

ANHANGUERA EDUCACIONAL
SCRN (ASA NORTE) 708/709 - BLOCO B, S/N, ASA NORTE - BRASÍLIA – DF

SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

AUTOR(ES):
LUCAS LOPES RIBEIRO

DESENVOLVIMENTO MOBILE
Construir a interface de uma aplicação Android.

BRASÍLIA
2025
Lucas Lopes Ribeiro

DESENVOLVIMENTO MOBILE
Construir a interface de uma aplicação Android.

Portifólio de Desenvolvimento Mobile apresentado ao(s) tutor(es): André Martins e Vinicius Mendes Gomes da Silva, do curso de Tecnologia em Análise e Desenvolvimento de Sistemas da instituição Anhanguera Educacional.

BRASÍLIA

2025

SUMÁRIO

1 INTRODUÇÃO.....	4
2 DESENVOLVIMENTO.....	5
3 CONCLUSÃO.....	11

1 INTRODUÇÃO

Neste trabalho, será desenvolvida uma aplicação mobile utilizando o software Android Studio, com o objetivo de criar um aplicativo para a HamburgueriaZ. O desenvolvimento será realizado na linguagem Java, utilizando o Software Development Kit (SDK) específico para Android, garantindo compatibilidade e eficiência na implementação das funcionalidades do sistema.

O projeto tem como finalidade proporcionar uma experiência intuitiva e dinâmica para os usuários, permitindo a realização de pedidos, a consulta ao cardápio e demais interações essenciais para o funcionamento da hamburgueria. A estrutura do aplicativo seguirá boas práticas de desenvolvimento, assegurando usabilidade, acessibilidade e segurança.

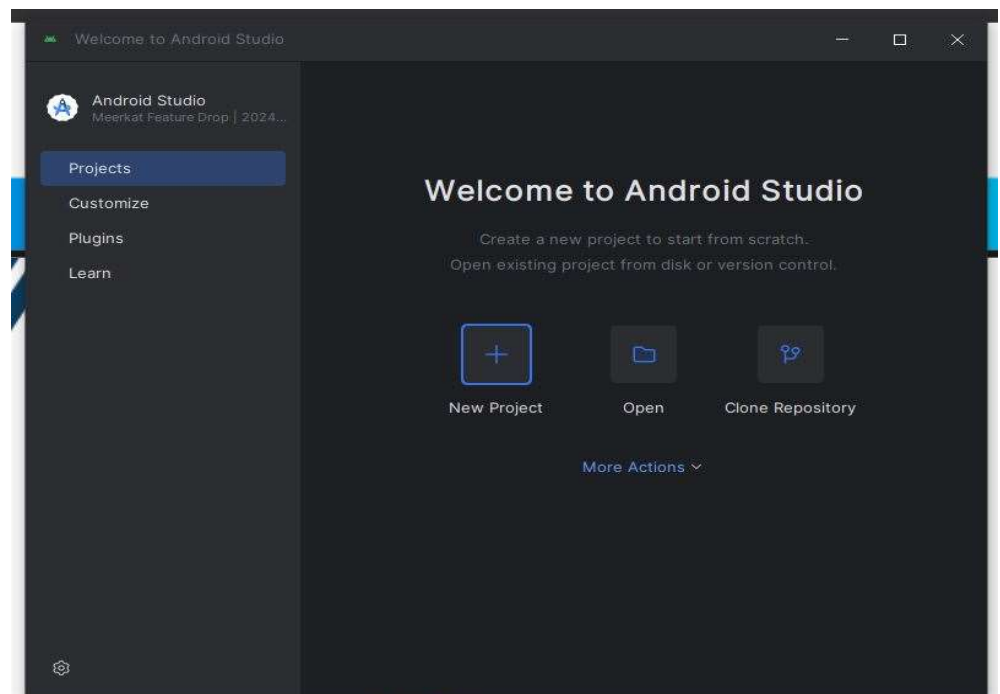
Dessa forma, este relatório apresentará as etapas seguidas no desenvolvimento do aplicativo, abordando os principais conceitos utilizados, desafios enfrentados e soluções implementadas ao longo do processo.

