

# Roteiro Aula Prática



## Desenvolvimento Mobile

## ROTEIRO DE AULA PRÁTICA

NOME DA DISCIPLINA: **Desenvolvimento Mobile**

Escolher um item.

Escolher um item.

### OBJETIVOS

#### Definição dos objetivos da aula prática:

Saber utilizar o software Android Studio;

Saber construir uma interface de aplicação Android, com estilos predefinidos e imagens;

Saber implementar as funcionalidades de um app na linguagem Java;

Saber utilizar os Intents para realizar a interação com aplicativos externos;

Saber exportar um projeto do Android Studio.

### INFRAESTRUTURA

#### Instalações:

Laboratório de Informática

#### Materiais de consumo:

Descrição	Quantid. de materiais por procedimento/atividade
Computador	01 por aluno

#### Software:

Sim ( X ) Não ( )

Em caso afirmativo, qual?

Pago ( ) Não Pago ( X )

Tipo de Licença: Freeware.

#### Descrição do software:

- **Android Studio Flamingo | 2022.2.1:** Ambiente de desenvolvimento integrado (IDE, na sigla em inglês) oficial para o desenvolvimento de apps Android e é baseado no IntelliJ IDEA. Além do editor de código e das ferramentas de desenvolvedor avançadas do IntelliJ, o

Android Studio oferece ainda mais recursos para aumentar sua produtividade na criação de apps Android.

<https://developer.android.com/studio>

- **(Pré-requisito) JAVA JDK 20:** Ambiente de desenvolvimento para construção de aplicativos e componentes usando a linguagem de programação Java.

<https://www.oracle.com/java/technologies/downloads/#java20>

#### **Equipamento de Proteção Individual (EPI):**

NSA.

### **PROCEDIMENTOS PRÁTICOS**

#### **Procedimento/Atividade Nº 1**

##### **Atividade proposta:**

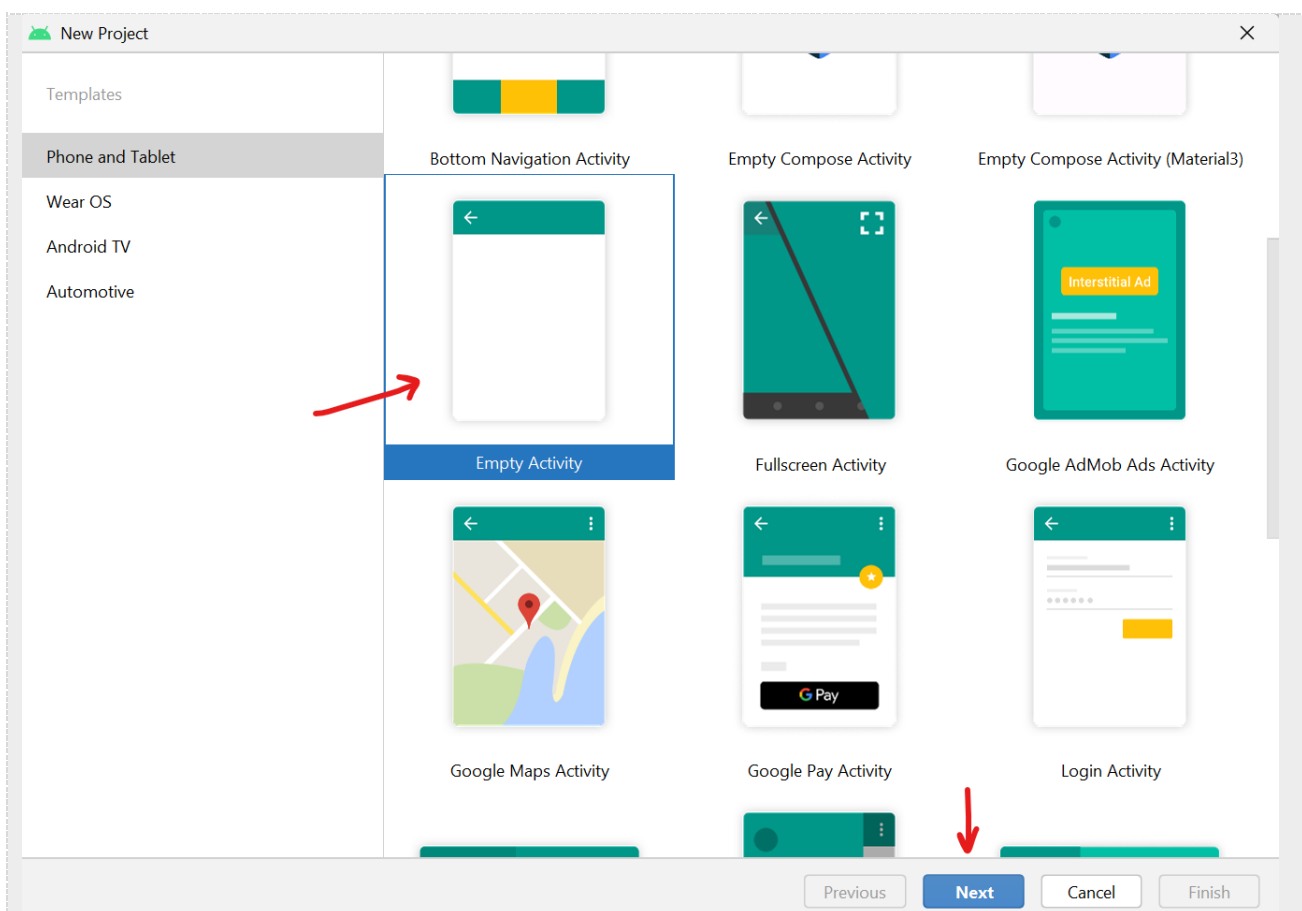
**Construir a interface de uma aplicação Android.**

