



Reskilling 4Employment Software Developer

Acesso móvel a sistemas de informação

Bruno Santos

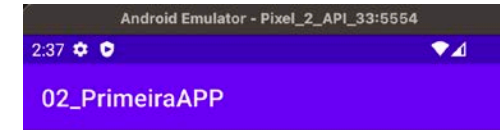
bruno.santos.mcv@msft.cesae.pt

Tópicos

- Criação de uma APP
- Ficheiro .kt e .xml
- Organização de Constraint Layout
- Referenciação de elementos visuais na lógica
- Operação de clique
- Conversões de tipos de dados

Primeira APP

- Vamos criar uma aplicação que peça ao utilizador o seu nome e apresente uma mensagem de boas-vindas seguida do nome do utilizador



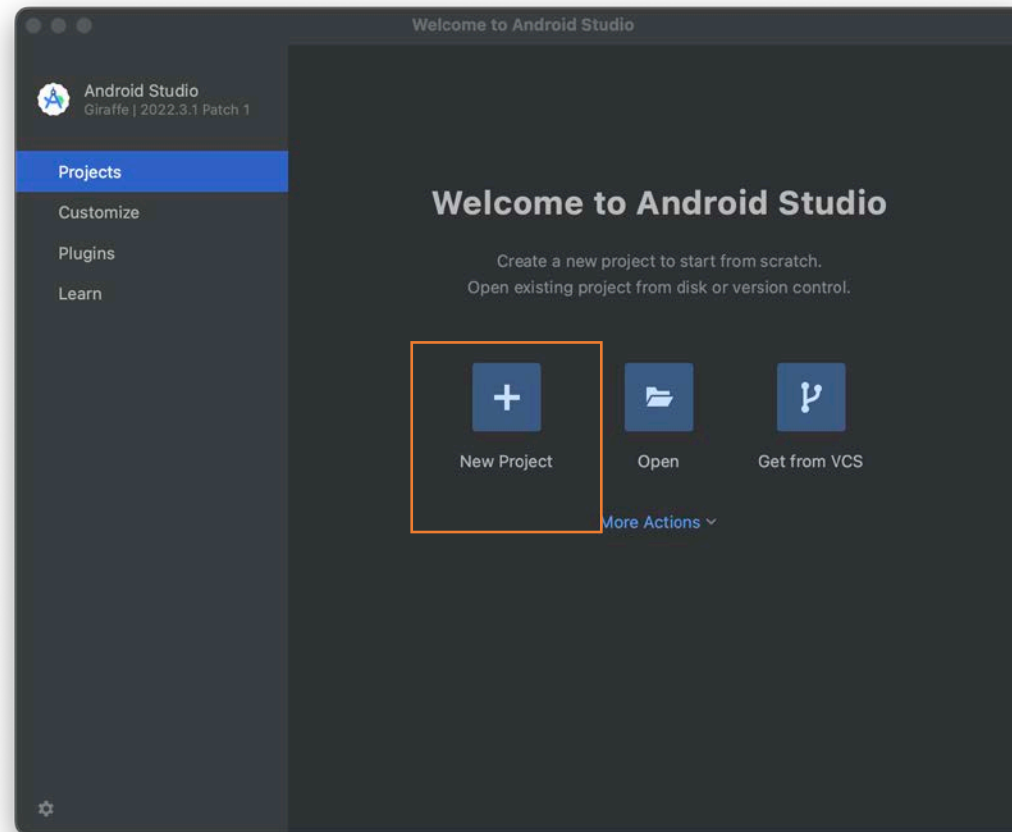
Insira o seu nome

DIZER OLÁ

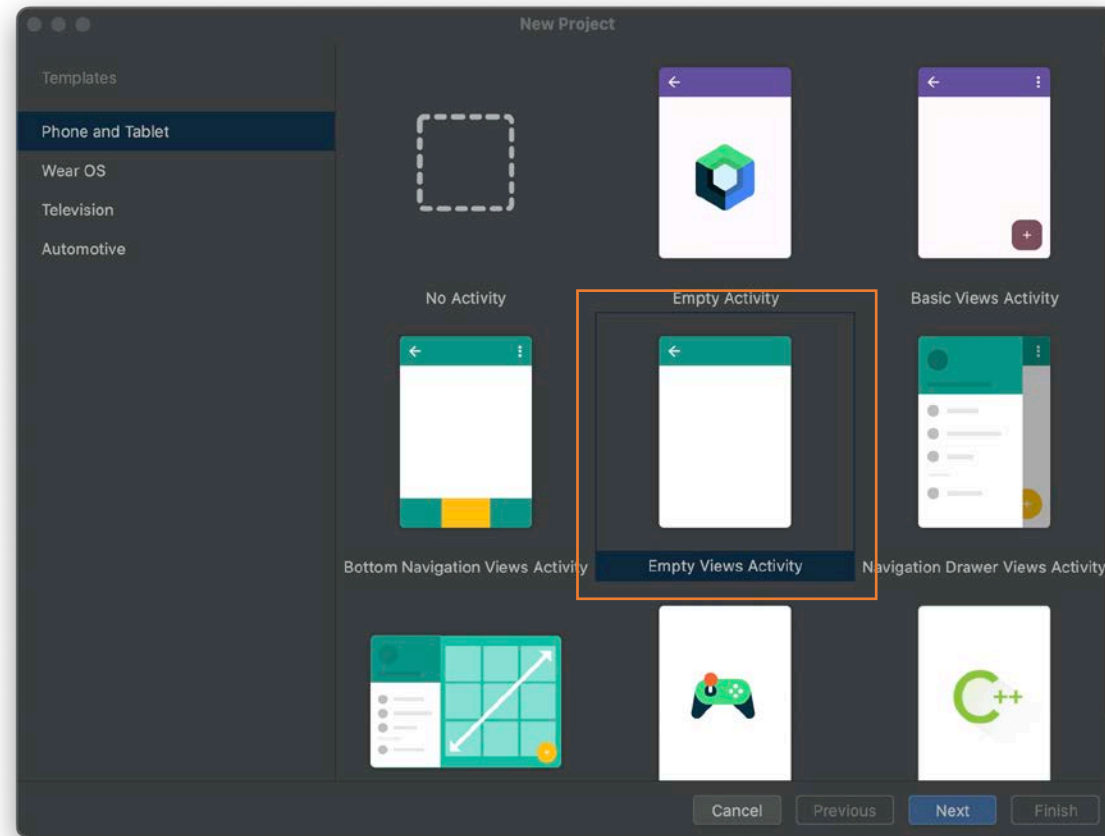
Primeira APP

- Para criar a nova aplicação devemos:
 1. Abrir Android Studio;
 2. Selecionar “New Project”;
 3. Escolher o template “Empty Views Activity”
 4. Alterar o “Name” para um nome sugestivo que permita sabermos qual a aplicação que estamos a criar;
 5. Selecionar a localização da aplicação;
 6. Verificar que a linguagem selecionada é Kotlin.