2 DESENVOLVIMENTO

Para o desenvolvimento do aplicativo **hamburgueriaz**, foi utilizada a versão API 24 (Android 7.0 - Nougat) como Minimum SDK. A escolha dessa versão se deve ao fato de oferecer melhorias significativas, como suporte a multitarefas, otimização do consumo de bateria e novas APIs de notificações. Além disso, ao adotar a API 24, o aplicativo mantém compatibilidade com uma ampla gama de dispositivos, sem comprometer o acesso a recursos modernos do sistema.

Outro aspecto relevante da configuração do projeto é o uso da linguagem Groovy DSL para a definição das configurações no Gradle. Essa linguagem é amplamente utilizada em projetos Android e permite uma estrutura de configuração simples e eficiente, garantindo integração com ferramentas de build e gerenciamento de dependências.

Segundo o raciocínio, abaixo é demonstrado as imagens mostrando o software já instalado e o projeto sendo iniciado:



New Project

Empty Views Activity

Creates a new empty activity

Name

Package name

Save location

Language

Minimum SDK

Your app will run on approximately 98,6% of devices.
[Help me choose](#)

Build configuration language

The application name for most apps begins with an uppercase letter

New Project

Templates

- Phone and Tablet
- Wear OS
- Television
- Automotive

No Activity

Empty Activity

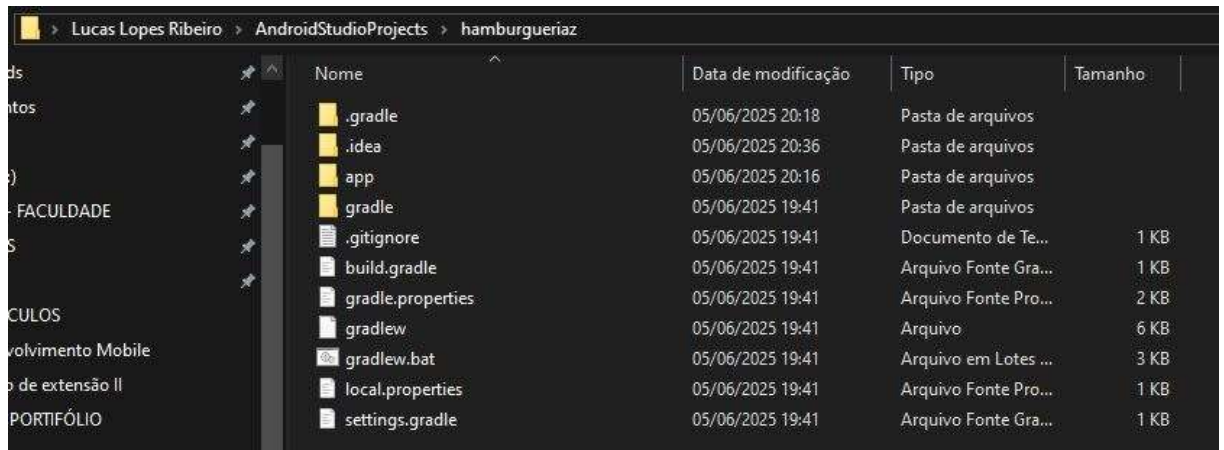
Gemini API Starter

Basic Views Activity

Bottom Navigation Views Activity

Empty Views Activity

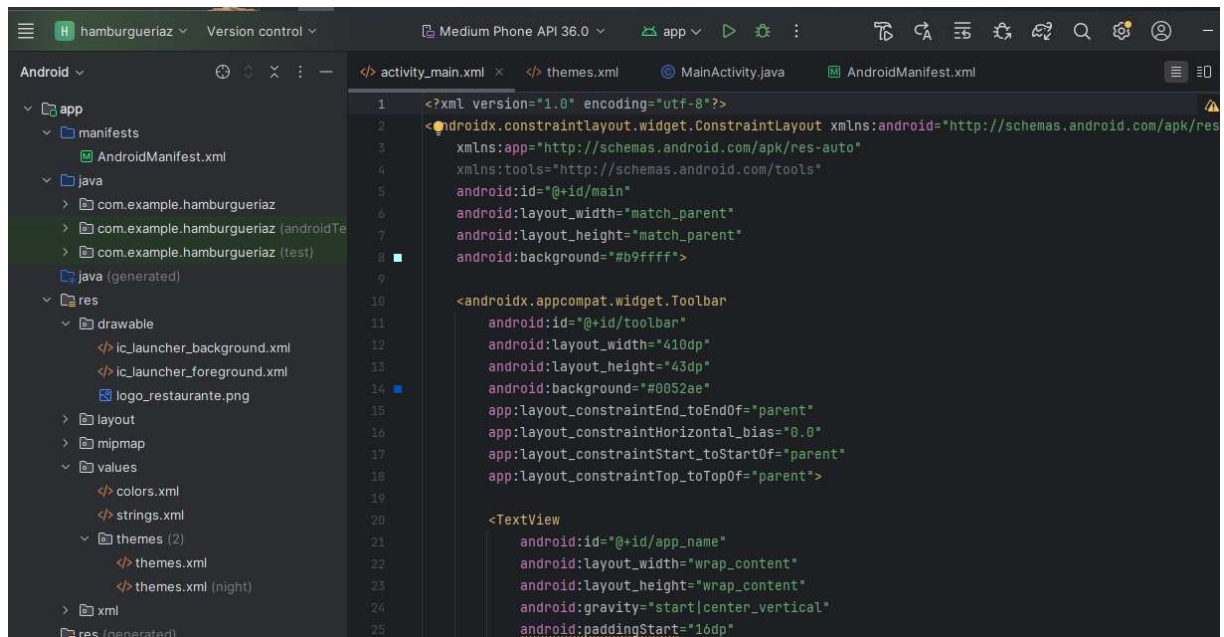
O caminho da pasta com o projeto:



Nome	Data de modificação	Tipo	Tamanho
.gradle	05/06/2025 20:18	Pasta de arquivos	
.idea	05/06/2025 20:36	Pasta de arquivos	
app	05/06/2025 20:16	Pasta de arquivos	
gradle	05/06/2025 19:41	Pasta de arquivos	
.gitignore	05/06/2025 19:41	Documento de Te...	1 KB
build.gradle	05/06/2025 19:41	Arquivo Fonte Gra...	1 KB
gradle.properties	05/06/2025 19:41	Arquivo Fonte Pro...	2 KB
gradlew	05/06/2025 19:41	Arquivo	6 KB
gradlew.bat	05/06/2025 19:41	Arquivo em Lotes ...	3 KB
local.properties	05/06/2025 19:41	Arquivo Fonte Pro...	1 KB
settings.gradle	05/06/2025 19:41	Arquivo Fonte Gra...	1 KB

Com tudo pronto e organizado, é iniciada a fase de construção do código e da lógica aplicada, como é possível ver nas imagens a baixo. A cadeia de arquivos e pastas juntamente com o código nas abas Activity_main.xml e MainActivity.java, juntamente com a aba themes.xml.