##### **Procedimentos para a realização da atividade:**

**A atividade está dividida em várias etapas para facilitar seu desenvolvimento!**

##### **Etapas 1: Criando um projeto**

Abra o Android Studio, e no menu superior vá em “File > New > New Project...”. Na janela que abrir, escolha “Empty Activity” e clique em “Next”.



Fonte: elaborado pelo autor

Em seguida, defina o nome do projeto (HamburgueriaZ), escolha a linguagem (no caso desta atividade, escolha Java), e escolha o SDK de desenvolvimento (API 23).

**New Project**

**Empty Activity**

Creates a new empty activity

Name:

Package name:

Save location:

Language:

Minimum SDK:

*i* Your app will run on approximately **95.6%** of devices.  
[Help me choose](#)

☐ Use legacy android.support libraries *?*  
Using legacy android.support libraries will prevent you from using the latest Play Services and Jetpack libraries

Fonte: elaborado pelo autor

*\*Se esta for a primeira vez que utiliza o software, pode ser que na próxima janela apareça um processo de instalação de pacotes adicionais. Apenas aguarde o término do processo, e finalize em "Finish" novamente.*

Após a finalização da criação do projeto, o software leva um tempo para estruturar todos os arquivos.

## Etapa 2: Criando a interface inicial

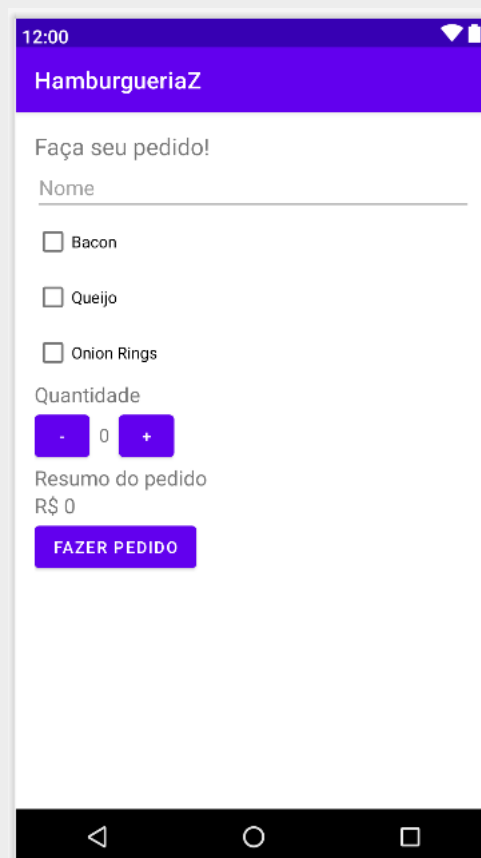
Nesta aula vamos iniciar o desenvolvimento de um aplicativo Android para uma hamburgueria, de modo que os clientes possam realizar os seus pedidos diretamente pelo App sem a necessidade de App de terceiros!

Em um primeiro momento, você precisa elaborar apenas a interface do projeto, preparando os elementos (views) que irão compor o aplicativo para que em uma etapa futura sejam adicionadas as implementações das funcionalidades.