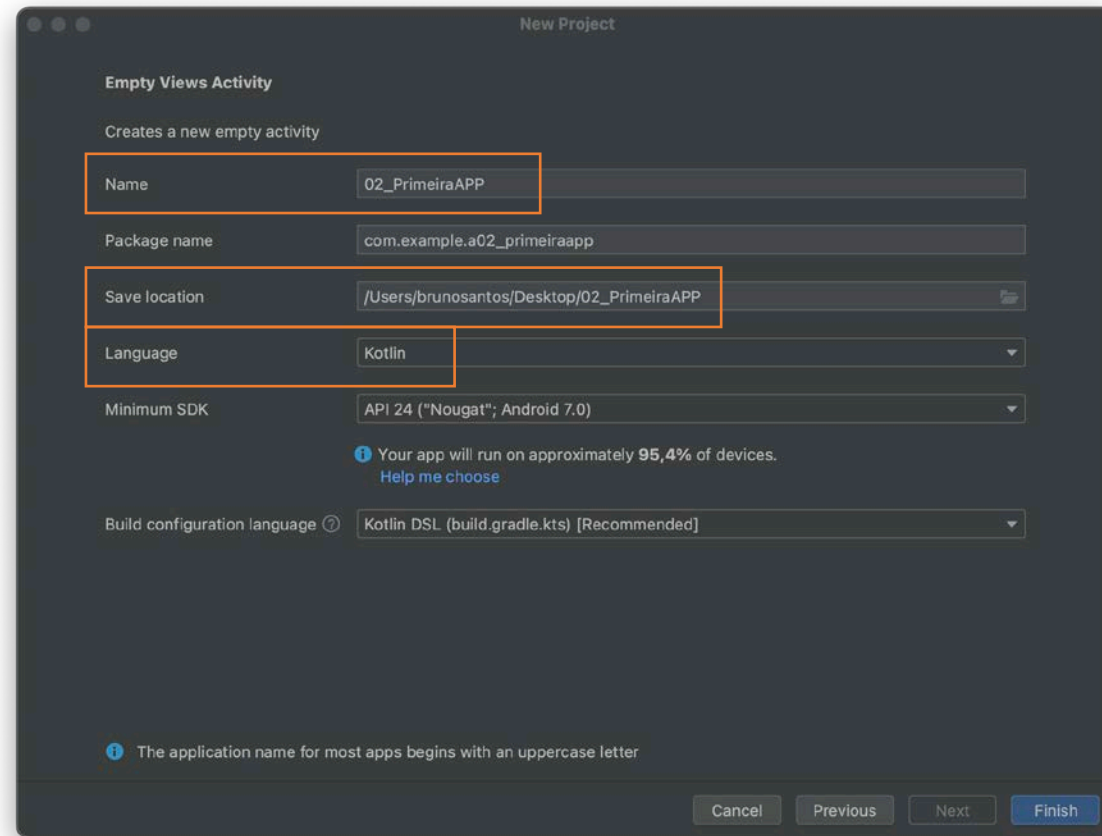
Primeira APP



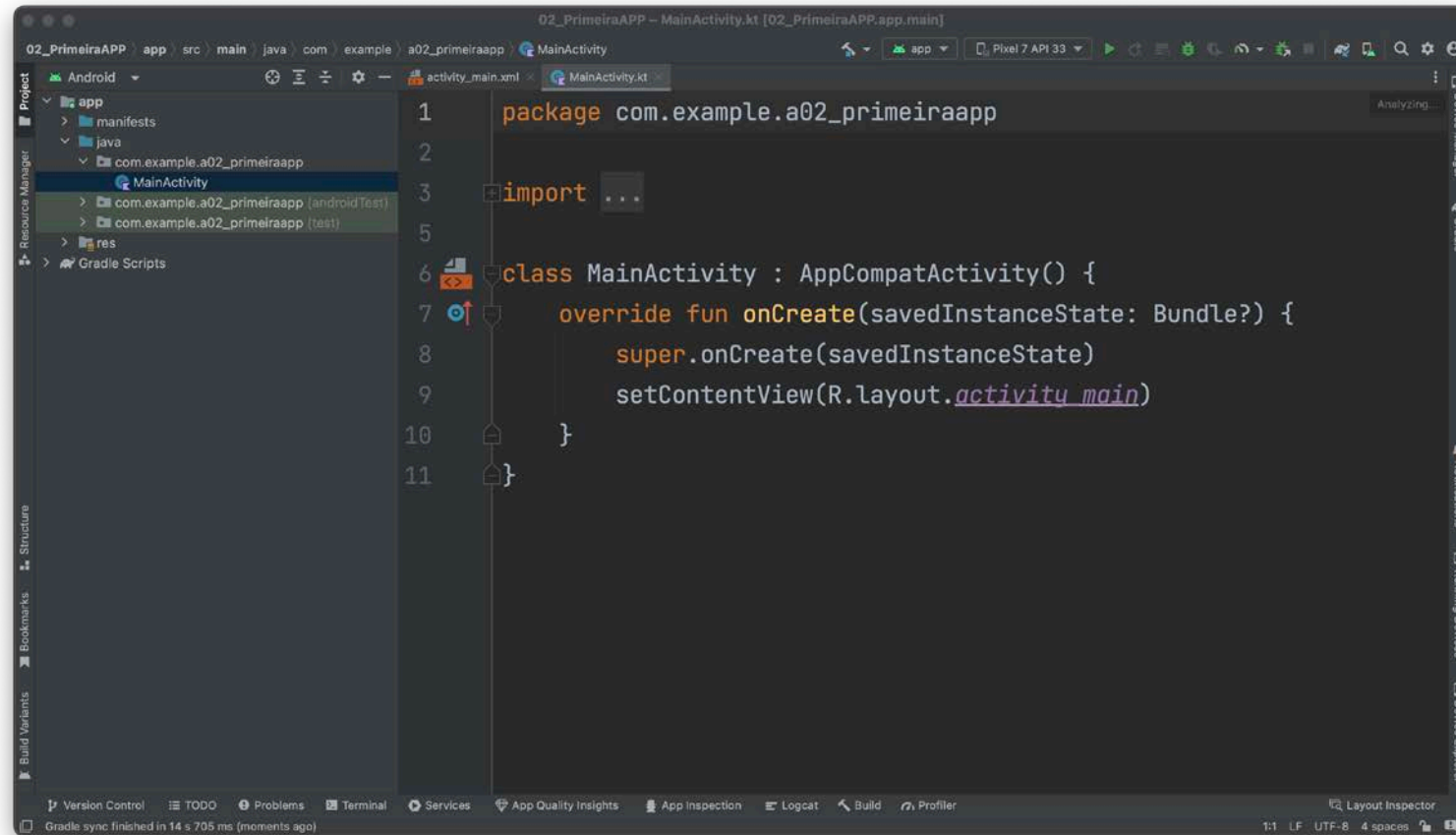
Primeira APP



Primeira APP



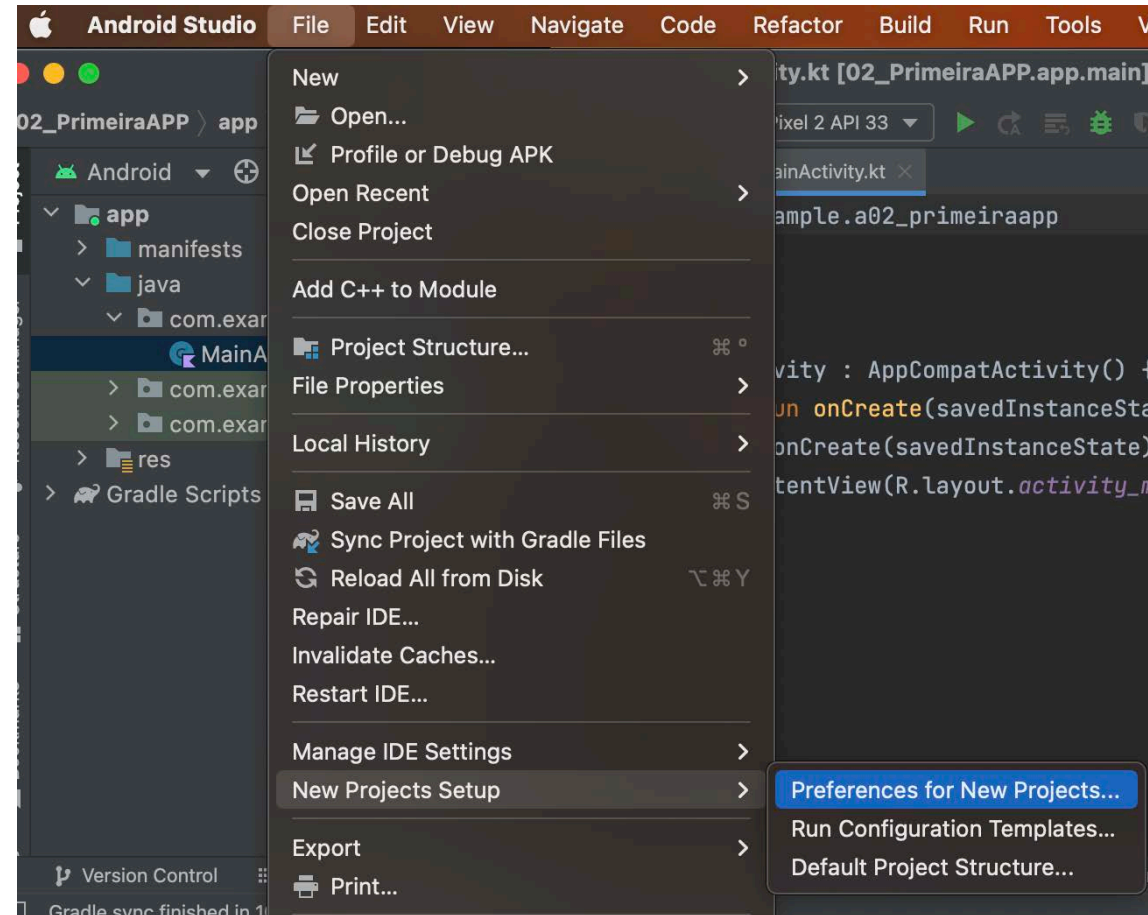
Primeira APP



