

Aqui está o **Passo 5 detalhado**, onde implementaremos as funcionalidades de **carrinho** e **finalização de pedido** no aplicativo **Tech Burger**. Vamos criar uma nova activity chamada **CartActivity** para gerenciar o carrinho de compras e adicionar a lógica para os botões **Ver Carrinho** e **Finalizar Pedido** no `MainActivity.kt`.

5. Implementar as Funcionalidades de Carrinho e Finalização de Pedido

Passo 5.1: Criar a Activity do Carrinho (CartActivity)

O carrinho exibirá os itens que o usuário adicionou e permitirá a modificação da quantidade de hambúrgueres.

Passo 5.1.1: Criar a Activity

1. No Android Studio, clique com o botão direito na pasta `java/com.example.techburger`.
2. Selecione **New > Activity > Empty Activity**.
3. Nomeie a activity como `CartActivity` e clique em **Finish**.

Passo 5.1.2: Definir o Layout do Carrinho (`activity_cart.xml`)

1. Abra o arquivo `res/layout/activity_cart.xml`.
2. Defina o layout que exibirá a lista de hambúrgueres adicionados ao carrinho, o total do pedido e os botões para **finalizar** ou **limpar** o carrinho:

xml

 Copiar código

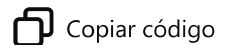
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" android:layout_width="match_parent"
android:layout_height="match_parent" android:orientation="vertical" android:padding="16dp"> <!-- RecyclerView para exibir os
itens no carrinho --> <androidx.recyclerview.widget.RecyclerView android:id="@+id/cartRecyclerView"
android:layout_width="match_parent" android:layout_height="0dp" android:layout_weight="1" /> <!-- Exibir o total do pedido -->
<TextView android:id="@+id/totalTextView" android:layout_width="wrap_content" android:layout_height="wrap_content"
android:text="Total: R$ 0.00" android:textSize="18sp" android:layout_marginTop="16dp" /> <!-- Botão para finalizar o pedido -->
<Button android:id="@+id/checkoutButton" android:layout_width="wrap_content" android:layout_height="wrap_content"
android:text="Finalizar Pedido" android:layout_marginTop="8dp" /> <!-- Botão para limpar o carrinho --> <Button
```

```
android:id="@+id/clearCartButton" android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="Limpar Carrinho" android:layout_marginTop="8dp" /> </LinearLayout>
```

Passo 5.1.3: Implementar a Lógica do Carrinho no CartActivity.kt

1. Abra o arquivo **CartActivity.kt**.
2. Adicione a seguinte lógica para exibir os itens do carrinho, calcular o total e lidar com os botões de **finalização** e **limpeza** do carrinho:

kotlin



```
package com.example.techburguer import android.os.Bundle import android.widget.Button import android.widget.TextView import android.widget.Toast import androidx.appcompat.app.AppCompatActivity import androidx.recyclerview.widget.LinearLayoutManager import androidx.recyclerview.widget.RecyclerView class CartActivity : AppCompatActivity() { private lateinit var cartRecyclerView: RecyclerView private lateinit var totalTextView: TextView private lateinit var checkoutButton: Button private lateinit var clearCartButton: Button override fun onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState) setContentView(R.layout.activity_cart) // Inicializar os componentes da interface cartRecyclerView = findViewById(R.id.cartRecyclerView) totalTextView = findViewById(R.id.totalTextView) checkoutButton = findViewById(R.id.checkoutButton) clearCartButton = findViewById(R.id.clearCartButton) // Configurar RecyclerView cartRecyclerView.layoutManager = LinearLayoutManager(this) val cartAdapter = CartAdapter(CartManager.getOrders()) cartRecyclerView.adapter = cartAdapter // Exibir o total do carrinho updateTotal() // Finalizar pedido checkoutButton.setOnClickListener { val total = CartManager.calculateCartTotal() if (total > 0) { Toast.makeText(this, "Pedido finalizado! Total: R$ %.2f".format(total), Toast.LENGTH_SHORT).show() CartManager.clearCart() updateTotal() cartAdapter.notifyDataSetChanged() // Atualizar o RecyclerView após limpar o carrinho } else { Toast.makeText(this, "Carrinho está vazio!", Toast.LENGTH_SHORT).show() } } // Limpar carrinho clearCartButton.setOnClickListener { CartManager.clearCart() Toast.makeText(this, "Carrinho limpo!", Toast.LENGTH_SHORT).show() updateTotal() cartAdapter.notifyDataSetChanged() // Atualizar o RecyclerView após limpar o carrinho } } // Atualizar o total do carrinho private fun updateTotal() { val total = CartManager.calculateCartTotal() totalTextView.text = "Total: R$ %.2f".format(total) } }
```

Explicação:

- **cartRecyclerView**: Exibe os itens adicionados ao carrinho.
- **updateTotal()**: Atualiza o total do carrinho exibido na tela.

- **checkoutButton:** Quando o botão "Finalizar Pedido" é clicado, o carrinho é limpo e o total é exibido.
- **clearCartButton:** Limpa o carrinho sem finalizar o pedido.

Passo 5.2: Criar o `CartAdapter` para o `RecyclerView` do Carrinho

1. Crie um novo arquivo Kotlin chamado `CartAdapter.kt` na pasta `java/com.example.techburguer`.
2. Adicione o seguinte código para configurar o adaptador do `RecyclerView` para o carrinho:

kotlin