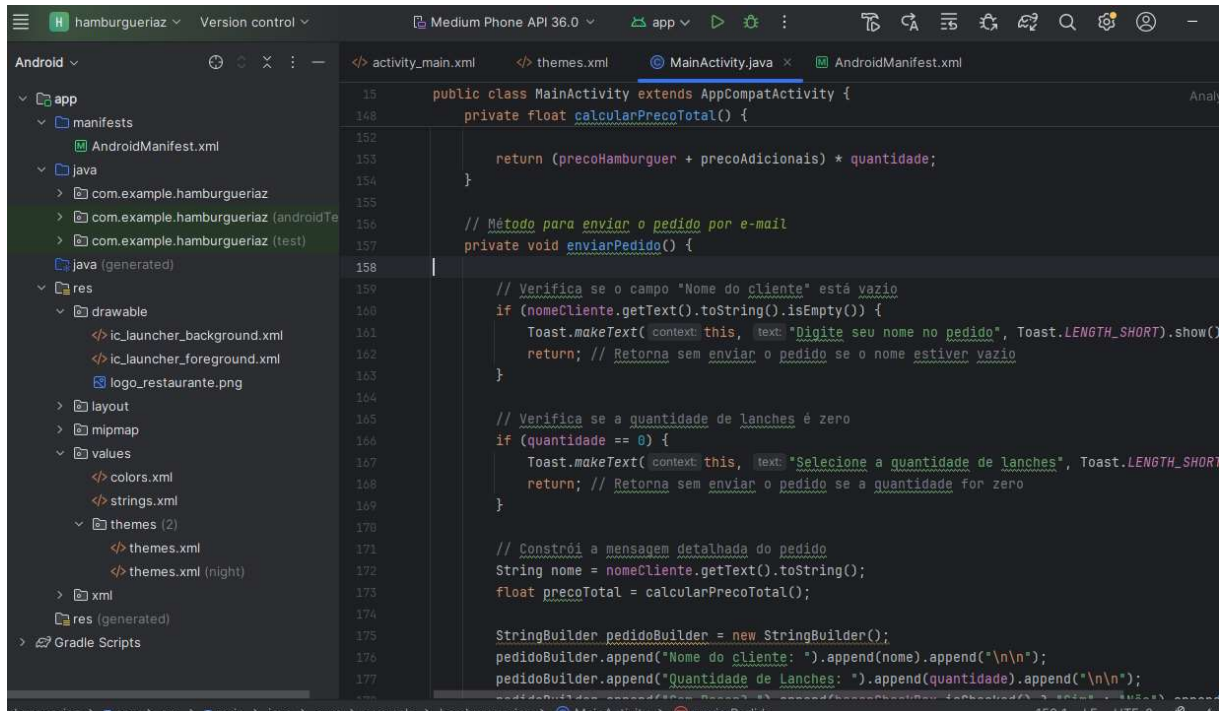
activity_main.xml:



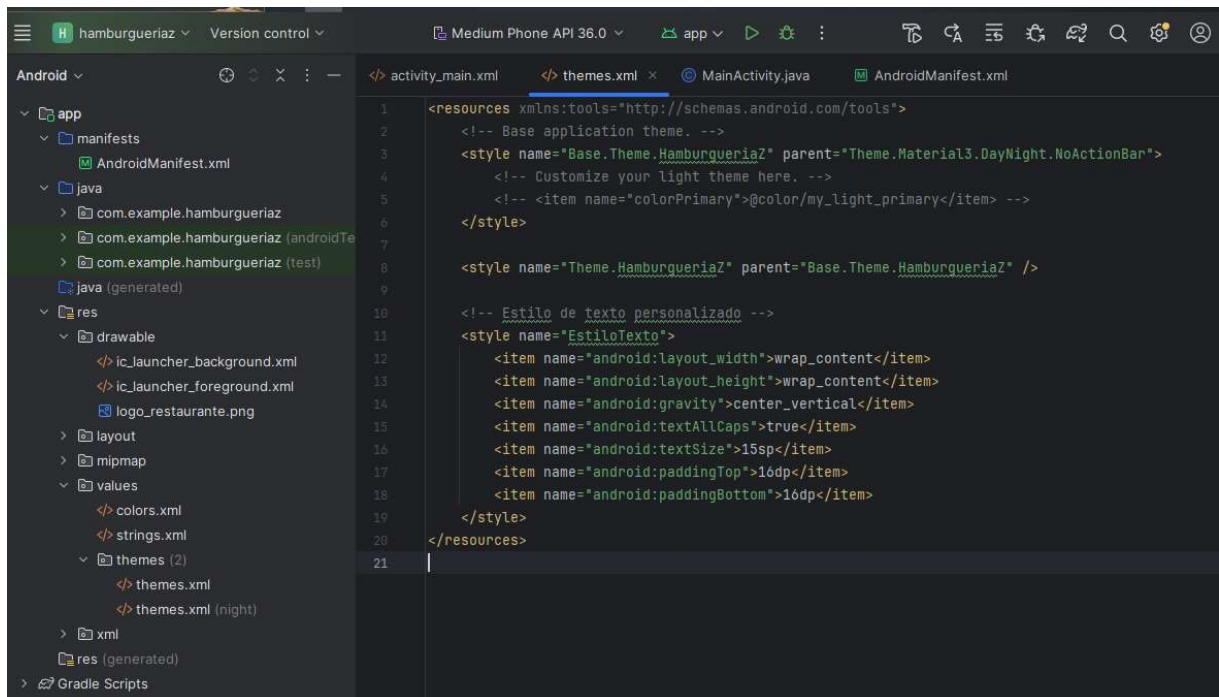
```

1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:id="@+id/main"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      android:background="#b9ffff">
9
10     <androidx.appcompat.widget.Toolbar
11         android:id="@+id/toolbar"
12         android:layout_width="410dp"
13         android:layout_height="43dp"
14         android:background="#0052ae"
15         app:layout_constraintEnd_toEndOf="parent"
16         app:layout_constraintHorizontal_bias="0.0"
17         app:layout_constraintStart_toStartOf="parent"
18         app:layout_constraintTop_toTopOf="parent">
19
20     <TextView
21         android:id="@+id/app_name"
22         android:layout_width="wrap_content"
23         android:layout_height="wrap_content"
24         android:gravity="start|center_vertical"
25         android:paddingStart="16dp"
  
