

FIA/P GRADUAÇÃO

Hybrid Mobile App Development

Prof. Andrey Masiero

#01 – Introdução do Desenvolvimento com Kotlin



ANDREY ARAUJO MASIERO

PROFESSOR GRADUAÇÃO E MBA NA FIAP

- MAIS DE 16 ANOS DE EXPERIÊNCIA
- MESTRADO EM ENGENHARIA ELÉTRICA COM ÊNFASE EM IA
- DOUTORADO EM ENGENHARIA ELÉTRICA COM ÊNFASE EM IA
- LECIONO TODOS OS NÍVEIS DE PROGRAMAÇÃO A 10 ANOS

CONTEÚDO: ANDREYMASIERO.COM

SEGUIE NO INSTA: [@ANDREYMASIERO](https://www.instagram.com/ANDREYMASIERO)

LINKEDIN: [ANDREYMASIERO](https://www.linkedin.com/in/ANDREYMASIERO)

EMAIL: PROFANDREY.MASIERO@FIAP.COM.BR

Objetivo da disciplina

- Preparar você para o mercado de trabalho.
- Apresentar as tecnologias mobile utilizadas no mercado.
- Desenvolver aplicações utilizando Kotlin e React Native.
- Realizar integrações entre diferentes dispositivos.

Conteúdo Programático

- Android Studio + Kotlin
- React Native
- Integração com IoT

METODOLOGIA E AVALIAÇÕES

- **Metodologia**

- Aulas “Hands On”;
- Projetos no Github;

- **Avaliações**

- **3 NACs** Prática;
- **Prova Semestral** - Prática;
- **AM** – Critérios de avaliação na área de apostilas, no portal do aluno.

Let's Code

Apresentações feitas, bora codar!

Começando o caminho Jedi



Kotlin

- Linguagem oficial do Android, desde a versão 3.0 do Android Studio.
- Escrever menos código, com maior leitura.
- Minimiza problemas com objetos nulos, tornando-a mais segura.
- Interoperabilidade com Java.

<https://developer.android.com/kotlin/index.html>



Pré-requisitos

- Java 8
- Kotlin
- Android Studio
- Android SDK + Emulador



Suas Finanças

Vamos ver se você investe bem!

