

Vamos ao **Passo 4**, que é a **implementação do backend utilizando Kotlin**. Este passo detalha como criar a lógica por trás do aplicativo, conectar os layouts XML ao Kotlin e implementar as funcionalidades de exibir produtos, adicionar ao carrinho e navegar entre as telas.

Passo 4: Implementação do Backend com Kotlin

4.1 Criando a Classe `AcaiProduct`


Primeiro, criaremos uma classe de dados (`data class`) chamada `AcaiProduct` para representar os produtos que serão exibidos na loja.

1. **Crie a classe** `AcaiProduct.kt` :

- No painel esquerdo, clique com o botão direito na pasta `java > com.example.acaiStore` , e crie um novo arquivo Kotlin chamado `AcaiProduct.kt` .

2. **Código da Classe** `AcaiProduct.kt` :

kotlin

 Copiar código

```
package com.example.acaiStore data class AcaiProduct( val id: Int, val name: String, val description: String, val price: Double, val imageUrl: String )
```

Essa classe contém as informações básicas de um produto, como `id` , `name` , `description` , `price` e `imageUrl` . Essas informações serão usadas para preencher os layouts da interface de usuário.

4.2 Criando o Adapter para o RecyclerView


O `RecyclerView.Adapter` é usado para adaptar a lista de produtos e exibi-los na interface. Vamos criar uma classe chamada `AcaiAdapter` que será responsável por isso.

1. **Crie a classe** `AcaiAdapter.kt` :

- No painel esquerdo, clique com o botão direito na pasta `java > com.example.acaiStore` , e crie um novo arquivo Kotlin chamado `AcaiAdapter.kt` .

2. **Código para o** `AcaiAdapter.kt` :

kotlin

 Copiar código

```
package com.example.acaiStore import android.view.LayoutInflater import android.view.View import android.view.ViewGroup import android.widget.Button import android.widget.ImageView import android.widget.TextView import androidx.recyclerview.widget.RecyclerView import com.bumptech.glide.Glide class AcaiAdapter( private val products: List<AcaiProduct>, private val onAddToCartClick: (AcaiProduct) -> Unit, private val onItemClick: (AcaiProduct) -> Unit ) : RecyclerView.Adapter<AcaiAdapter.AcaiViewHolder>() { override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): AcaiViewHolder { val view = LayoutInflater.from(parent.context) .inflate(R.layout.item_acai, parent, false) return AcaiViewHolder(view) } override fun onBindViewHolder(holder: AcaiViewHolder, position: Int) { val product = products[position] holder.bind(product) } override fun
```

```

getItemCount(): Int = products.size inner class AcaiViewHolder(itemView: View) :
RecyclerView.ViewHolder(itemView) { private val productImage: ImageView =
itemView.findViewById(R.id.productImage) private val productName: TextView =
itemView.findViewById(R.id.productName) private val productDescription: TextView =
itemView.findViewById(R.id.productDescription) private val productPrice: TextView =
itemView.findViewById(R.id.productPrice) private val addToCartButton: Button =
itemView.findViewById(R.id.addToCartButton) fun bind(product: AcaiProduct) {
productName.text = product.name productDescription.text = product.description
productPrice.text = "R$ ${product.price}" // Carregando a imagem usando Glide
Glide.with(itemView.context) .load(product.imageUrl) .into(productImage) // Ação ao
clicar no botão "Adicionar" addToCartButton.setOnClickListener {
onAddToCartClick(product) } // Ação ao clicar no item da lista
itemView.setOnClickListener { onItemClick(product) } } } }

```

4.3 Ligando o RecyclerView com o Adapter no MainActivity.kt

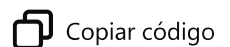
Agora vamos conectar o RecyclerView ao AcaiAdapter no MainActivity.kt e definir a funcionalidade de exibir a lista de produtos e interagir com o carrinho de compras.

1. Abra o arquivo MainActivity.kt :

- O arquivo já deve existir, pois foi criado automaticamente quando o projeto foi gerado.

2. Código para o MainActivity.kt :

kotlin



```

package com.example.acaiStore import android.content.Intent import android.os.Bundle
import android.widget.Toast import androidx.appcompat.app.AppCompatActivity import
androidx.recyclerview.widget.LinearLayoutManager import
androidx.recyclerview.widget.RecyclerView import
com.google.android.material.floatingactionbutton.FloatingActionButton class
MainActivity : AppCompatActivity() { private lateinit var recyclerView: RecyclerView
private lateinit var acaiAdapter: AcaiAdapter private lateinit var cartFab:
FloatingActionButton // Lista de produtos fictícios (substitua com dados reais ou de
uma API) private val products = listOf( AcaiProduct(1, "Açaí Tradicional", "Açaí com
banana e granola", 12.99, "https://example.com/acai1.jpg"), AcaiProduct(2, "Açaí com
Morango", "Açaí com morango e leite condensado", 15.99,
"https://example.com/acai2.jpg"), AcaiProduct(3, "Açaí com Leite em Pó", "Açaí com
leite em pó e mel", 14.99, "https://example.com/acai3.jpg") ) override fun
onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState)
setContentView(R.layout.activity_main) recyclerView = findViewById(R.id.recyclerView)
cartFab = findViewById(R.id.fab_cart) // Configurando o RecyclerView com o Adapter
acaiAdapter = AcaiAdapter(products, onAddToCartClick = { product ->
CartManager.addToCart(product) Toast.makeText(this, "${product.name} adicionado ao
carrinho!", Toast.LENGTH_SHORT).show() }, onItemClick = { product -> // Navegar para a
tela de detalhes do produto val intent = Intent(this,
ProductDetailsActivity::class.java) intent.putExtra("PRODUCT_ID", product.id)
startActivity(intent) }) recyclerView.layoutManager = LinearLayoutManager(this)
recyclerView.adapter = acaiAdapter // Ação ao clicar no botão flutuante para ver o