Requisitos da interface:

- Campo para inserir o nome do cliente
- Uma lista do tipo checkbox para o cliente escolher os adicionais em seu hambúrguer.
- Seleção de quantidade de hambúrgueres que será pedido. Faça dois botões, um para adicionar e outro para subtrair a quantidade. Exiba também uma view com a quantidade escolhida.
- Uma exibição do preço total do pedido
- Um botão para enviar o pedido.

Segue um exemplo desta interface, que você poderá usar como guia. Note que as cores, tamanhos ou disposições dos elementos na tela não precisam ser iguais as da imagem. Fica a seu critério o “toque visual” que você quiser dar ao seu aplicativo! O foco aqui é utilizar as views corretas para cada tipo de elemento (texto, botão, inserção de dados, escolhas, ...)

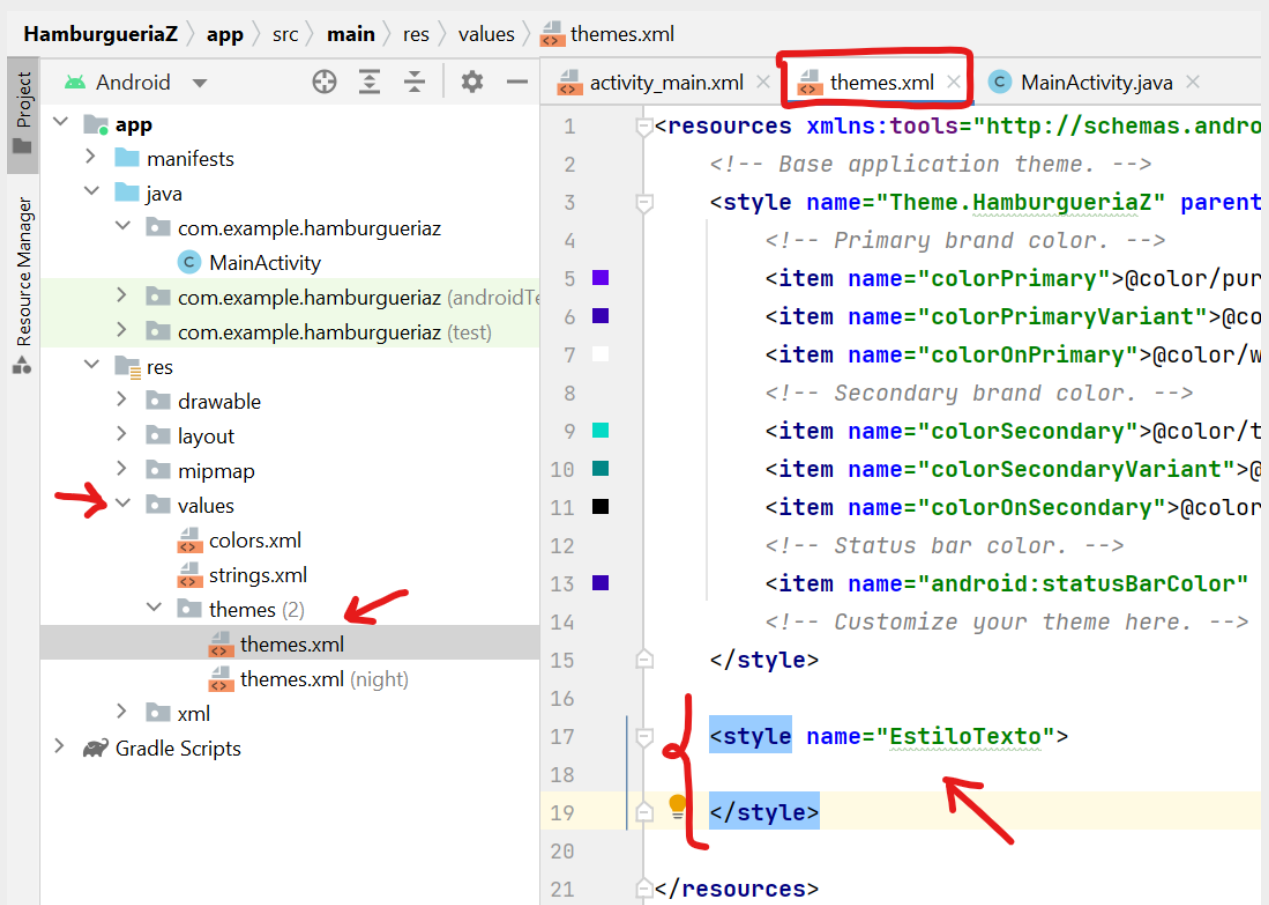


Fonte: elaborado pelo autor

### Etapa 3: Padronizar o estilo dos textos.

Ao invés de definir propriedades de texto em cada view separadamente, podemos criar um estilo pré-definido que pode ser aplicado a várias views de texto para que todas sigam o mesmo formato. E o mais legal é que quando você precisa alterar algo, como o tamanho da fonte por exemplo, você altera somente o estilo e a alteração é replicada para todas as views que possuem este estilo (muito melhor do que mudar cada view separadamente, uma por uma, caso não estivéssemos usando estilos).