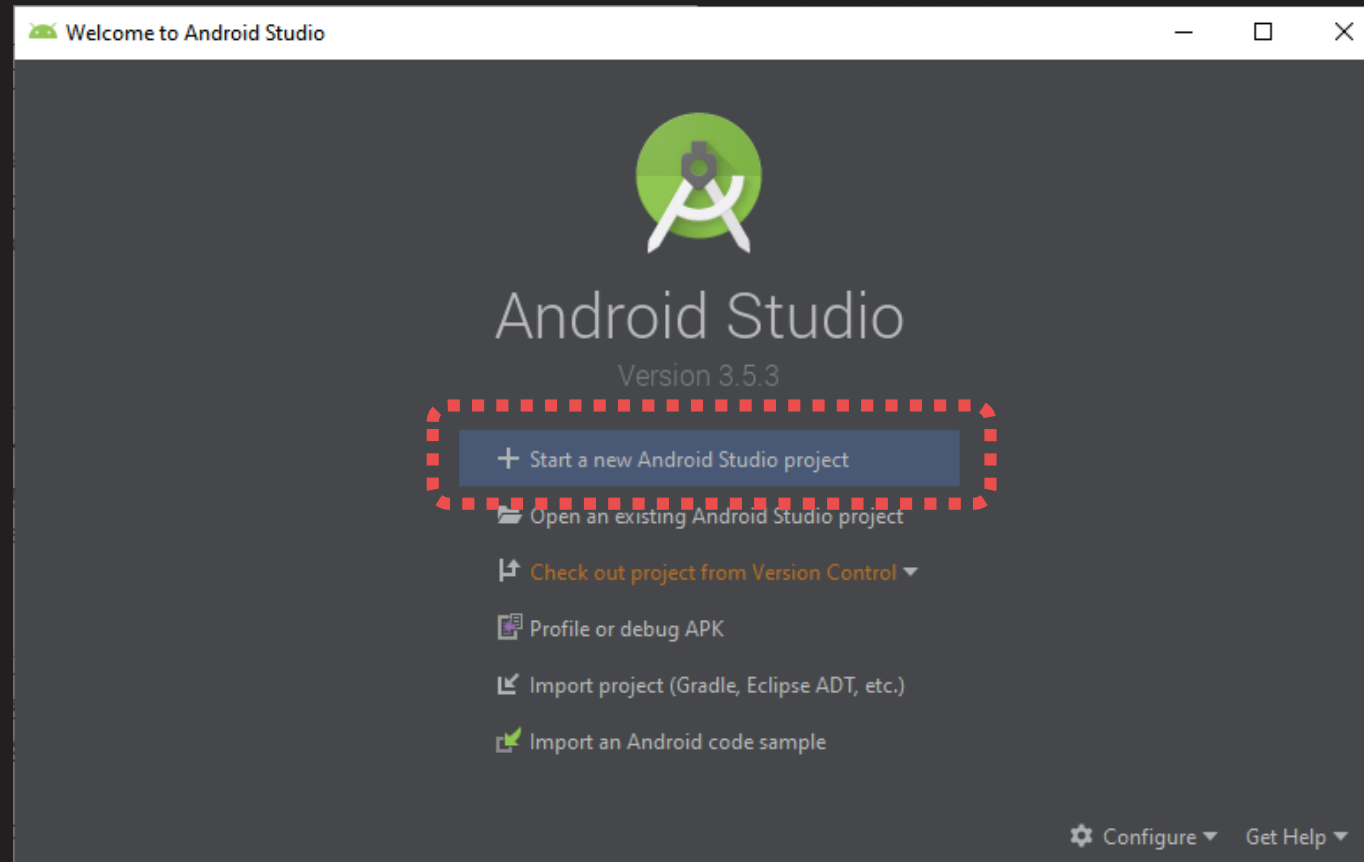
Começando o projeto

- Vamos desenvolver um pequeno controle financeiro.
- A ideia é apresentar o Kotlin e como funciona uma aplicação 100% nativa de um OS Mobile.
- O foco é na linguagem e algumas coisas do Android.