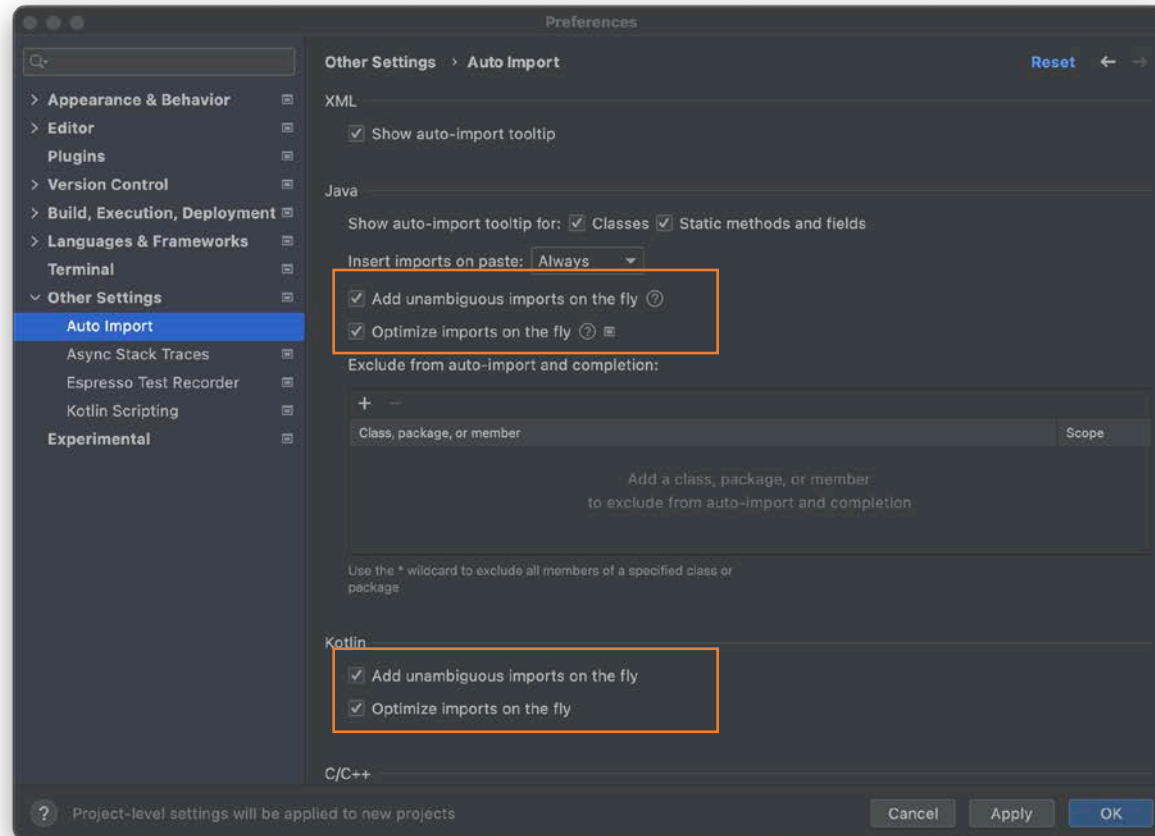
Configurações

- Para que todos os imports sejam feitos de forma automática podemos ativar a opção em:
 1. Menu File → New Project Settings → Preferences for New Projects
 2. Other Settings → Auto Import
 3. Em Kotlin marcar as opções (opcional para Java)
 1. Add unambiguous imports on the fly
 2. Optimize on the fly

