

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.

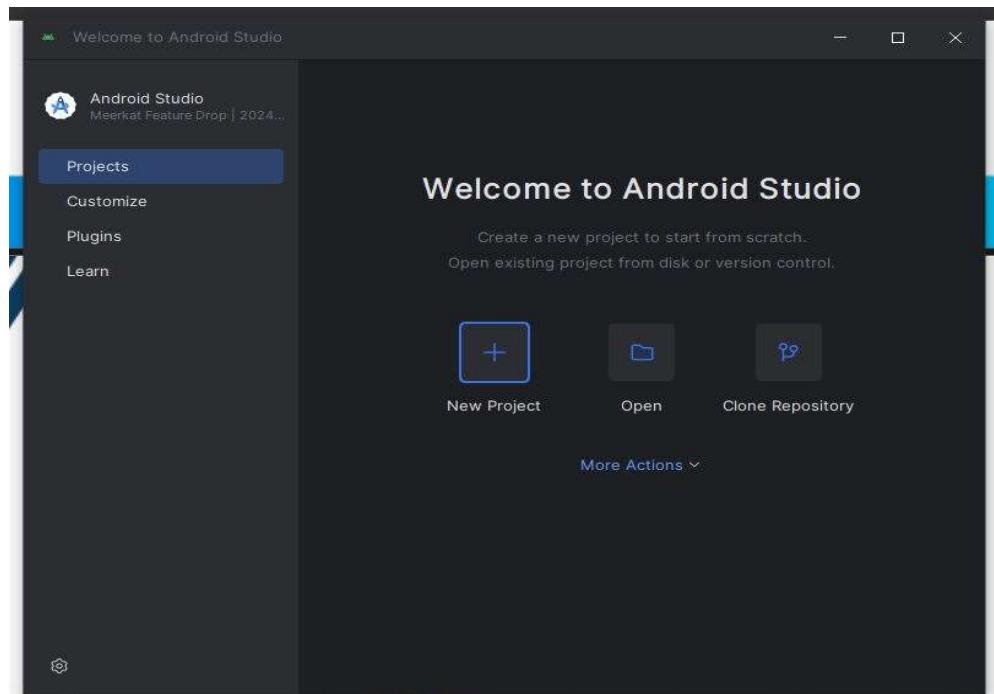
Dessa forma, este relatório apresentará as etapas seguidas no desenvolvimento do aplicativo, abordando os principais conceitos utilizados, desafios enfrentados e soluções implementadas ao longo do processo.