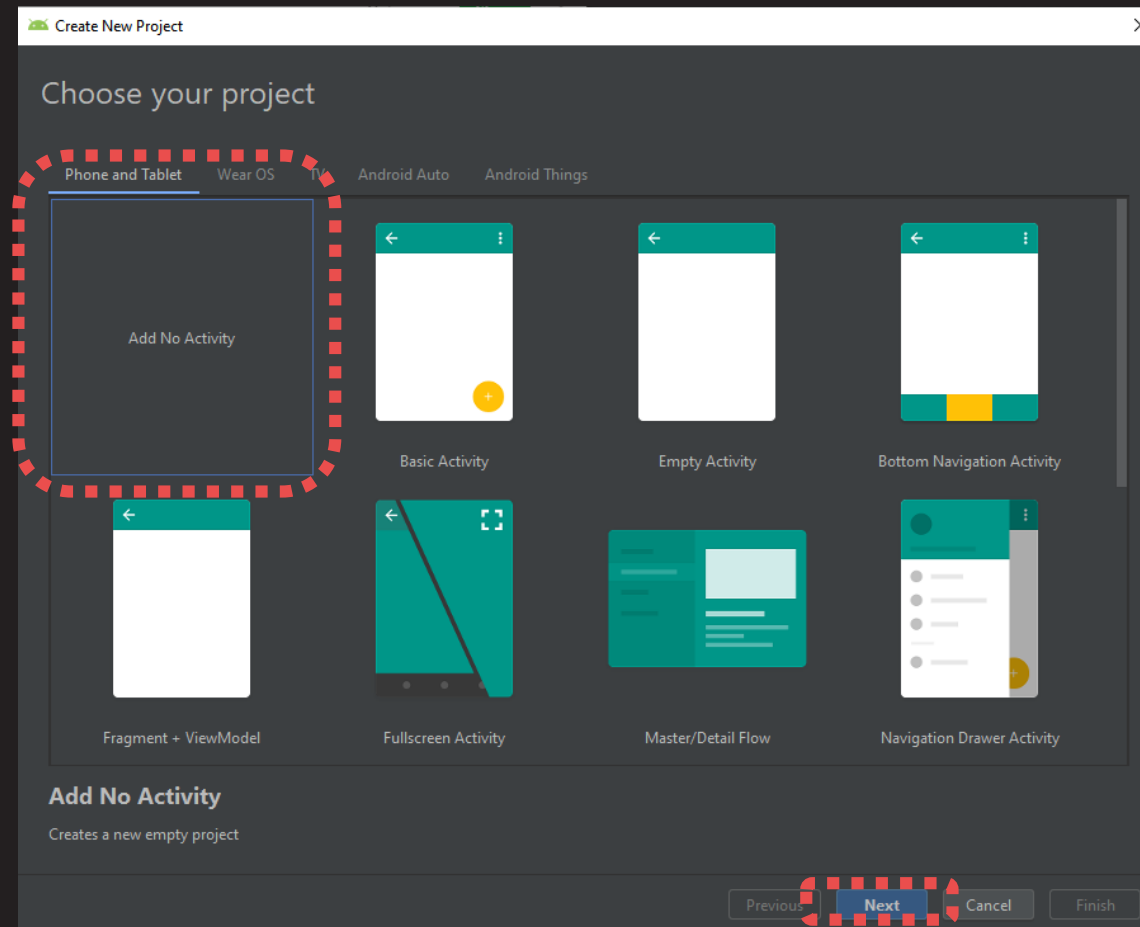
Criando o projeto

- Abra o Android Studio, selecione Start a new AS project



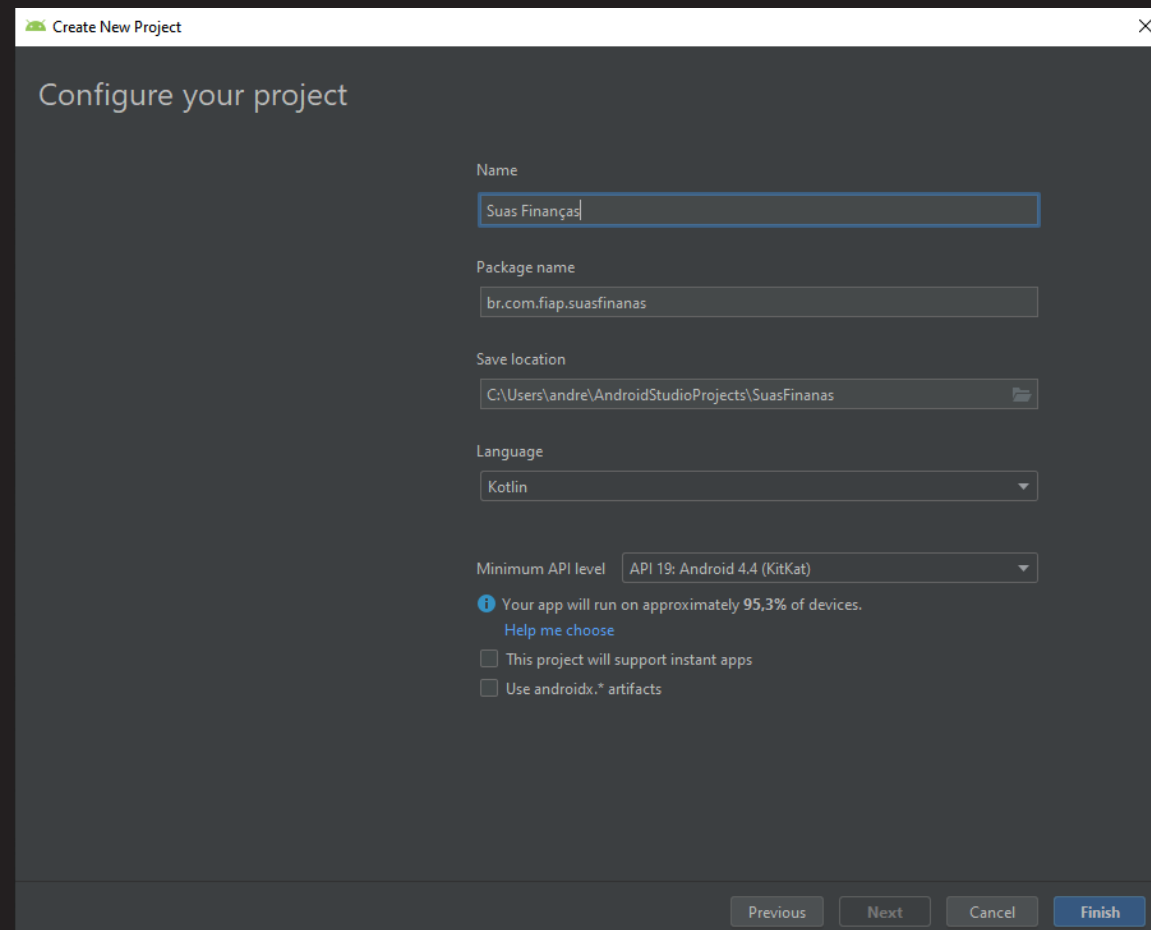
Criando o projeto

- Vamos criar uma tela sem template, selecione Add No Activity e clique Next.



Criando o projeto

- Configure o projeto como apresentado e clique em Finish.



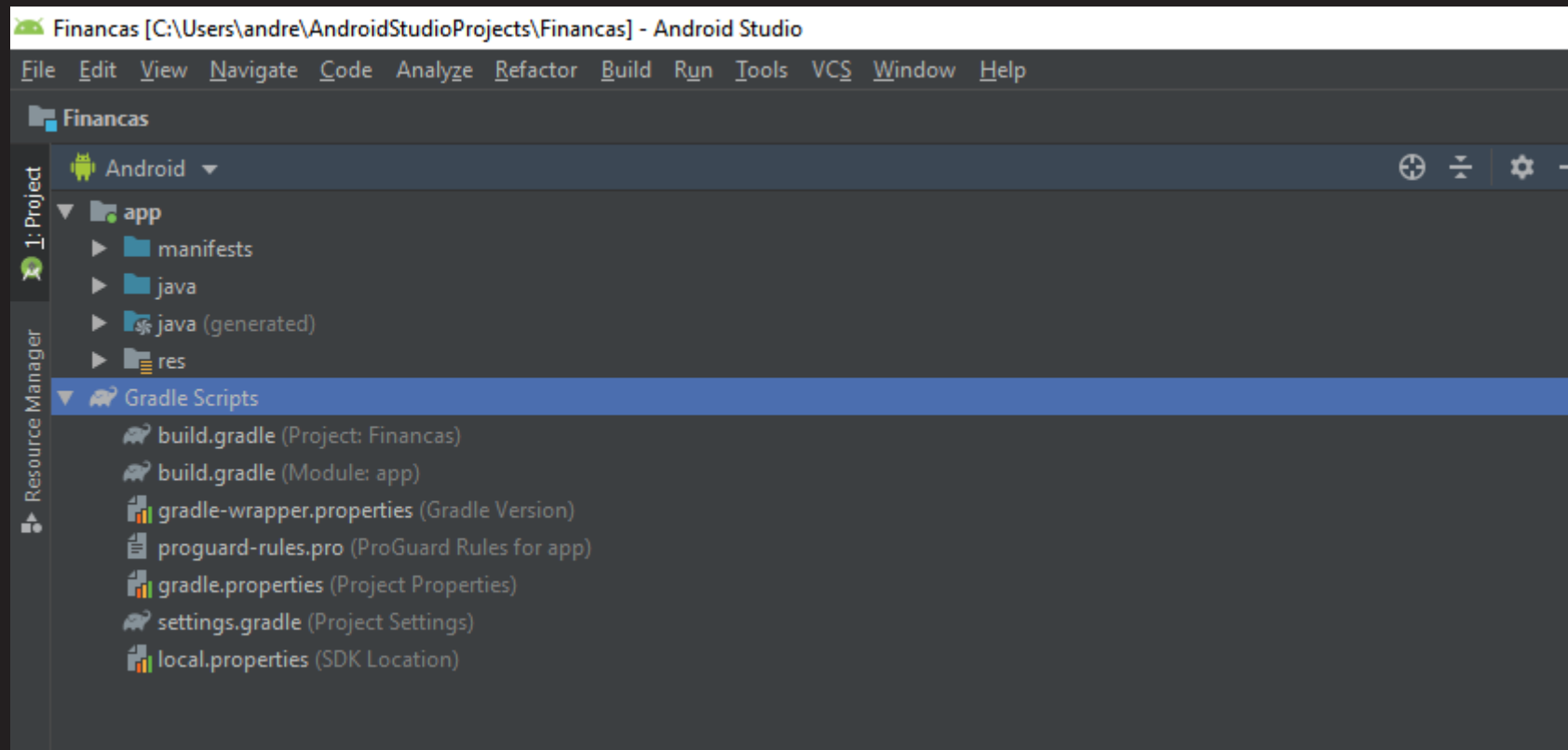
The screenshot shows the 'Create New Project' dialog box in Android Studio. The title bar reads 'Create New Project' with a close button. The main heading is 'Configure your project'. The form contains the following fields and options:

- Name:** A text field containing 'Suas Finanças'.
- Package name:** A text field containing 'br.com.fiap.suasfinanas'.
- Save location:** A text field containing 'C:\Users\andre\AndroidStudioProjects\SuasFinanas' with a folder icon on the right.
- Language:** A dropdown menu with 'Kotlin' selected.
- Minimum API level:** A dropdown menu with 'API 19: Android 4.4 (KitKat)' selected.
- Information:** A blue information icon followed by the text 'Your app will run on approximately 95,3% of devices.' and a link 'Help me choose'.
- Options:** Two unchecked checkboxes: 'This project will support instant apps' and 'Use androidx.* artifacts'.

At the bottom, there are four buttons: 'Previous' (disabled), 'Next' (disabled), 'Cancel', and 'Finish' (highlighted in blue).

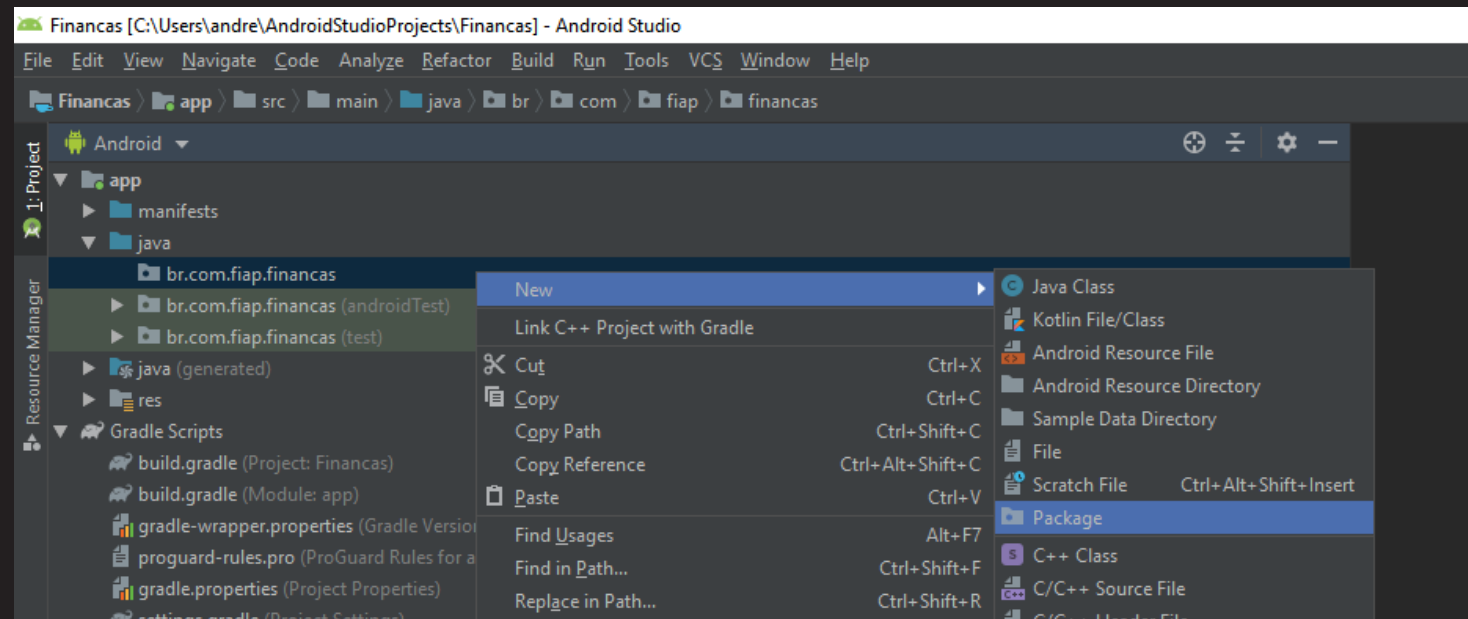
Estrutura do projeto

- Após a criação do projeto, o Android Studio deverá apresentar a seguinte estrutura.