Configurações



Configurações



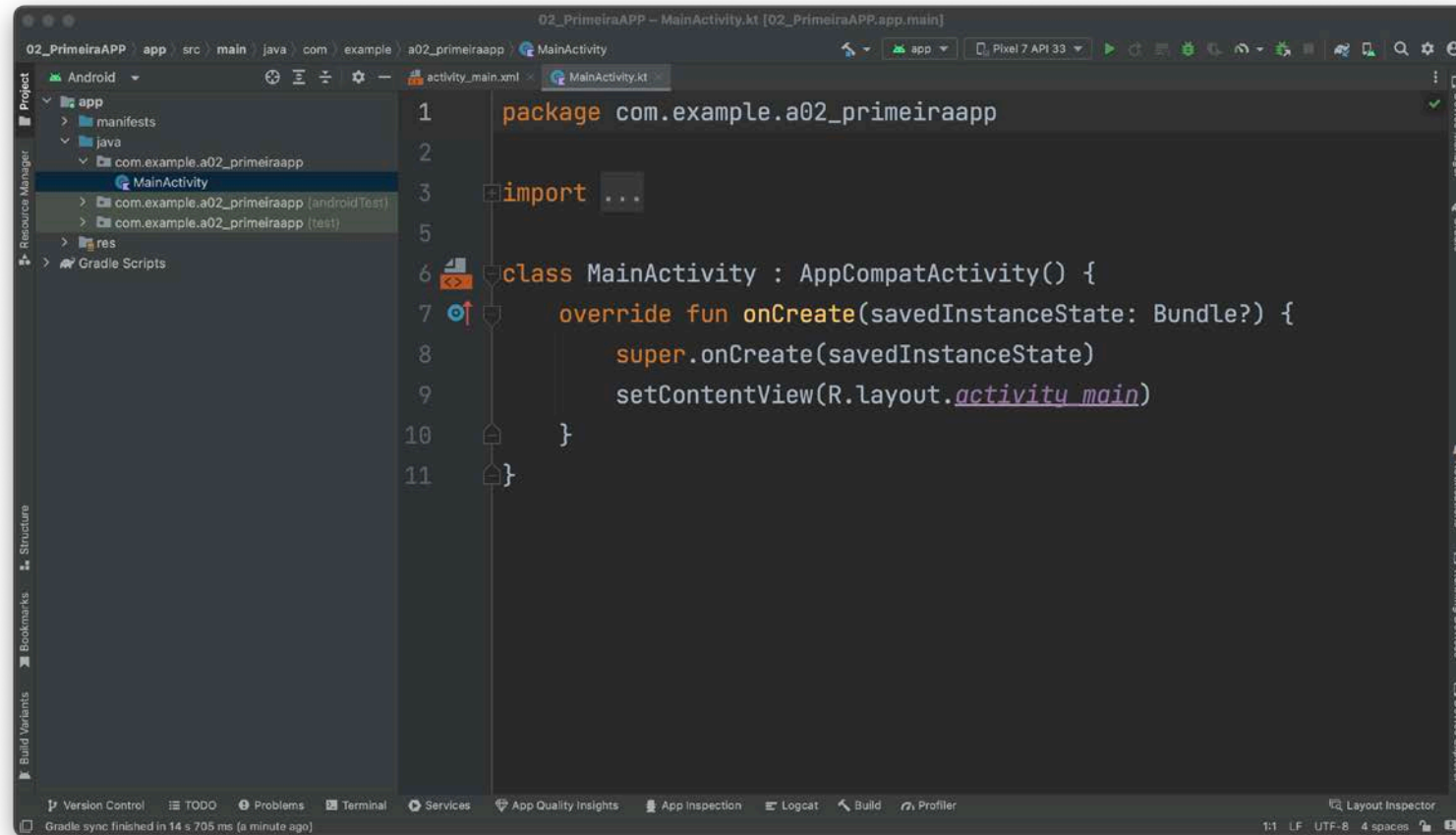
Primeira APP

- Criada a aplicação são apresentados 2 ficheiros abertos:
 - MainActivity.kt
 - activity_main.xml

Primeira APP

- O MainActivity.kt é o ficheiro onde vamos programar todos os eventos e lógica da Activity.
- Mais à frente vamos verificar e perceber todos os elementos presentes no ficheiro.