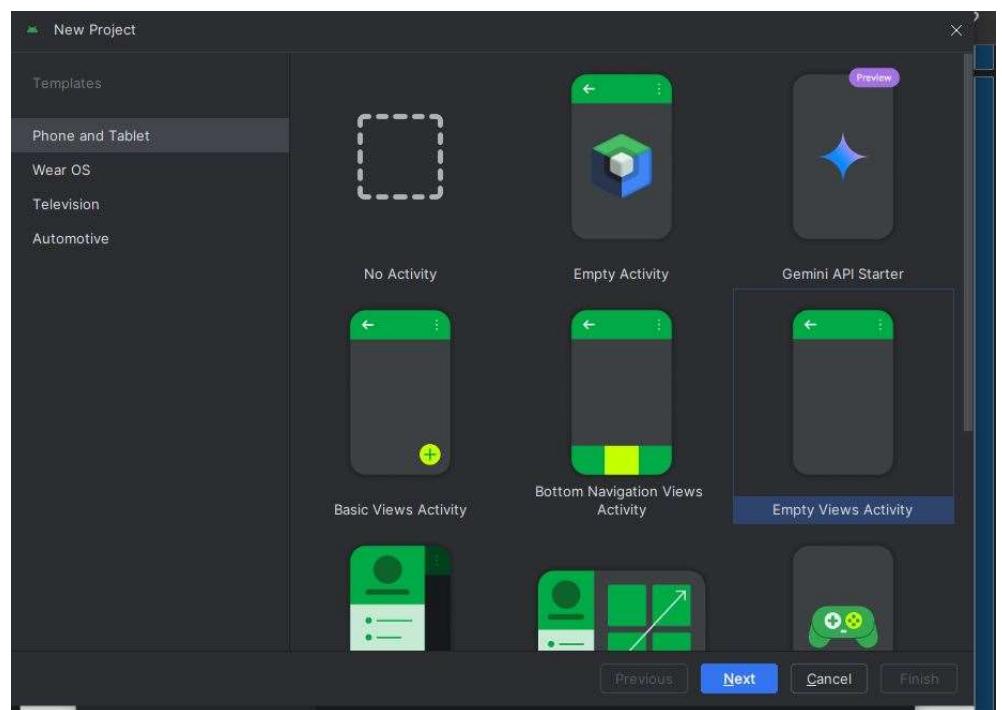
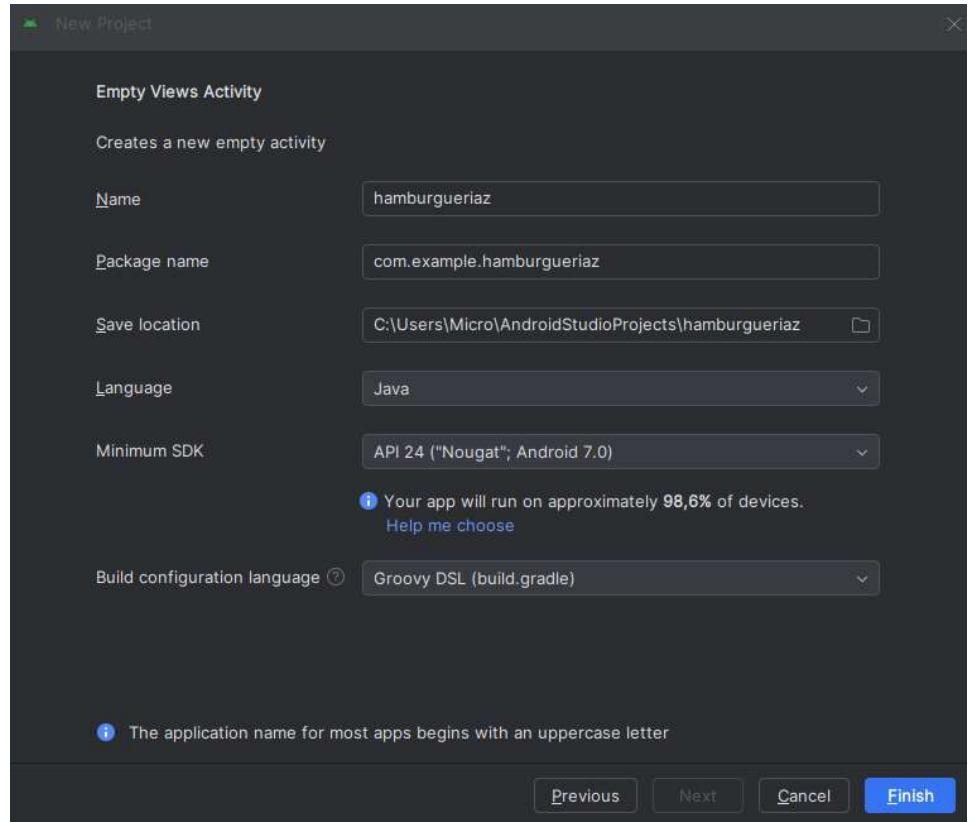
2 DESENVOLVIMENTO

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.