```
package com.example.techburguer import android.view.LayoutInflater import android.view.View import android.view.ViewGroup import
android.widget.ImageView import android.widget.TextView import androidx.recyclerview.widget.RecyclerView class
CartAdapter(private val orderList: List<Order>) : RecyclerView.Adapter<CartAdapter.CartViewHolder>() { // ViewHolder para o
RecyclerView class CartViewHolder(view: View) : RecyclerView.ViewHolder(view) { val burgerImageView: ImageView =
view.findViewById(R.id.burgerImageView) val burgerNameTextView: TextView = view.findViewById(R.id.burgerNameTextView) val
burgerQuantityTextView: TextView = view.findViewById(R.id.burgerQuantityTextView) val burgerPriceTextView: TextView =
view.findViewById(R.id.burgerPriceTextView) } // Inflar o layout de cada item no RecyclerView override fun
onCreateViewHolder(parent: ViewGroup, viewType: Int): CartViewHolder { val view =
LayoutInflater.from(parent.context).inflate(R.layout.cart_item, parent, false) return CartViewHolder(view) } // Ligar os dados do
pedido ao ViewHolder override fun onBindViewHolder(holder: CartViewHolder, position: Int) { val order = orderList[position] val
burger = order.burger holder.burgerNameTextView.text = burger.name holder.burgerQuantityTextView.text = "Quantidade:
${order.quantity}" holder.burgerPriceTextView.text = "R$ %.2f".format(burger.price * order.quantity)
holder.burgerImageView.setImageResource(burger.imageResId) } // Retornar o número de itens no carrinho override fun
getItemCount(): Int { return orderList.size } }
```

Explicação:


- **CartAdapter:** Adapta os itens do carrinho para serem exibidos no `RecyclerView`.
- **onBindViewHolder:** Define o nome, a quantidade, o preço e a imagem de cada item do carrinho.

Passo 5.2.1: Definir o Layout de Item do Carrinho (`cart_item.xml`)

1. Crie um novo arquivo de layout em `res/layout` chamado `cart_item.xml`.

2. Adicione o seguinte código para definir o layout de cada item do carrinho:

xml

 Copiar código

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" android:layout_width="match_parent"
android:layout_height="wrap_content" android:orientation="horizontal" android:padding="16dp"> <!-- Imagem do hambúrguer -->
<ImageView android:id="@+id/burgerImageView" android:layout_width="80dp" android:layout_height="80dp"
android:src="@drawable/ic_burger" android:contentDescription="Imagem do Hambúrguer" /> <!-- Layout para texto (nome, quantidade,
preço) --> <LinearLayout android:layout_width="0dp" android:layout_height="wrap_content" android:layout_weight="1"
android:orientation="vertical" android:layout_marginStart="16dp"> <!-- Nome do hambúrguer --> <TextView
android:id="@+id/burgerNameTextView" android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="Nome
do Burguer" android:textSize="18sp" android:textStyle="bold" /> <!-- Quantidade --> <TextView
android:id="@+id/burgerQuantityTextView" android:layout_width="wrap_content" android:layout_height="wrap_content"
android:text="Quantidade: 1" android:textSize="14sp" android:textColor="#757575" android:layout_marginTop="4dp" /> <!-- Preço
total --> <TextView android:id="@+id/burgerPriceTextView" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:text="R$ 00,00" android:textSize="16sp" android:textColor="#FF9800"
android:layout_marginTop="8dp" /> </LinearLayout> </LinearLayout>
```

Passo 5.3: Integrar o Carrinho no MainActivity.kt

Agora vamos modificar o **MainActivity.kt** para adicionar a funcionalidade de abrir o **CartActivity** quando o botão "Ver Carrinho" for clicado.

1. Abra o arquivo **MainActivity.kt**.
2. Adicione a seguinte lógica ao método `onCreate` para abrir a **CartActivity**:

kotlin

 Copiar código

```
viewCartButton.setOnClickListener { val intent = Intent(this, CartActivity::class.java) startActivity(intent) }
```

Agora, quando o botão "Ver Carrinho" for clicado, o usuário será redirecionado para a tela do carrinho.

Passo 5.4: Testar o Carrinho e a Finalização de Pedido

1. Execute o aplicativo no emulador ou em um dispositivo real.
 2. Adicione alguns hambúrgueres ao carrinho e clique no botão "Ver Carrinho" para ver os itens adicionados.
 3. Teste o botão "Finalizar Pedido" para garantir que o pedido seja finalizado e o carrinho seja limpo.
 4. Teste o botão "Limpar Carrinho" para garantir que os itens sejam removidos.
-

Com isso, o **Passo 5** está completo. Agora, você tem um carrinho funcional que permite visualizar os itens adicionados, limpar o carrinho e finalizar o pedido no aplicativo **Tech Burger**.