```

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

4.4 Implementando o Gerenciador de Carrinho (CartManager)

Vamos criar uma classe `CartManager` que vai gerenciar os itens no carrinho de compras. Essa classe armazenará os produtos adicionados ao carrinho e permitirá o acesso à lista de produtos no carrinho.

1. Crie a classe `CartManager.kt` :

- No painel esquerdo, clique com o botão direito na pasta `java > com.example.acaiStore` , e crie um novo arquivo Kotlin chamado `CartManager.kt` .

2. Código para o `CartManager.kt` :

kotlin



```
package com.example.acaiStore object CartManager { private val cartItems =
mutableListOf<AcaiProduct>() // Adicionar um produto ao carrinho fun
addToCart(product: AcaiProduct) { cartItems.add(product) } // Retornar os itens do
carrinho fun getCartItems(): List<AcaiProduct> = cartItems // Limpar o carrinho fun
clearCart() { cartItems.clear() } }
```

4.5 Tela de Detalhes do Produto (ProductDetailsActivity.kt)

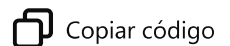
Agora vamos criar a tela de detalhes do produto, onde o usuário pode ver mais informações sobre um produto específico.

1. Crie a classe `ProductDetailsActivity.kt` :

- No painel esquerdo, clique com o botão direito na pasta `java > com.example.acaiStore` , e crie um novo arquivo Kotlin chamado `ProductDetailsActivity.kt` .

2. Código para o `ProductDetailsActivity.kt` :

kotlin



```
package com.example.acaiStore import android.os.Bundle import android.widget.Button
import android.widget.ImageView import android.widget.TextView import
androidx.appcompat.app.AppCompatActivity import com.bumptech.glide.Glide class
ProductDetailsActivity : AppCompatActivity() { private lateinit var productImage:
ImageView private lateinit var productName: TextView private lateinit var
productDescription: TextView private lateinit var productPrice: TextView private
lateinit var addToCartButton: Button override fun onCreate(savedInstanceState:
Bundle?) { super.onCreate(savedInstanceState)
setContentView(R.layout.activity_product_details) productImage =
findViewById(R.id.productImageDetails) productName =
findViewById(R.id.productNameDetails) productDescription =
findViewById(R.id.productDescriptionDetails) productPrice =
findViewById(R.id.productPriceDetails) addToCartButton =
```

```

findViewById(R.id.addToCartButtonDetails) val productId =
intent.getIntExtra("PRODUCT_ID", -1) // Simular busca do produto pelo ID (substitua
por chamada de API ou banco de dados) val product =
MainActivity().products.firstOrNull { it.id == productId } product?.let {
productName.text = it.name productDescription.text = it.description productPrice.text
= "R$ ${it.price}" Glide.with(this) .load(it.imageUrl) .into(productImage) // Ação
para adicionar o produto ao carrinho addToCartButton.setOnClickListener {
CartManager.addToCart(it) finish() // Volta à tela principal após adicionar ao
carrinho } } } }


```

4.6 Tela do Carrinho de Compras (CartActivity.kt)

Finalmente, vamos criar a tela do carrinho de compras onde o usuário pode ver todos os itens adicionados e finalizar a compra.

1. **Crie a classe** CartActivity.kt :
 - No painel esquerdo, clique com o botão direito na pasta `java > com.example.acaiStore` , e crie um novo arquivo Kotlin chamado `CartActivity.kt` .
2. **Código para o** CartActivity.kt :

kotlin

 Copiar código

```

package com.example.acaiStore import android.os.Bundle import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity import
androidx.recyclerview.widget.LinearLayoutManager import
androidx.recyclerview.widget.RecyclerView class CartActivity : AppCompatActivity() {
private lateinit var recyclerView: RecyclerView private lateinit var
totalPriceTextView: TextView override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState) setContentView(R.layout.activity_cart) recyclerView
= findViewById(R.id.cartRecyclerView) totalPriceTextView =
findViewById(R.id.totalPrice) val cartItems = CartManager.getCartItems()
recyclerView.layoutManager = LinearLayoutManager(this) recyclerView.adapter =
AcaiAdapter(cartItems, {}, {}) // Atualiza o preço total val totalPrice =
cartItems.sumOf { it.price } totalPriceTextView.text = "Total: R$ $totalPrice" } }

```

Resumo

Agora, com o backend em Kotlin implementado, o aplicativo de loja de venda de açaí está funcional. O usuário pode visualizar a lista de produtos, ver detalhes dos produtos, adicionar itens ao carrinho e visualizar o carrinho de compras. Se precisar de mais detalhes ou ajustes, estou à disposição!