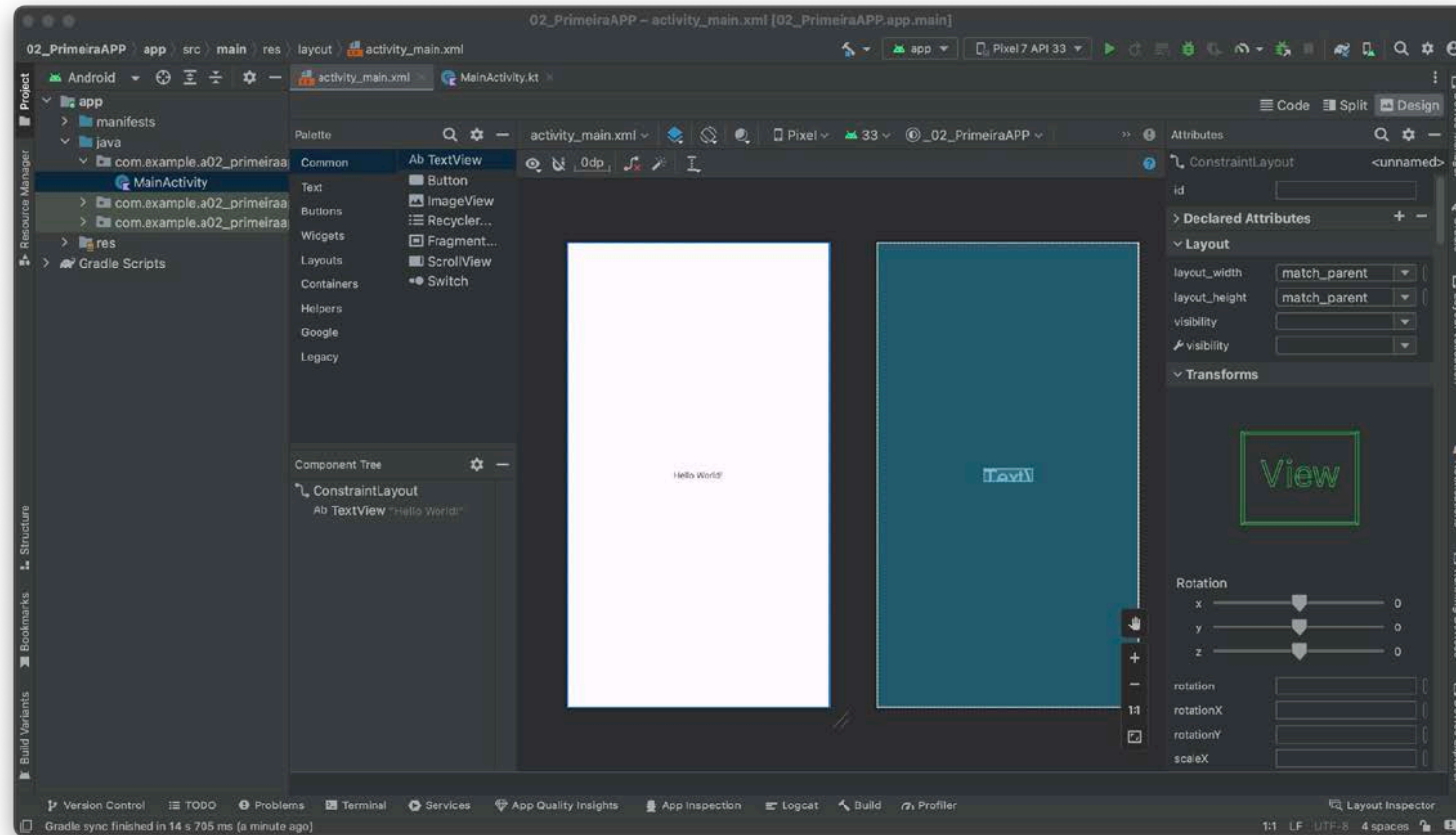
Primeira APP



Primeira APP

- O `activity_main.xml` é o ficheiro de layout onde é desenhada a interface da Activity.
- IMPORANTE: Os ficheiros `MainActivity.kt` e `activity_main.xml` estão interligados, daí os seus nomes serem semelhantes