E onde criamos isto? No arquivo “themes.xml”, dentro da pasta “res” do diretório do seu app. Dentro do arquivo, crie uma outra tag style (após a que já existe por padrão) com o nome “EstiloTexto”.



Fonte: elaborado pelo autor

Dentro da tag style “EstiloTexto” coloque as propriedades textuais que você deseja aplicar nos textos do seu app.

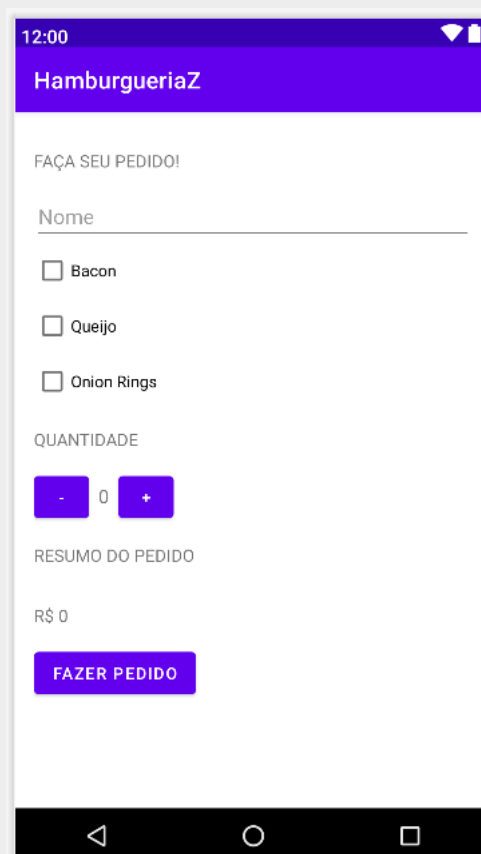
Abaixo seguem as alterações que você deverá fazer:

- Ajuste as propriedades `layout_width` e `layout_height` para `wrap_content`
- Altere a propriedade `android:gravity` para `center_vertical`
- Deixe o texto com todas as letras maiúsculas
- Tamanho do texto: 15sp
- Adicione um padding inferior e um padding superior de 16dp

Após a criação do estilo pré-definido, aplique-o às seguintes Views em seu projeto:

- Na view com o texto: Faça seu pedido
- Na view com o texto: Quantidade
- Na view com o texto: Resumo do pedido
- Na view que terá o valor total do pedido.

Seguindo a ideia de interface da aula anterior, após aplicar as modificações, seu app deve ficar assim:



Fonte: elaborado pelo autor



Muito mais agradável, não é mesmo?

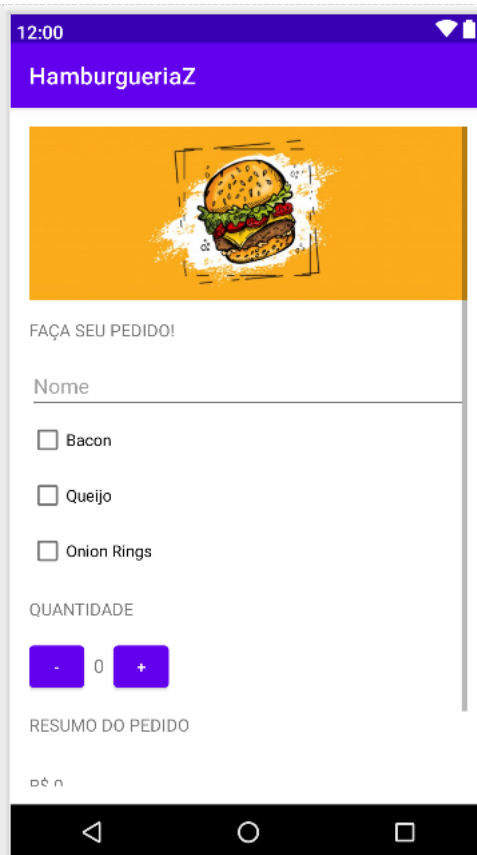
#### **Etapas 4: Adicionando imagens**

A hamburgueria solicitou que você inclua uma imagem contendo a logo do restaurante, conforme a seguir:



Fonte: elaborada pelo autor.