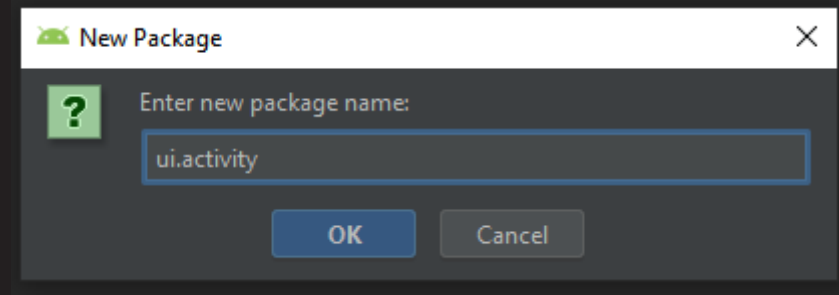
Criando o pacote

- No Android uma tela é composta por um XML que descreve o layout e uma classe Java/Kotlin chamada de Activity.
- Vamos criar um pacote para organizar a activity, com o botão direito no pacote br.com.fiap.projeto -> New -> Package



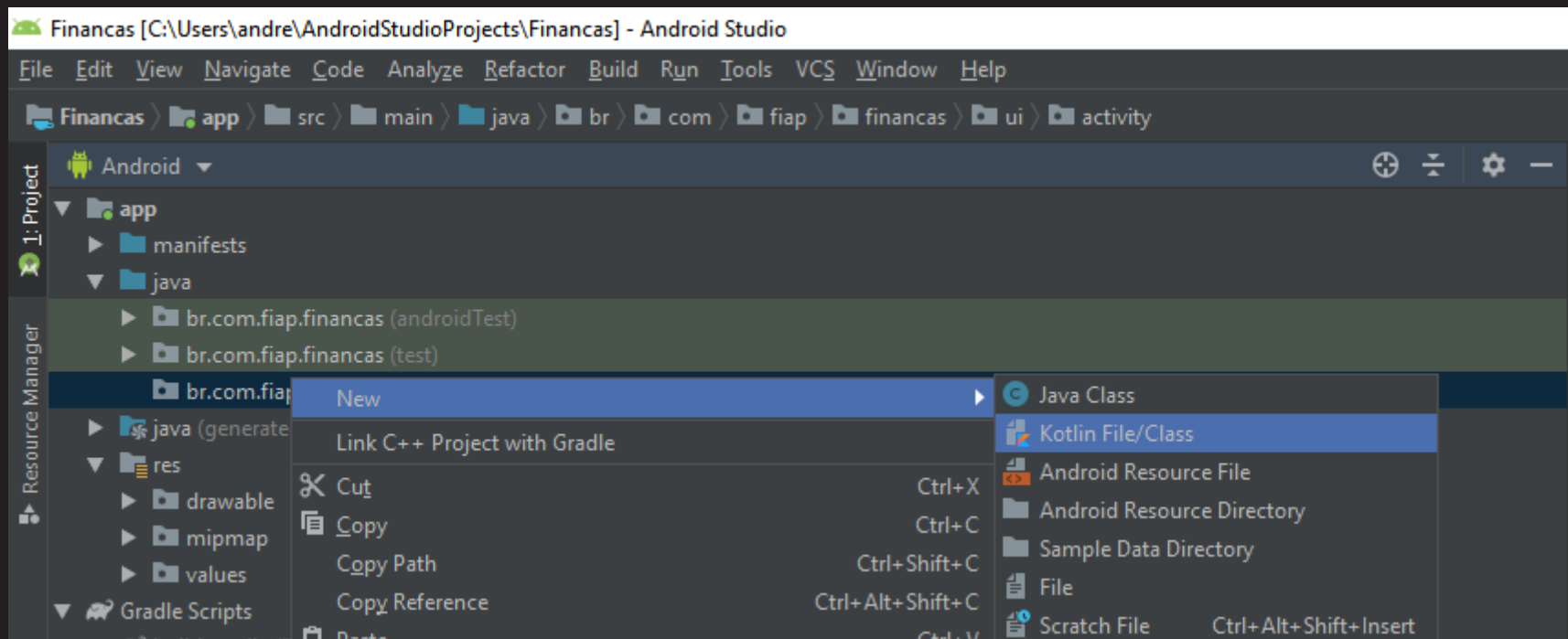
Criando o pacote

- Agora daremos o nome de `ui.activity` para o pacote e clicamos em OK



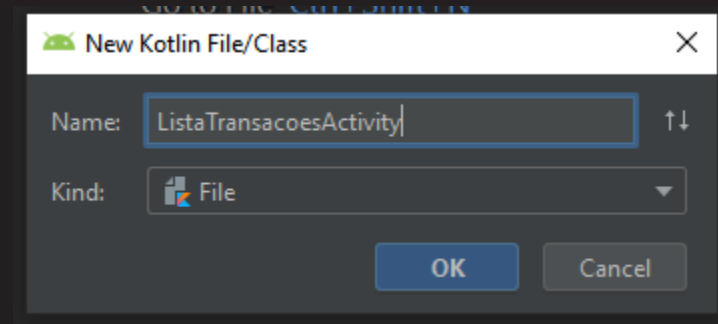
Criando a Activity

- A Activity é uma classe Kotlin. Para criá-la clique com o botão direito no pacote que criamos e faça:



Criando a Activity

- Dê o nome de ListaTransacoesActivity, clique em ok.