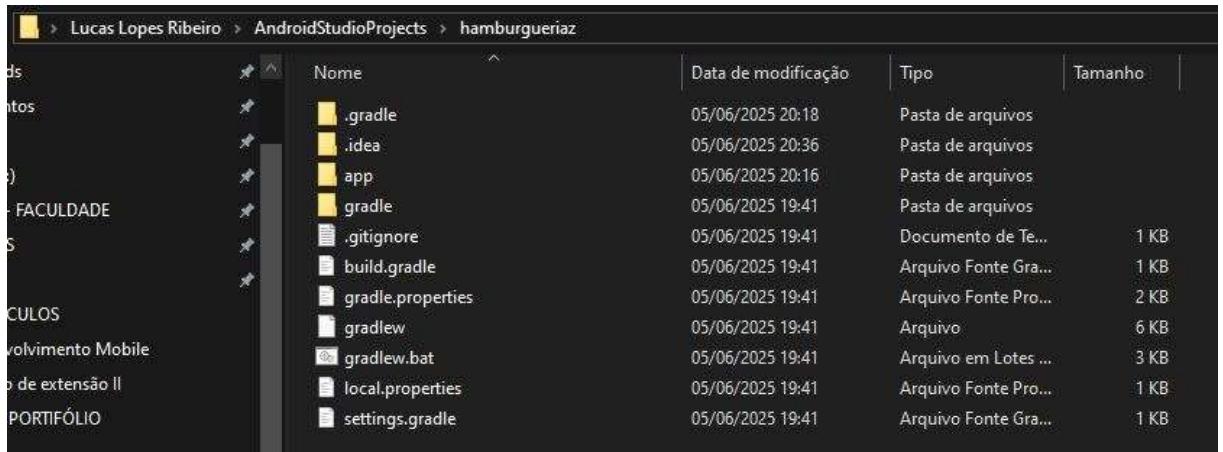
Outro aspecto relevante da configuração do projeto é o uso da linguagem Groovy DSL para a definição das configurações no Gradle. Essa linguagem é amplamente utilizada em projetos Android e permite uma estrutura de configuração simples e eficiente, garantindo integração com ferramentas de build e gerenciamento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

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





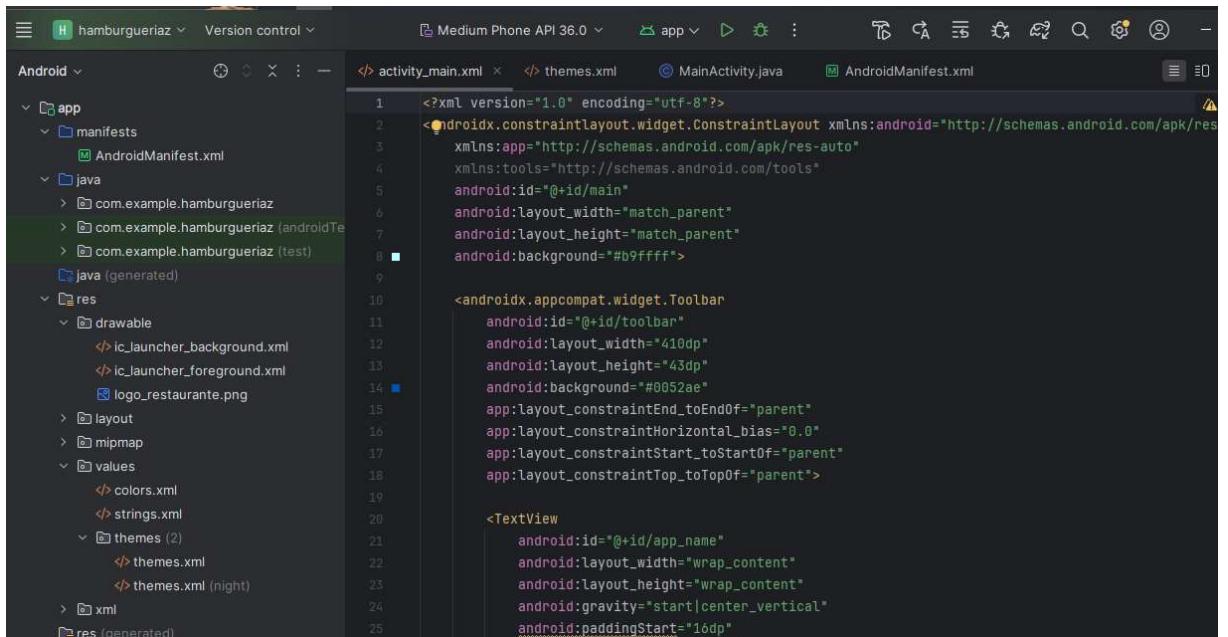
O caminho da pasta com o projeto:



	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 Texto	1 KB
	build.gradle	05/06/2025 19:41	Arquivo Fonte Gráfica	1 KB
	gradle.properties	05/06/2025 19:41	Arquivo Fonte Propriedades	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 Propriedades	1 KB
	settings.gradle	05/06/2025 19:41	Arquivo Fonte Gráfica	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:



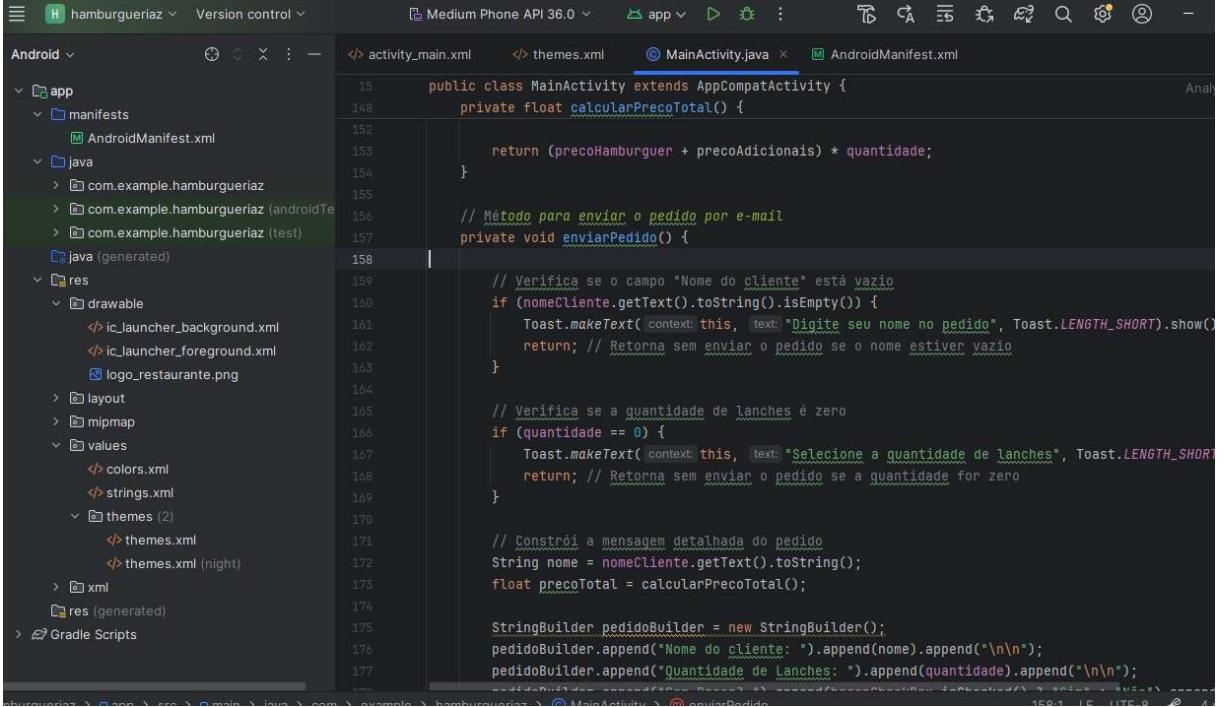
```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#b9ffff">

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="410dp"
        android:layout_height="43dp"
        android:background="#0052ae"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <TextView
            android:id="@+id/app_name"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="start|center_vertical"
            android:paddingStart="16dp"
    
```

MainActivity.java:

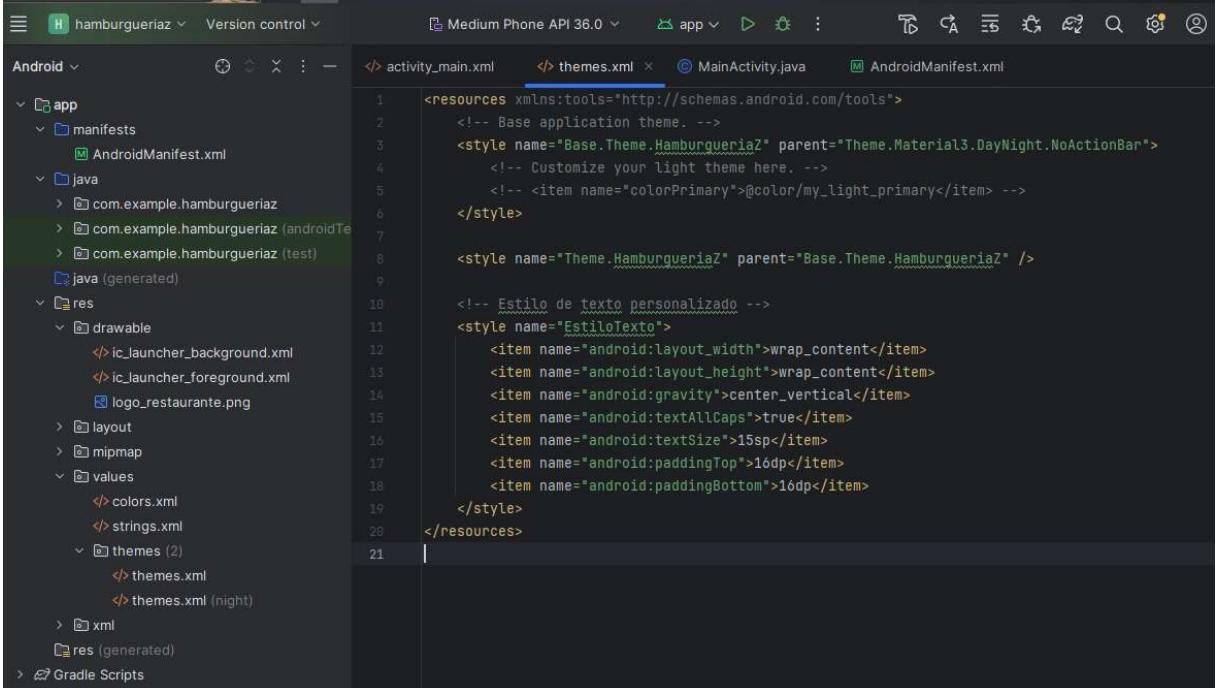


```

15  public class MainActivity extends AppCompatActivity {
16      private float calcularPrecoTotal() {
17          return (precoHamburguer + precoAdicionais) * quantidade;
18      }
19
20      // Método para enviar o pedido por e-mail
21      private void enviarPedido() {
22
23          // Verifica se o campo "Nome do cliente" está vazio
24          if (nomeCliente.getText().toString().isEmpty()) {
25              Toast.makeText(context, "Digite seu nome no pedido", Toast.LENGTH_SHORT).show();
26              return; // Retorna sem enviar o pedido se o nome estiver vazio
27          }
28
29          // Verifica se a quantidade de lanches é zero
30          if (quantidade == 0) {
31              Toast.makeText(context, "Selecione a quantidade de lanches", Toast.LENGTH_SHORT).show();
32              return; // Retorna sem enviar o pedido se a quantidade for zero
33          }
34
35          // Constrói a mensagem detalhada do pedido
36          String nome = nomeCliente.getText().toString();
37          float precoTotal = calcularPrecoTotal();
38
39          StringBuilder pedidoBuilder = new StringBuilder();
40          pedidoBuilder.append("Nome do cliente: ").append(nome).append("\n\n");
41          pedidoBuilder.append("Quantidade de Lanches: ").append(quantidade).append("\n\n");
42          pedidoBuilder.append("Preço Total: ").append(precoTotal);
43
44          Intent emailIntent = new Intent(Intent.ACTION_SENDTO, Uri.parse("mailto:"));
45          emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Pedido de HamburgueriaZ");
46          emailIntent.putExtra(Intent.EXTRA_TEXT, pedidoBuilder.toString());
47
48          startActivity(Intent.createChooser(emailIntent, "Enviar Pedido"));
49      }
50
51  }