Insira a imagem em seu projeto por meio de uma view específica para imagens, a `<ImageView>`. A imagem deve ficar no topo da tela, no estilo de um banner, e explore os tipos de propriedades que podemos aplicar em imagens para deixá-la conforme a figura a seguir.



Fonte: elaborado pelo autor

Até aqui, trabalhamos com a construção da interface do nosso aplicativo Android HamburgueriaZ. Portanto, com este passo concluídos, vamos implementar as funcionalidades do app, ou seja, a parte lógica da aplicação.

Lembre-se que esta implementação é feita em um arquivo separado, o “MainActivity.java”.

## Procedimento/Atividade Nº 2

### Atividade proposta:

**Desenvolver funcionalidades de uma aplicação Android, e trabalhar com Intents.**

Até aqui, trabalhamos com a construção da interface do nosso aplicativo Android HamburgueriaZ. Portanto, com este passo concluídos, vamos implementar as funcionalidades do app, ou seja, a parte lógica da aplicação.

Lembre-se que esta implementação é feita em um arquivo separado, o “MainActivity.java”.

### Etapa 5:

Implemente uma função “somar” para o botão “+”, e outra função “subtrair” para o botão “-“. E que a cada chamada destas funções, a quantidade deve ser atualizada na view que fica entre os dois botões.

\*Atenção! Cuidado para não permitir a seleção de quantidades negativas!

#### **Etapas 6:**

Implemente uma função para o botão “Enviar pedido”.

- Identifique o nome que o usuário digitou
- Identifique quais adicionais o usuário selecionou
- Com base nos adicionais selecionados e na quantidade de hambúrgueres, faça uma função que some o valor total do pedido (esta função será chamada dentro da função “enviar pedido”). Considere o preço base do hambúrguer por R\$ 20, e que os adicionais bacon, queijo e onion rings custem, respectivamente, R\$ 2, R\$ 2 e R\$ 3.
- Com base nas informações recolhidas, crie uma mensagem que será exibida na view “resumo do pedido” da seguinte maneira:

Nome do cliente

Tem Bacon? Sim/Não

Tem Queijo? Sim/Não

Tem Onion Rings? Sim/Não

Quantidade: \_\_\_\_

Preço final: R\$ \_\_\_\_

*\*Não se esqueça que você deve ajustar as views de botões da interface para a chamada correta das funções implementadas*

#### **Etapas 7:**

Vamos finalizar o aplicativo que estamos desenvolvendo ao longo destas aulas adicionando uma função que fará o pedido ser enviado por e-mail. Usaremos uma estrutura chamada “Intents” para este fim!

Pesquise sobre o uso de Intents em aplicações Android e como utilizá-los para chamar aplicativos de e-mail de terceiros.

- Crie um Intent do tipo ACTION\_SENDTO, relacionado ao envio de e-mails. O Intent deverá ser criado dentro da função “enviarPedido”.
- Deixe o assunto (subject) do e-mail com a seguinte mensagem padrão: "Pedido de (nome do cliente)";
- Deixe no corpo do email a mensagem contendo o resumo do pedido. Aquela que você elaborou na aula anterior:

Nome do cliente

Tem Bacon? Sim/Não

Tem Queijo? Sim/Não

Tem Onion Rings? Sim/Não

Quantidade: \_\_\_\_

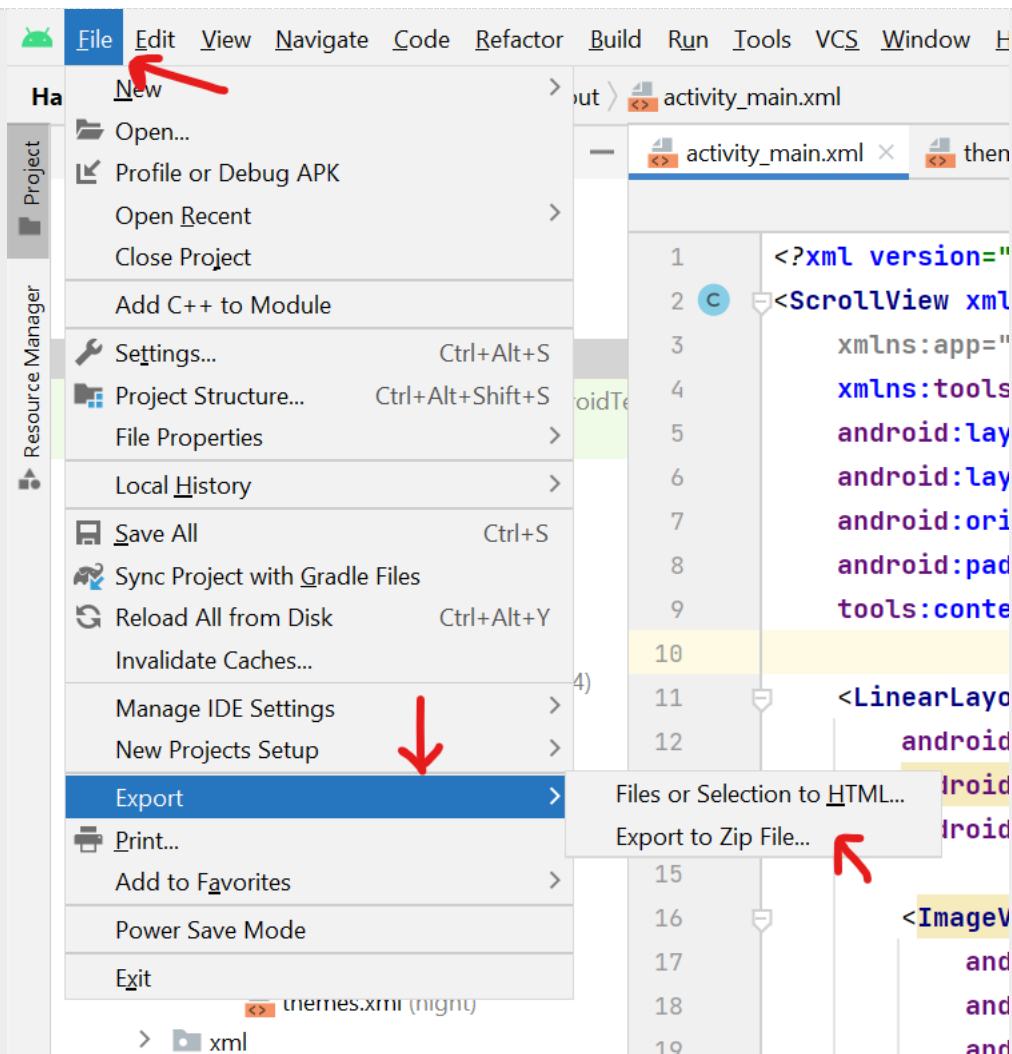
Preço final: R\$ \_\_\_\_

Resumindo o resultado esperado: Ao clicar no botão “Enviar Pedido”, aplicação deverá chamar o aplicativo padrão de e-mail do smartphone (Gmail, por exemplo), que irá abrir já na tela com uma nova mensagem, com o assunto e corpo do e-mail já preenchidos conforme especificado no Intent.

**Pronto! Agora seu aplicativo Android está finalizado!! Parabéns!!**

**Mas falta um último passo: exportar o seu projeto!**

Vá em “File > Export > Export to Zip File...”



Fonte: elaborado pelo autor

Na janela seguinte, escolha o local que deseja salvar o arquivo do projeto e pronto! Bem simples!

#### Checklist:

- Acessar o Android Studio
- Criar um novo projeto
- Criar a interface do app HamburgueriaZ
- Realizar ajustes e incrementos na interface do app
- Construir as funções “somar” e “subtrair” para viabilizar o seletor de quantidade
- Contruir a função “enviarPedido”
- Criar o objeto Intent que irá realizar a chamada do aplicativo de e-mail padrão do smartphone

- Exportar o projeto para um arquivo .zip

## RESULTADOS

### Resultados da aula prática:

É esperado que o aluno ao término da prática o aluno saiba desenvolver uma aplicação simples para a plataforma Android com o Android Studio, incluindo o desenvolvimento da interface e da parte lógica da aplicação.