Criando a Activity

- O resultado deve ser esse:

```
1 package br.com.fiap.financas.ui.activity
2
3
4 class ListaTransacoesActivity {
5
6 }
```

Criando a Activity

- Vamos fazer com que ela herde uma activity base do Android, AppCompatActivity

```
1 package br.com.fiap.financas.ui.activity
2
3 import android.support.v7.app.AppCompatActivity
4
5
6 class ListatransacoesActivity : AppCompatActivity() {
7
8 }
```

Representa o extends, símbolo para herança no Kotlin

Chamada do construtor da super classe

Criando a Activity

- Vamos adicionar o método onCreate. Ele invoca o nosso layout quando a classe é instanciada.

```
6
7 class ListaTransacoesActivity : AppCompatActivity() {
8
9     override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11     }
12 }
```

Tipo do parâmetro

Igual ao override do Java

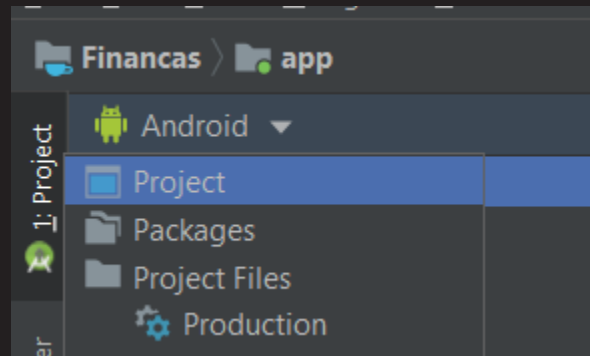
Para o Kotlin tudo é função.

Chama o método da classe Mãe

Informa que pode ser nulo

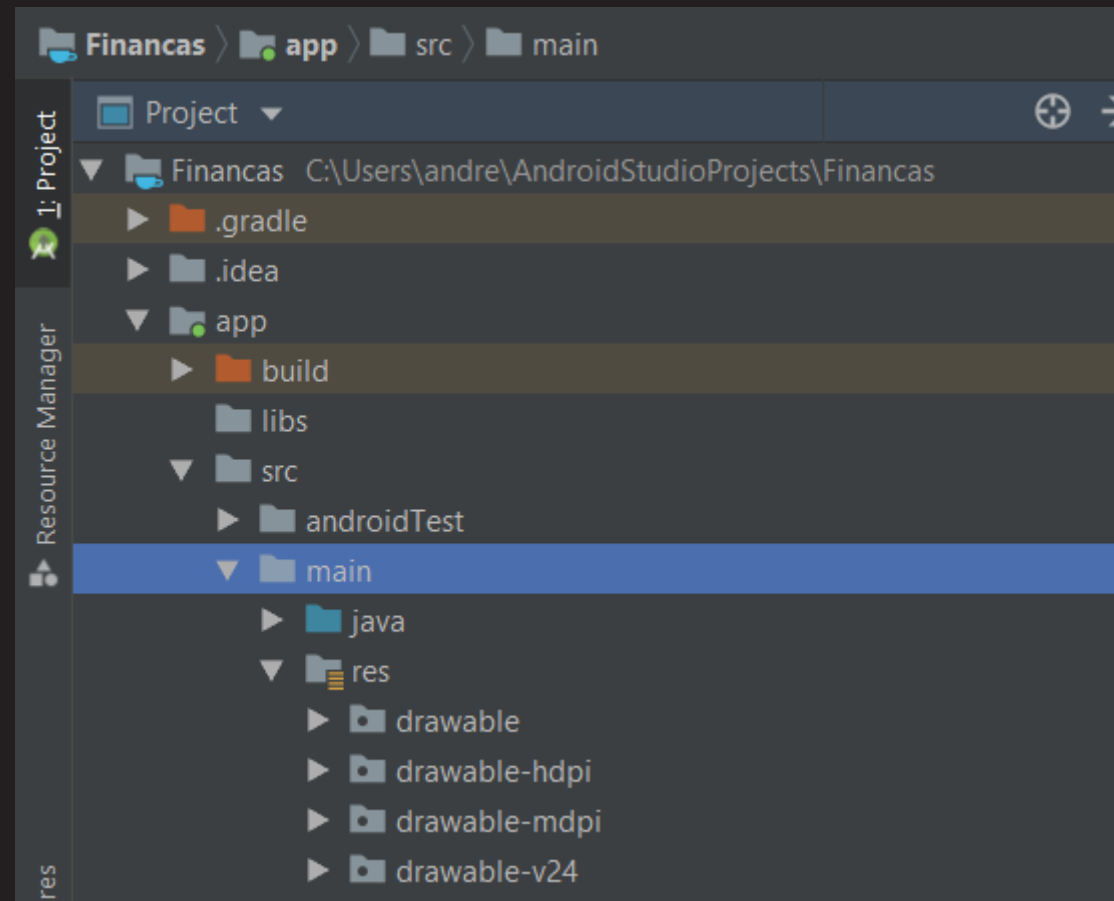
Carregando um layout

- Faça o download do arquivo resources.zip, e o descompacte.
- Troque a vista de Android para Projeto, no package explorer:



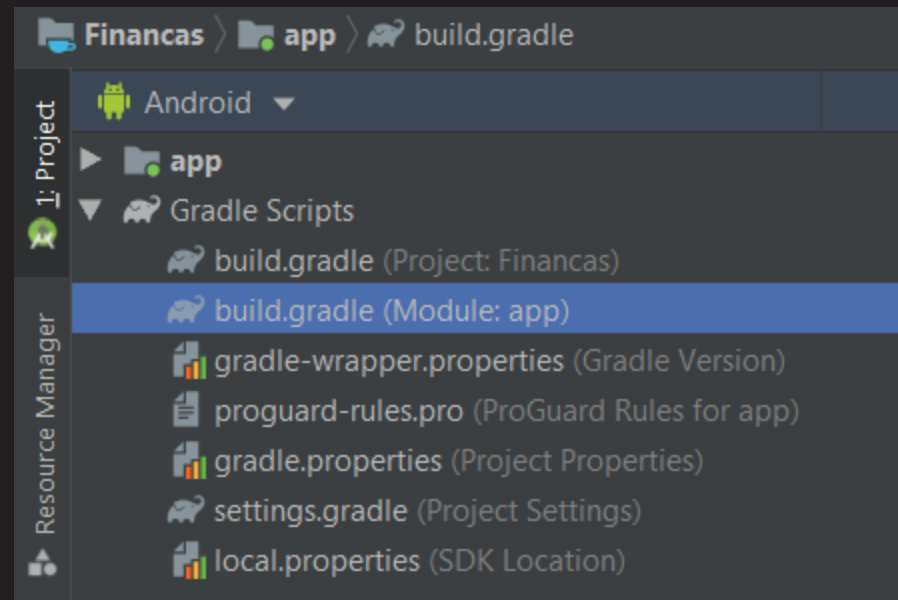
Carregando um layout

- Cole a pasta resources em app -> src -> main:



Configurando o layout

- Nosso layout possui alguns componentes externos. Então vamos adicionar essas dependências do projeto no arquivo build.gradle



Configurando o layout

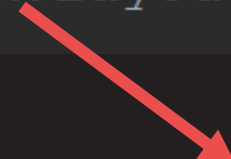
- Copie as seguintes dependências dentro do arquivo:

```
implementation 'com.android.support:appcompat-v7:26.0.1'  
implementation 'com.android.support:design:26.0.1'  
implementation 'com.android.support:cardview-v7:26.0.1'  
implementation 'com.github.clans:fab:1.6.4'
```

Invocando o layout

- Agora que importamos o layout no projeto, vamos vincula-lo com a classe Activity. Faremos isso através do método setContentView:

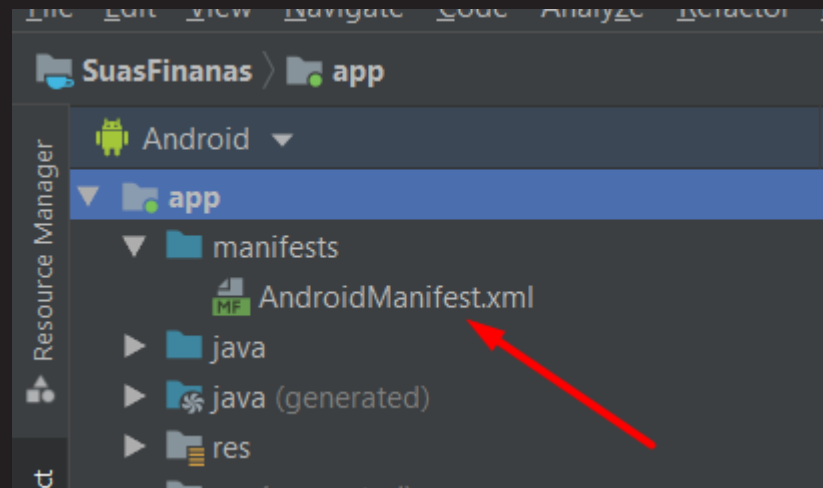
```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_lista_transacoes)
```



O objeto R é uma classe utilizada pelo próprio Android, para acessar a pasta de resources através do código Kotlin.

Invocando o layout

- Precisamos registrar a Activity no arquivo AndroidManifest. Acesse o arquivo dentro da pasta app -> manifests



Invocando o layout

- Altere para o seguinte código:

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Suas Finanças"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme" >
    <activity android:name="br.com.fiap.suasfinancas.ui.activity.ListaTransacoesActivity">
        <intent-filter>
            <category android:name="android.intent.category.LAUNCHER" />
            <action android:name="android.intent.action.MAIN" />
        </intent-filter>
    </activity>
</application>
```

Invocando o layout

Classe que define sua activity

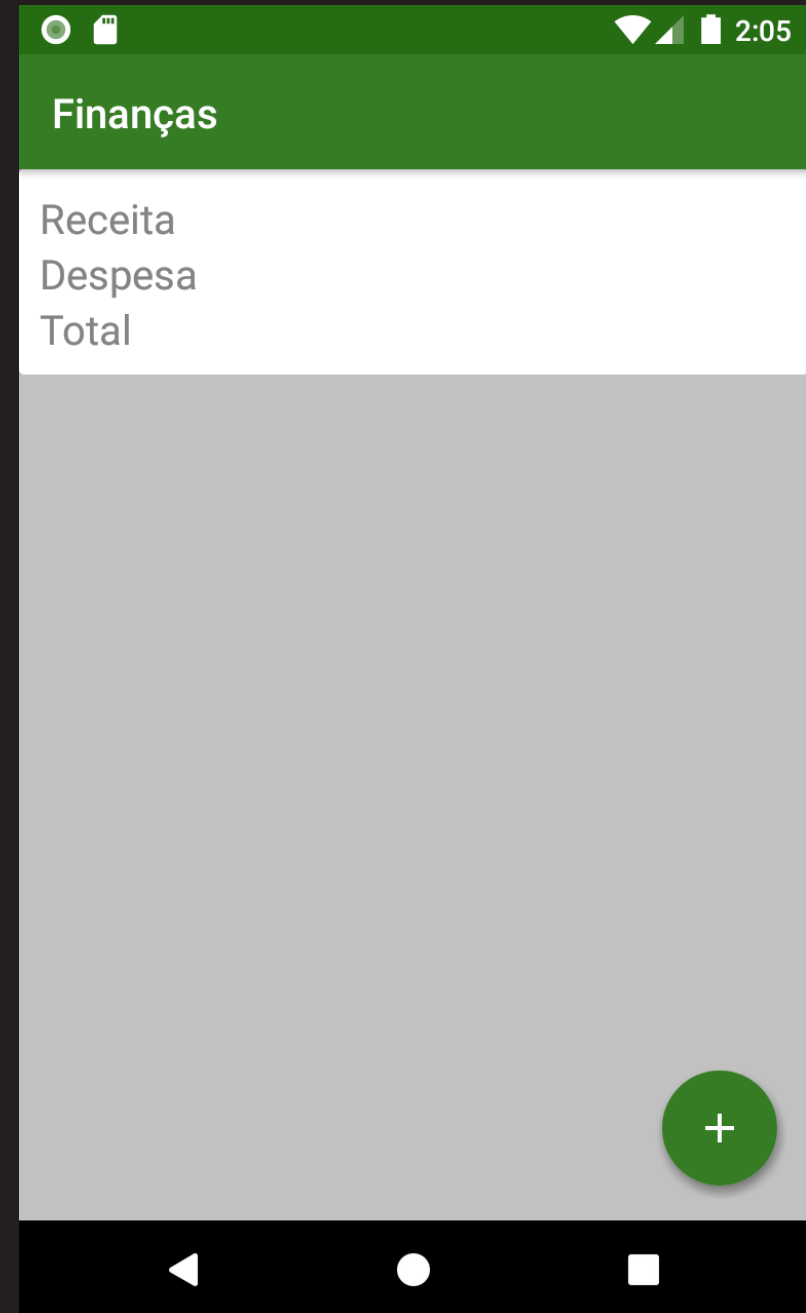
```
<activity android:name="br.com.fiap.suasfinancas.ui.actitivity.ListaTransacoesActivity">
  <intent-filter>
    <category android:name="android.intent.category.LAUNCHER" />
    <action android:name="android.intent.action.MAIN" />
  </intent-filter>
</activity>
```

Qual a ação que o aplicativo deve realizar? Nesse caso, se comportar como a classe principal. A primeira na execução.

Tipo de activity que está sendo vinculada. Launcher é a primeira chamada ao executar o app.

Invocando o layout

- Ao executar o resultado deve ser preenchido da seguinte maneira:



Trabalhando com Listas

Como apresentar seus dados

Criando uma lista

- O primeiro passo para integrarmos uma lista com a view é identificar o elemento de view no layout. Utilizamos o id do elemento com o método `findViewById`.

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_lista_transacoes)  
  
    findViewById<ListView>(R.id.lista_transacoes_listview)  
}
```

Criando uma lista

- Contudo, o plugin kotlin-android-extensions, nos permite simplificar o nosso código para apenas o nome do elemento da view:

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_lista_transacoes)  
  
    lista_transacoes_listview  
}
```

Criando uma lista


- Esse é um conceito de synthetic do Kotlin. Para que funcione o seguinte import deve ser feito em sua classe:

```
import kotlinx.android.synthetic.main.activity_lista_transacoes.*
```

Criando uma lista

- O componente ListView deve receber a lista que será exibida na tela. Vamos criar a lista:

```
val transacoes = listOf("Comida: R$ 20,00",  
    "Crédito: R$ 100,00")
```



val define que a lista é imutável.
Se precisar reinstanciar sua
lista/variável utilize **var** no local.



O método `listOf` retorna uma
lista de objetos

Criando uma lista

- Agora vamos atribuir a lista ao componente. Precisamos de um objeto do padrão adapter. Utilizaremos o ArrayAdapter por ser mais simples, nesse primeiro momento.

```
val arrayAdapter = ArrayAdapter(context: this,  
    android.R.layout.simple_list_item_1,  
    transacoes)
```

Recebe o contexto
que será aplicado, no
caso a própria Activity

Lista com os objetos para
serem exibidos.

O layout que deve seguir, no caso utilizaremos
o do próprio Android para testarmos. Depois
faremos o nosso.

Criando uma lista

- Por fim, vamos vincular o nosso ListView com o adapter que criamos.

```
lista_transacoes_listview.adapter = arrayAdapter
```

Criando uma lista

- Confira o resultado:



Exercícios

Hora do descanso.

Exercícios

- Crie um layout que exiba uma lista de itens do tipo texto. Pode ser qualquer aplicação.
- Vincule um activity para este layout e exiba as informações, desejadas.

Próximos Passos

O que veremos na próxima aula

Na próxima aula...

- Criar o modelo de Transação
- Melhorar o layout do adapter
- Mais padrões de projeto



Copyright © 2020
Prof. Andrey Masiero

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).

Do or do not. There is no try – Mestre Yoda