```

Themes.xml:

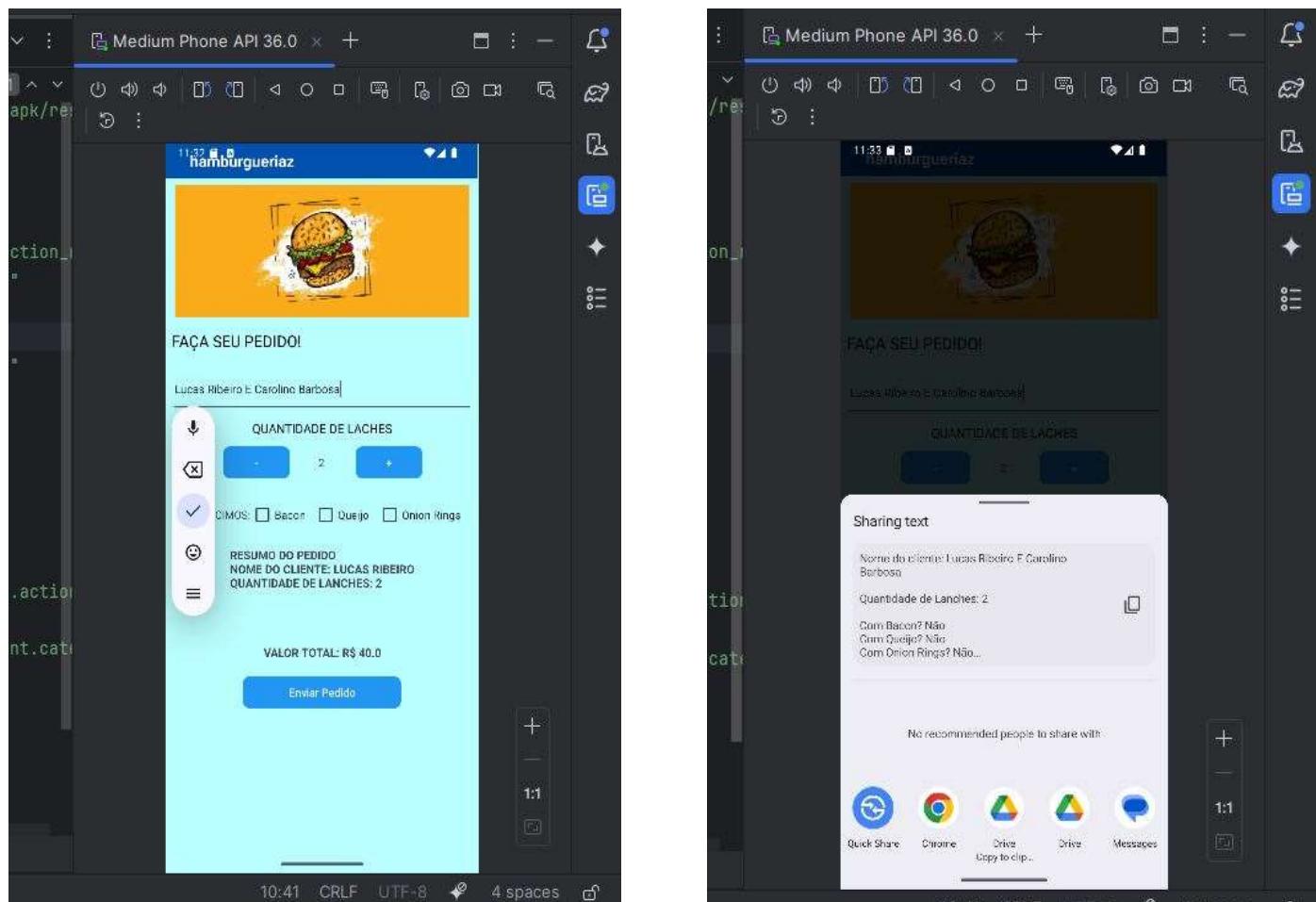


```

1 <resources xmlns:tools="http://schemas.android.com/tools">
2     <!-- Base application theme. -->
3     <style name="Base.Theme.HamburgueriaZ" parent="Theme.Material3.DayNight.NoActionBar">
4         <!-- Customize your light theme here. -->
5         <item name="colorPrimary">@color/my_light_primary</item>
6     </style>
7
8     <style name="Theme.HamburgueriaZ" parent="Base.Theme.HamburgueriaZ" />
9
10    <!-- Estilo de texto personalizado -->
11    <style name="EstiloTexto">
12        <item name="android:layout_width">wrap_content</item>
13        <item name="android:layout_height">wrap_content</item>
14        <item name="android:gravity">center_vertical</item>
