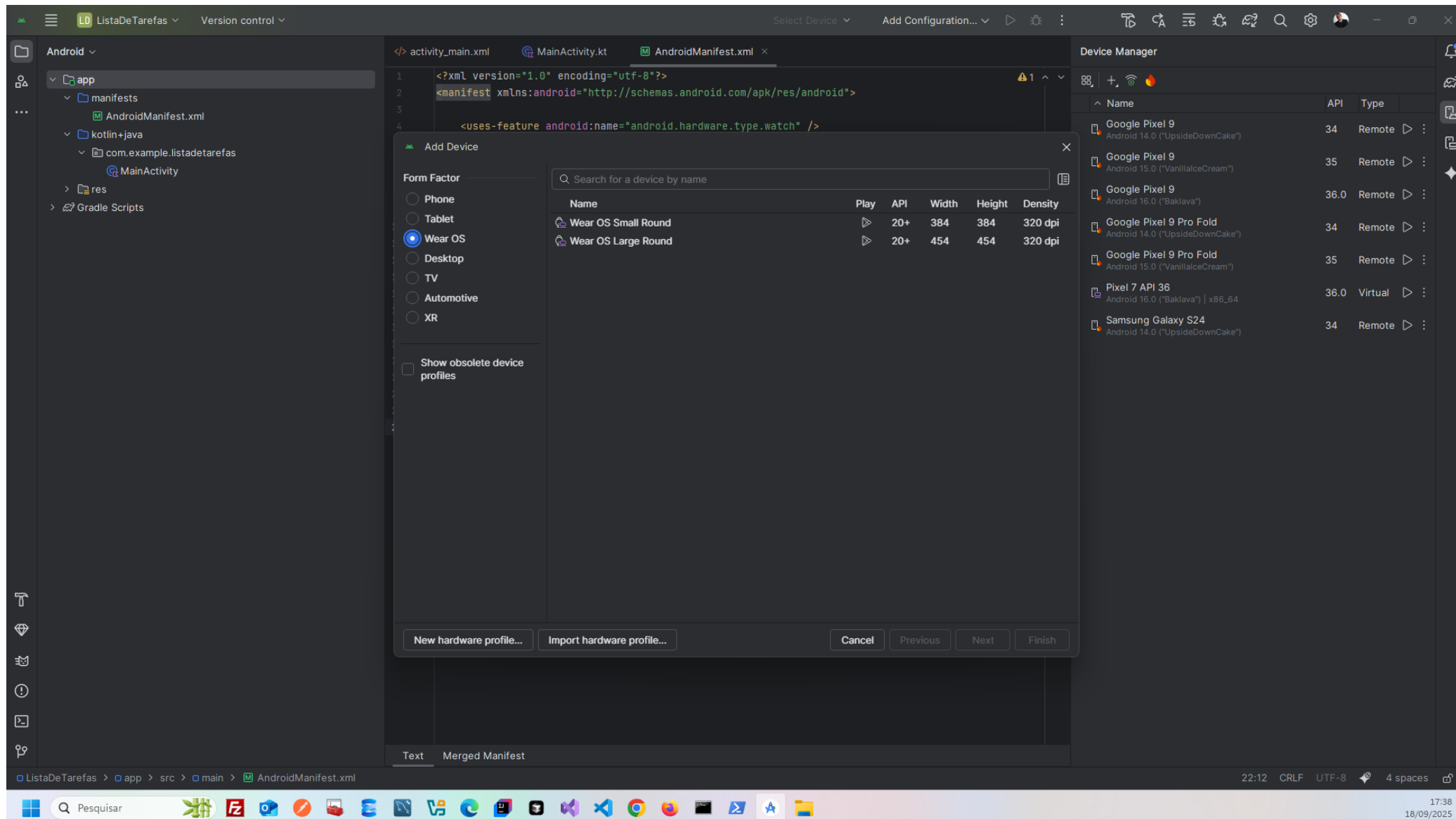
Microatividade 3: Arquivos de lógica e configurações

Resultado da tela:



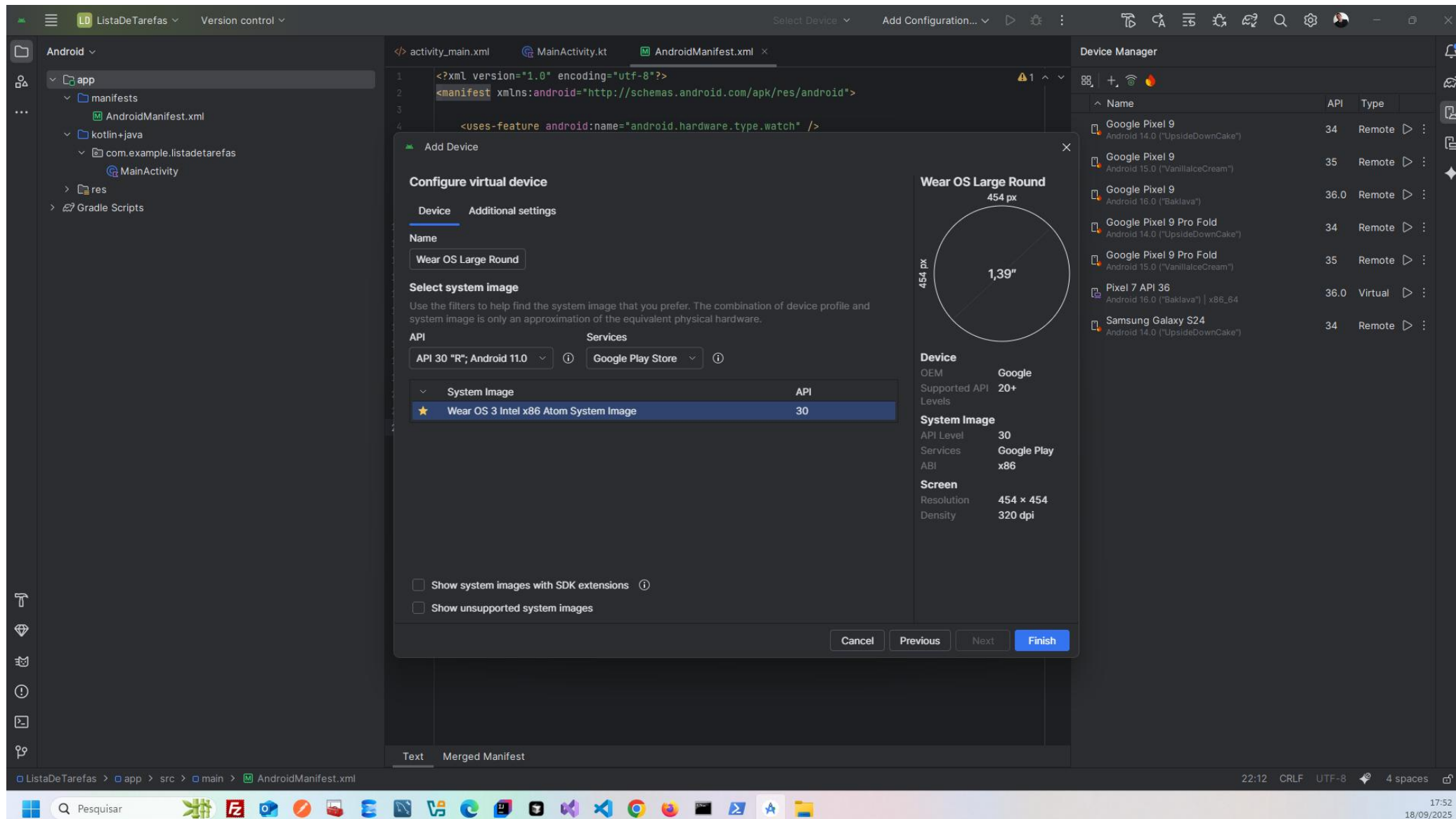
Microatividade 4: Criando um emulador

Resultado da tela:



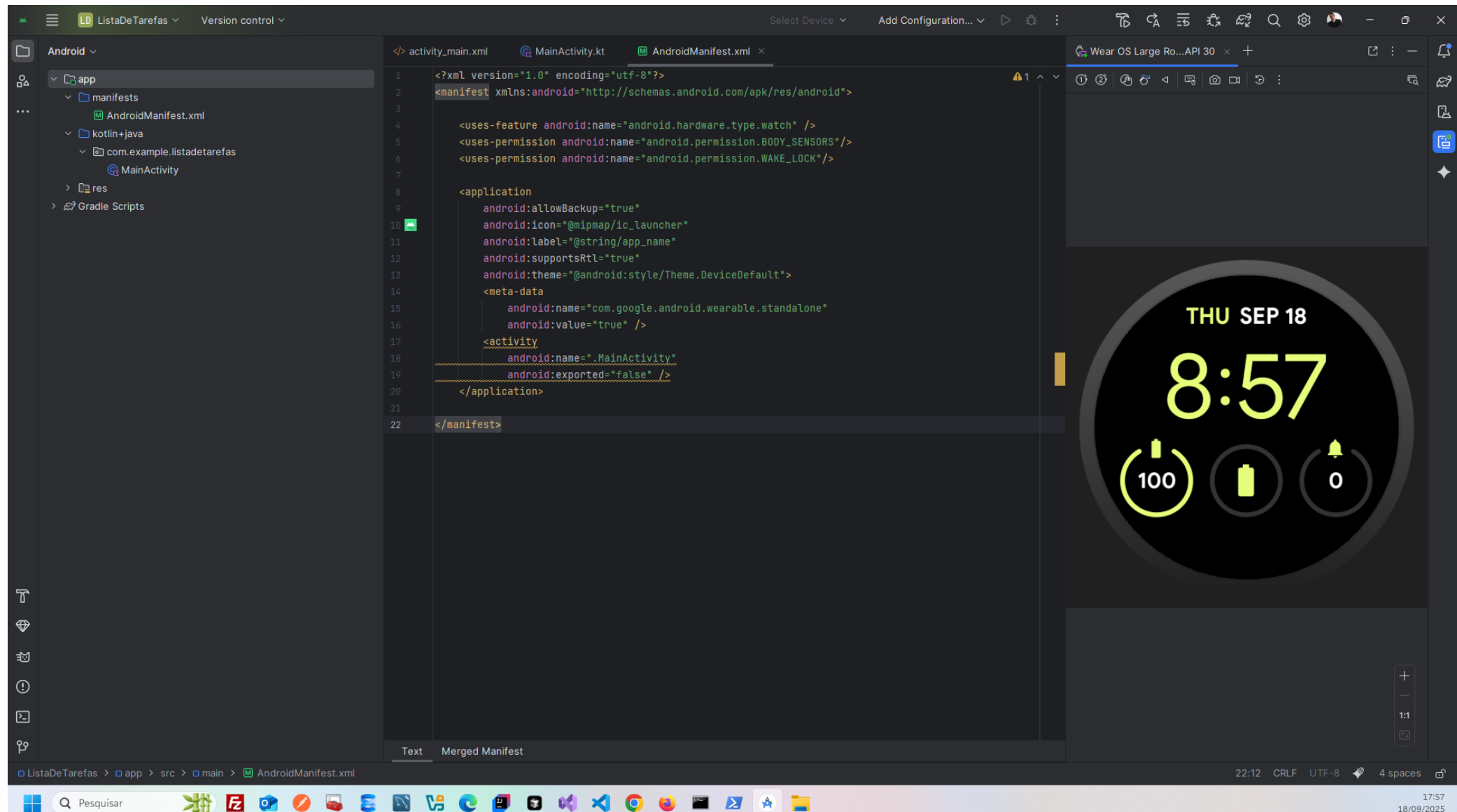
Microatividade 4: Criando um emulador

Resultado da tela:



Microatividade 5: Capturas de tela

Resultado da tela:



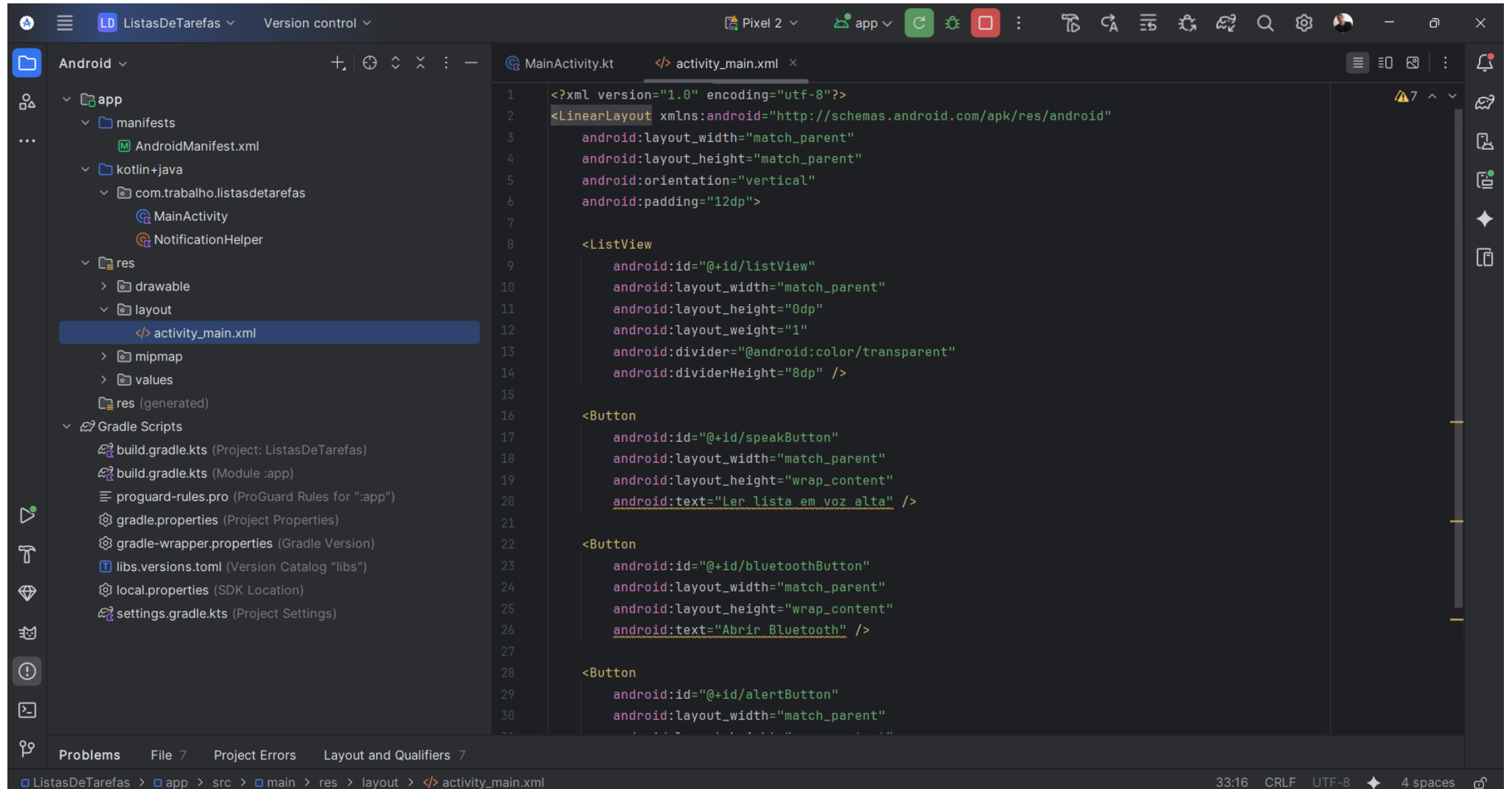
Microatividade 5: Capturas de tela

Resultado da tela:



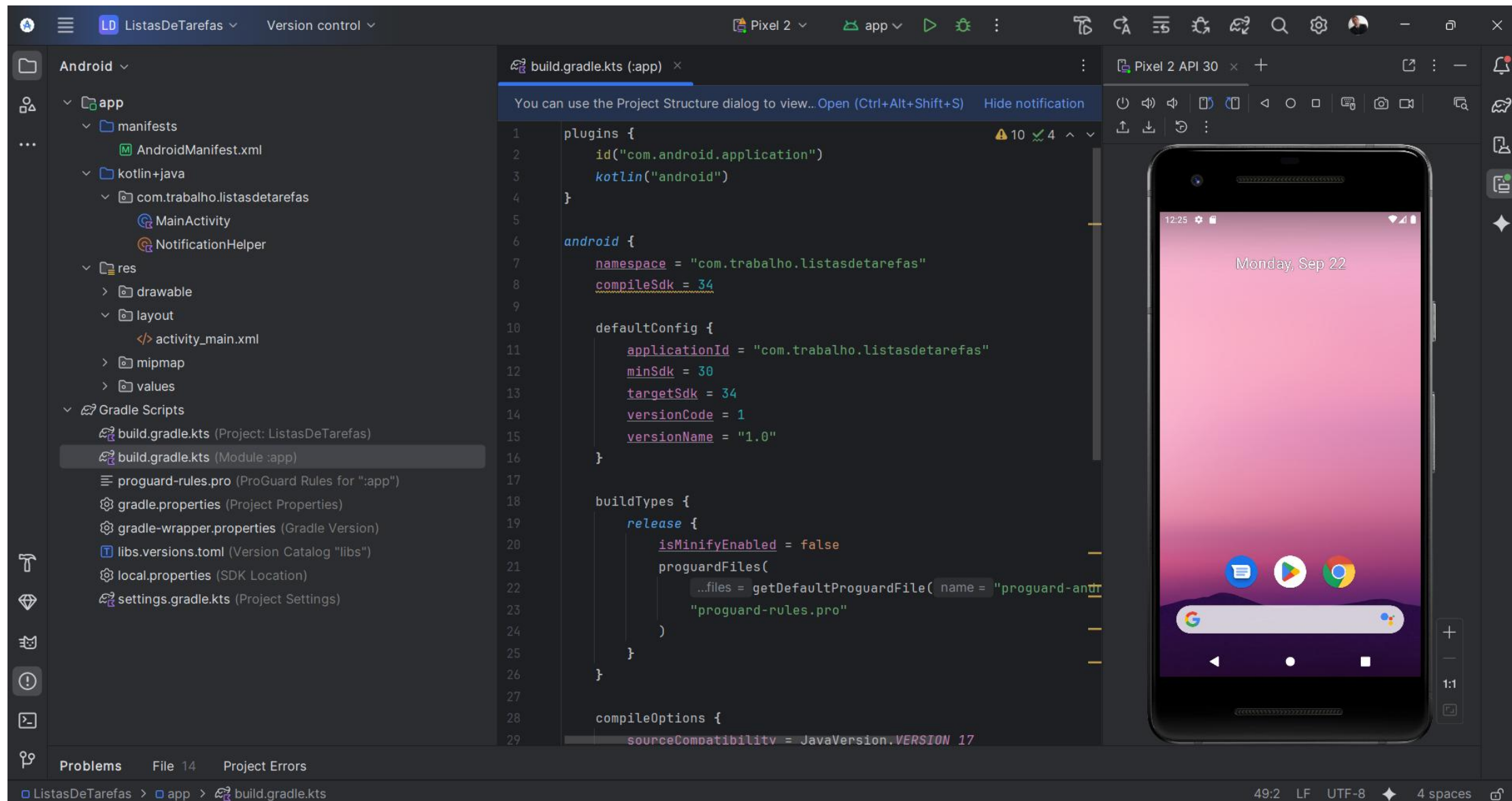
Trabalho Prático

Resultado da tela:



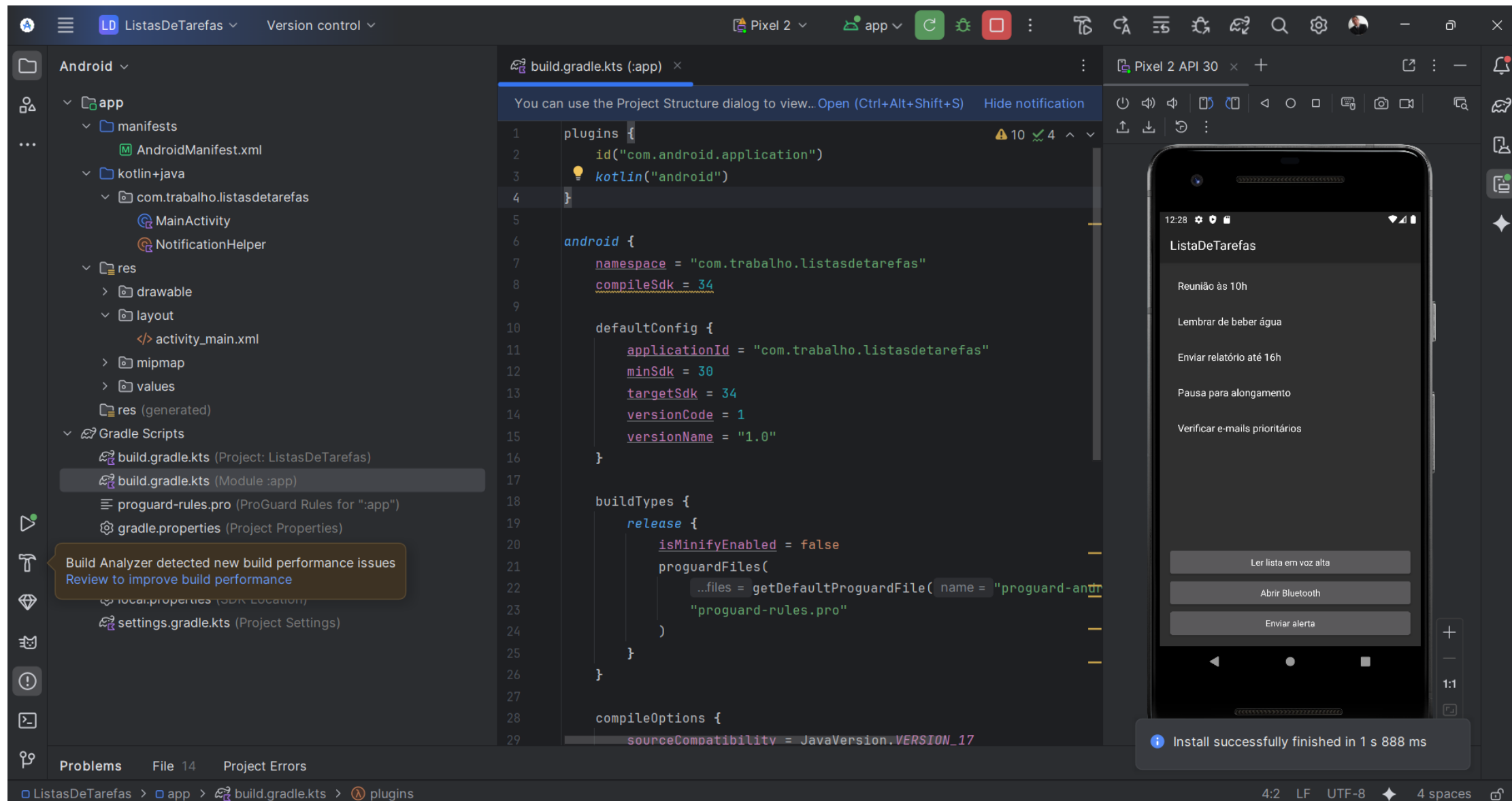
Trabalho Prático

Resultado da tela:



Trabalho Prático

Resultado da tela:



Relatório – Microatividades 2 a 5 e Trabalho Final (Microatividade 1 feita no trabalho anterior)

Microatividade 2

- Configuração do ambiente no Android Studio e SDKs do Wear OS.
- Criação do projeto “ListaDeTarefas” (módulo app) para Wear OS.
- Ajuste inicial do Gradle do módulo: compileSdk 34, targetSdk 34, minSdk 30, Kotlin/Java 17.
- Estrutura básica: MainActivity, layout com ListView e botões (Ler, Bluetooth, Alerta).

Microatividade 3

- Implementação do TextToSpeech em PT-BR.
- Ações dos botões e do item da lista: leitura do item selecionado e da lista completa.
- Tratativas simples de erro de inicialização do TTS.

Microatividade 4

- Criação e execução no Emulador Wear OS (dispositivo round).
- Sincronização do Gradle e rebuild do projeto.
- Criação do canal de notificação via classe auxiliar (NotificationHelper).

Microatividade 5

- Geração das capturas de tela com o app em execução no relógio (emulador e/ou relógio físico).
- Armazenamento das imagens para a documentação final.

Trabalho final

- **Consolidação das funcionalidades:**
 1. Lista de tarefas com leitura por voz (TTS).
 2. Notificação de alerta com canal dedicado.
 3. Atalho para configurações de Bluetooth.
- Ajustes no Manifest (incluindo POST_NOTIFICATIONS para API 33+ quando necessário).
- Dependências essenciais adicionadas (androidx.core, activity-ktx, wear).
- Build sem erros e testes funcionais no emulador.
- Preparação do repositório e documentação do projeto.

Conclusão

O aplicativo Wear OS foi configurado, implementado e testado com sucesso. Os principais desafios estiveram nos callbacks de áudio e no envio de notificações, resolvidos ao simplificar o callback e usar o NotificationManager do framework com canal de notificação. O app cumpre os objetivos propostos: lista de tarefas, leitura por voz, alerta e atalho para Bluetooth. Como melhorias futuras, podem ser incluídos gerenciamento completo das tarefas (adicionar/editar/excluir), solicitação de permissões em tempo de execução quando aplicável e, se necessário, reintrodução do monitoramento de dispositivos de áudio com compatibilidade garantida.

Repositório do GitHub

<https://github.com/mauriciocampos1234/ListasDeTarefas>