```

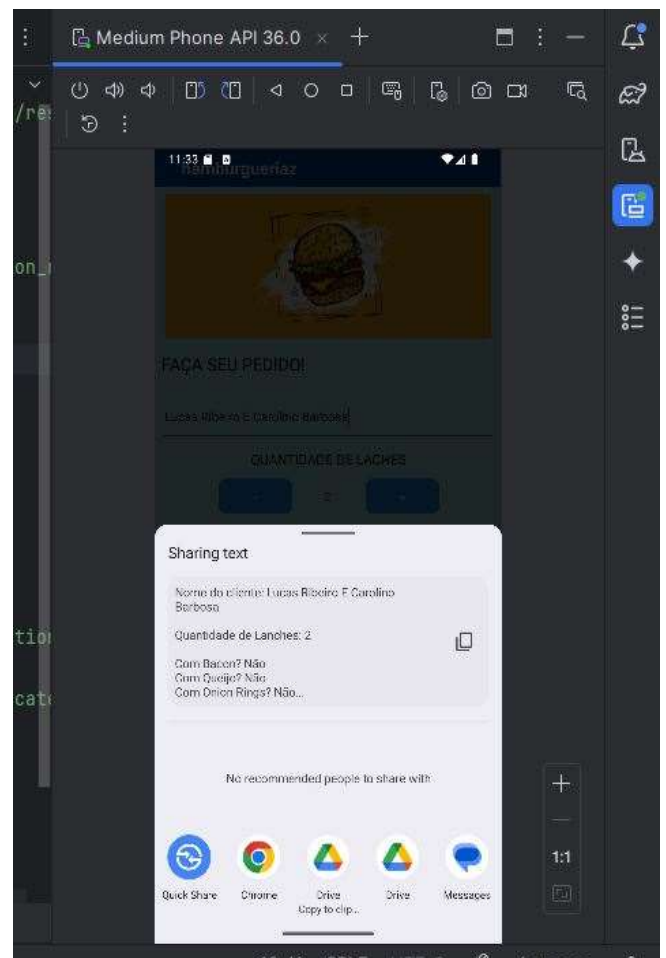
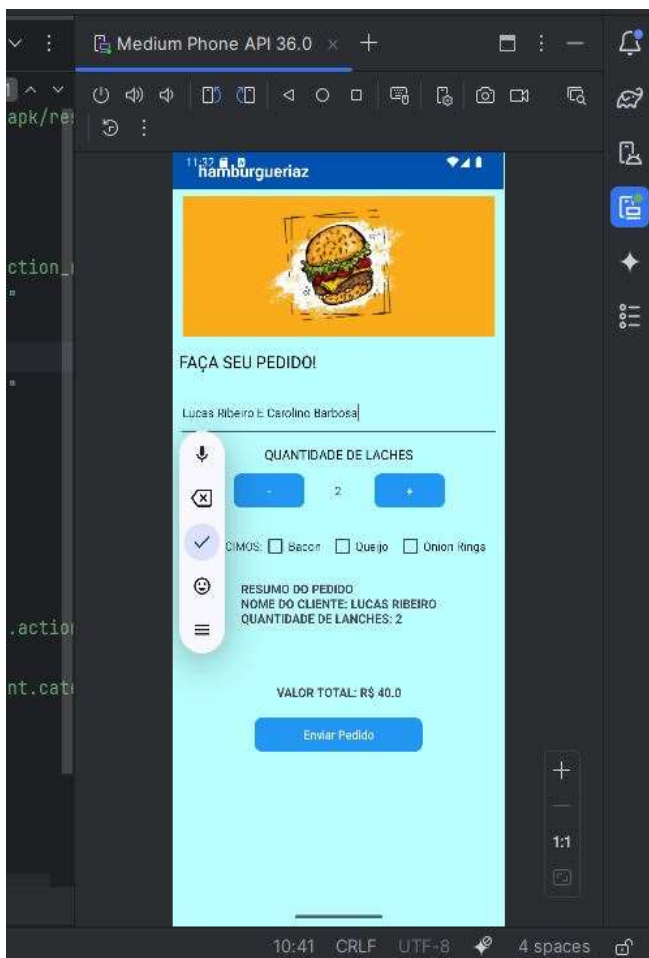
MainActivity.java:



Themes.xml:

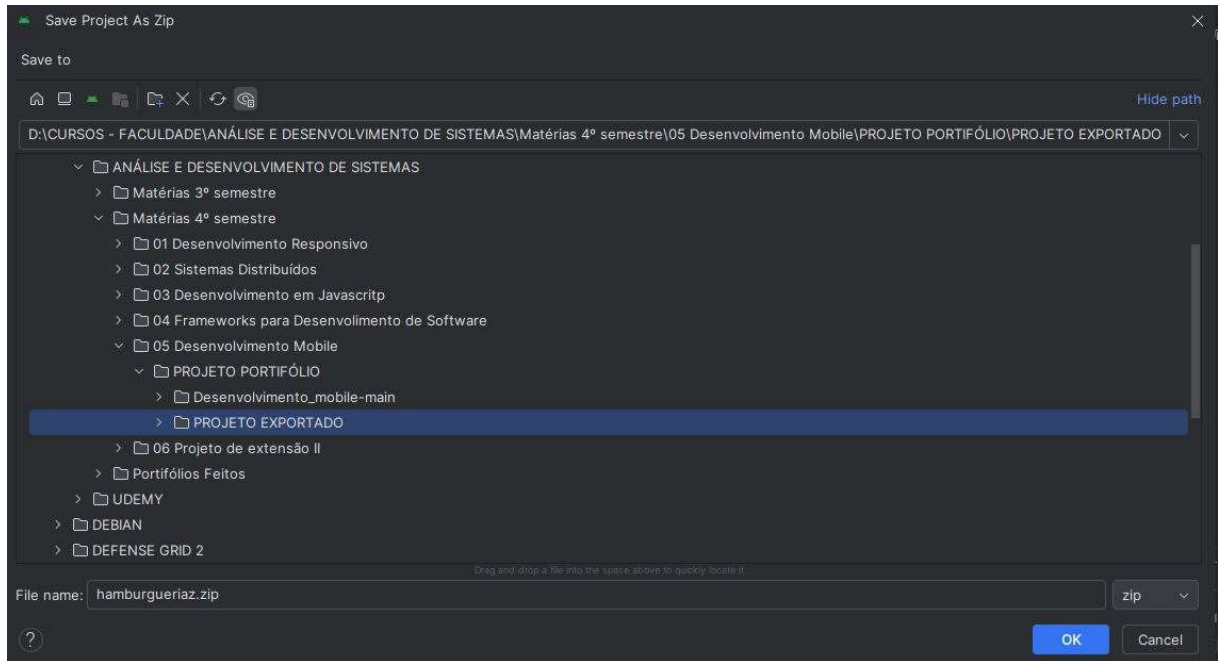


Seguindo o raciocínio, a próxima etapa é clicar no botão “Run App” e ver se Android Studio consegue fazer a build do projeto. Uma tela de carregamento com o emulador do SmartPhone aparece e começa a instalação do aplicativo. No caso deste projeto, a tela do aplicativo aparece e é possível preencher os campos do pedido do lanche, onde foi simulado o pedido para um casal. De acordo com as imagens abaixo:

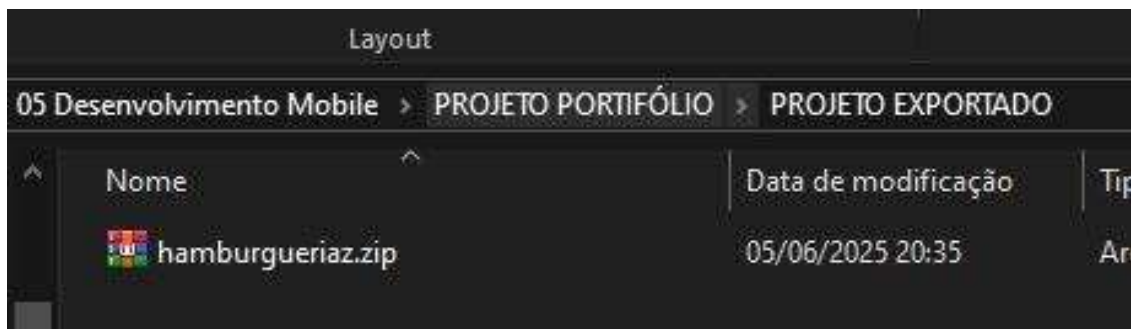


Como demonstrado até aqui, o projeto hamburgueriaZ foi concluído e funciona corretamente. Então o passo final do tutorial desta atividade é exportar o arquivo do projeto e finalizar.

Exportando arquivo:



Arquivo salvo na pasta:



3 CONCLUSÃO

O desenvolvimento do HamburgueriaZ foi realizado de maneira estruturada conforme orientação do arquivo tutorial do portfólio. Desde a configuração inicial no Android Studio até a implementação das funcionalidades a API 24, o aplicativo possui botões para adicionar e remover acompanhamentos e campo de preenchimento de informações. Assim, foram criadas e ajustadas interfaces visuais e funcionalidades lógicas, como a seleção de quantidade e o envio do pedido. Após a construção do código, o projeto passou pela fase de testes, sendo executado no emulador para validar seu funcionamento.

Com tudo concluído e rodando corretamente, o último passo foi a exportação do projeto. O resultado é um aplicativo funcional, intuitivo e pronto para uso