15        <item name="android:textAllCaps">true</item>
16        <item name="android:textSize">15sp</item>
17        <item name="android:paddingTop">10dp</item>
18        <item name="android:paddingBottom">16dp</item>
19    </style>
20
21 </resources>

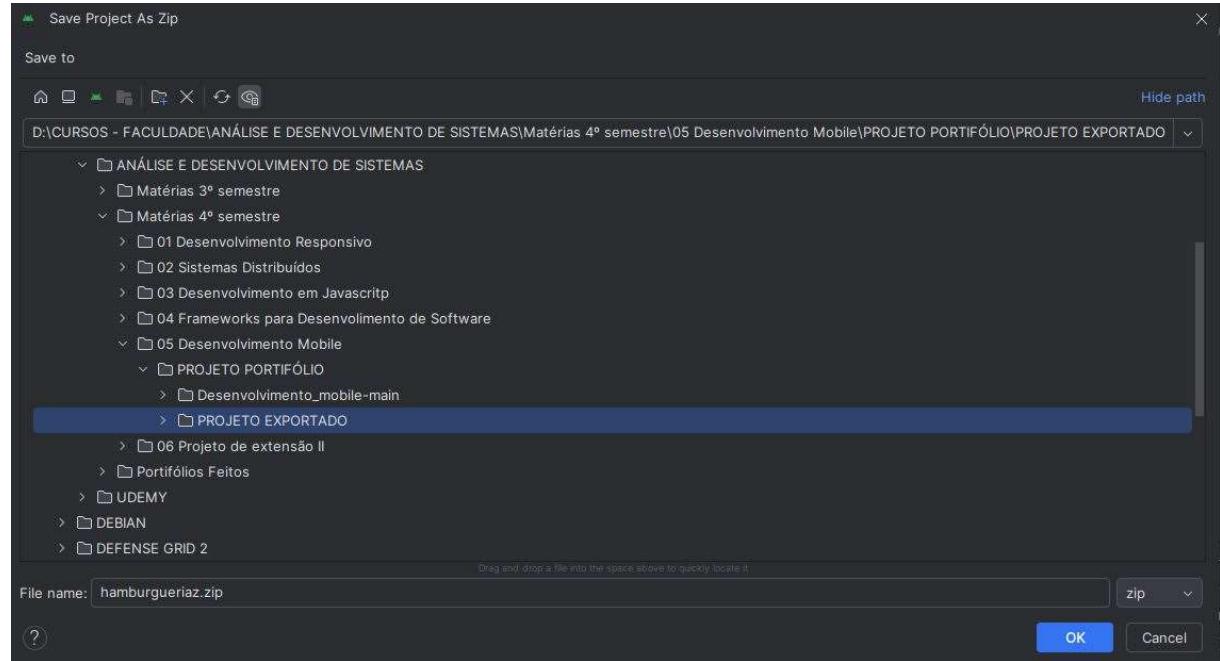
```

Seguindo o raciocínio, a proxima 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:

Layout			
05 Desenvolvimento Mobile > PROJETO PORTIFÓLIO > PROJETO EXPORTADO			
Nome	Data de modificação	Tipo	Ações
hamburgueriaz.zip	05/06/2025 20:35	Arquivo ZIP	

3 CONCLUSÃO

O desenvolvimento do HamburgueriaZ foi realizado de maneira estruturada conforme orientação do arquivo tutorial do portifólio. 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