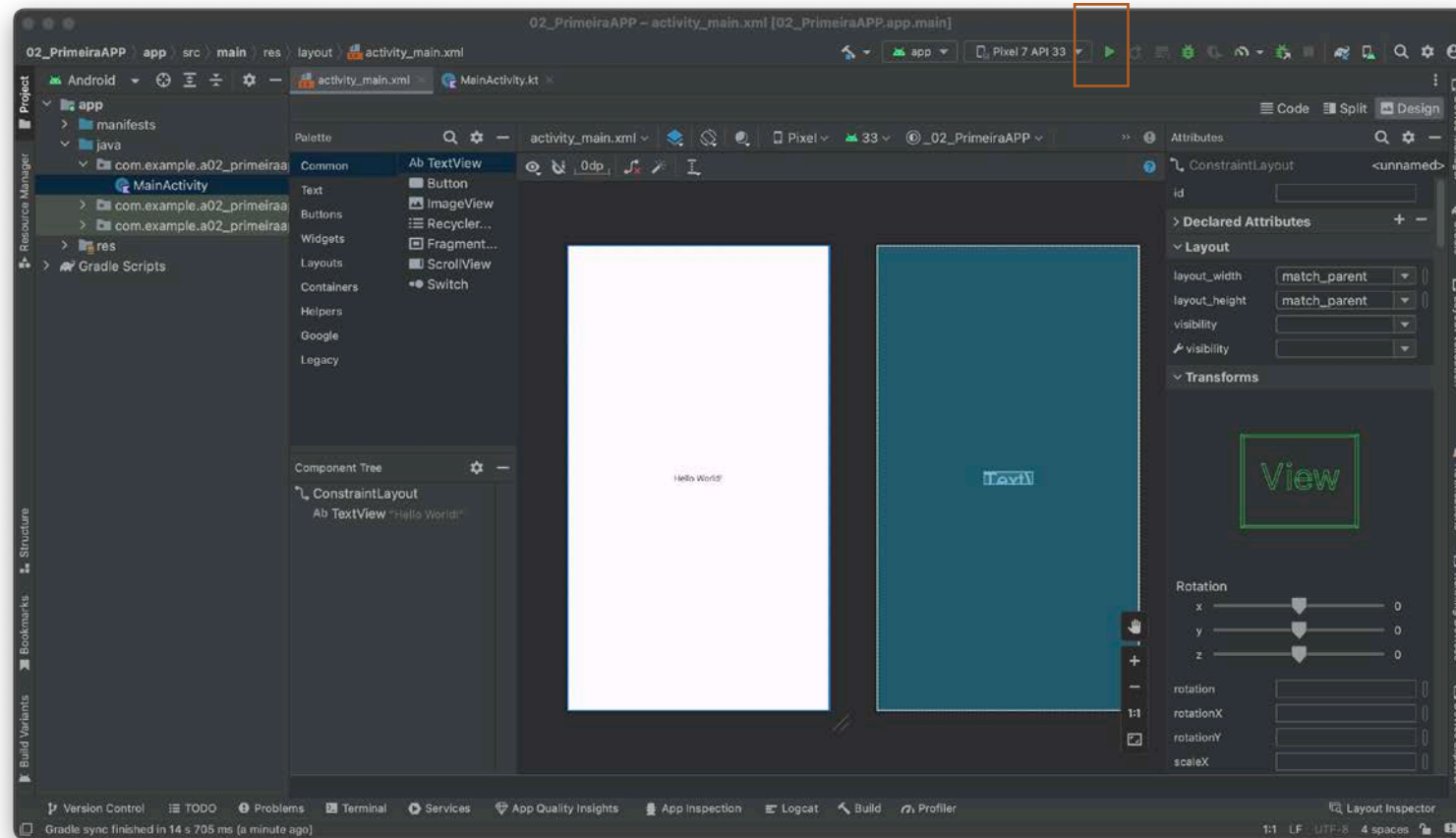
Primeira APP



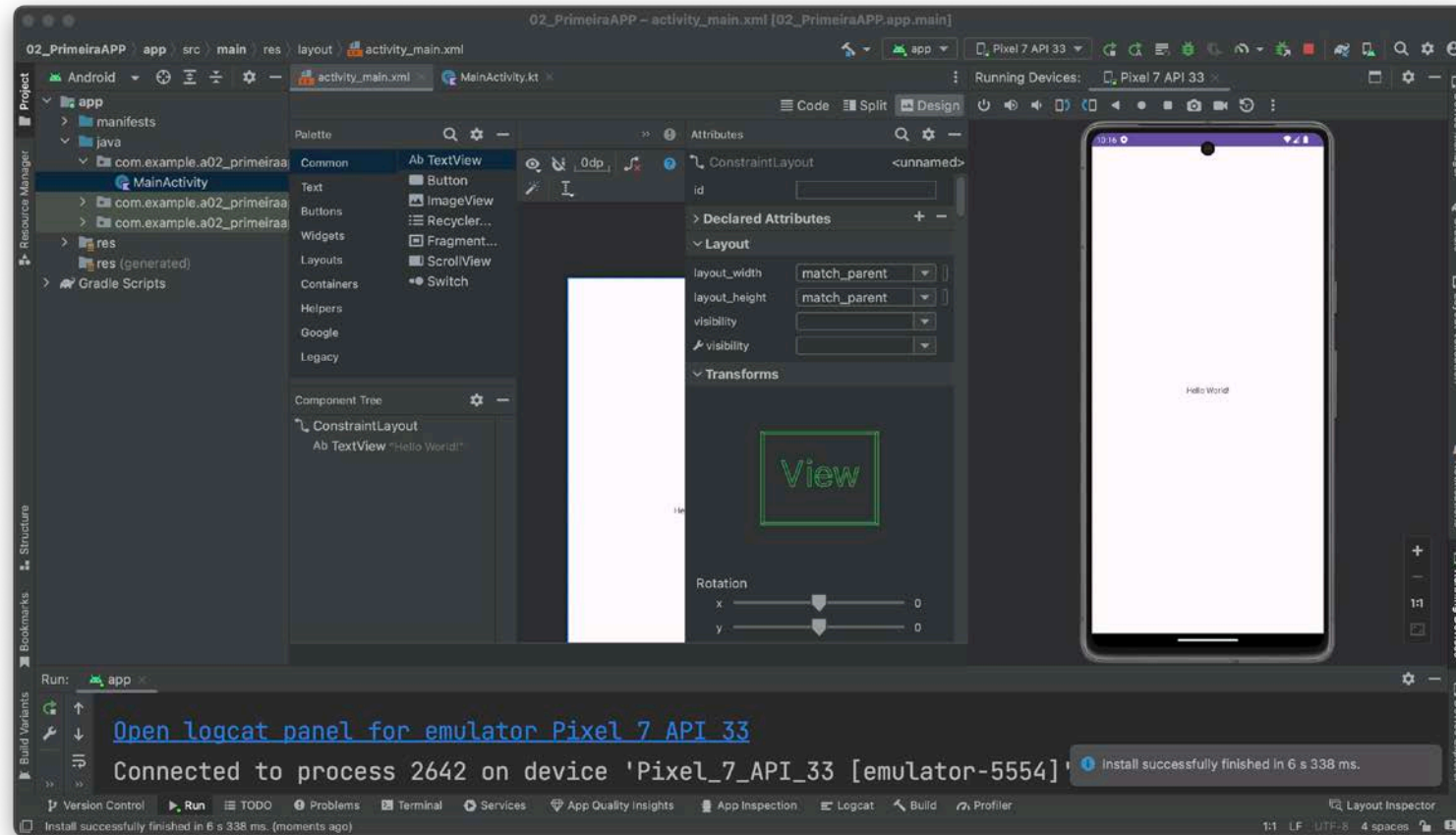
Primeira APP

- Vamos testar a aplicação como ela foi criada.
- Clicamos o botão Run na barra de ferramentas (ou alternativamente menu “Run” e opção “Run ‘app’”, ou Shift + F10).

Primeira APP



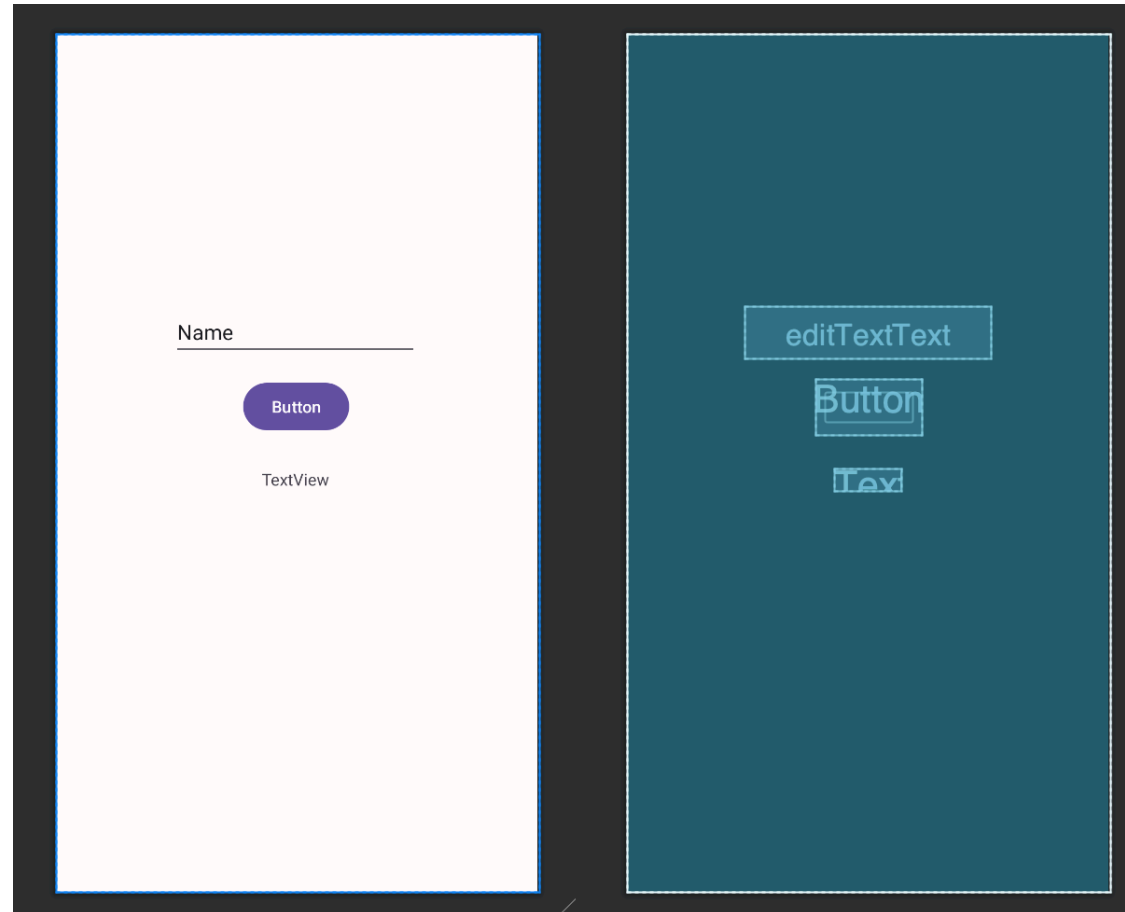
Primeira APP



Primeira APP

- Vamos eliminar a TextView “Hello World!” e acrescentar:
- 1 “Plain Text” (vai ser apresentado como Edit Text no layout) para o utilizador escrever o seu nome)
- 1 “Button” para o utilizador clicar e realizar a operação.
- 1 “TextView” para apresentar o resultado final.

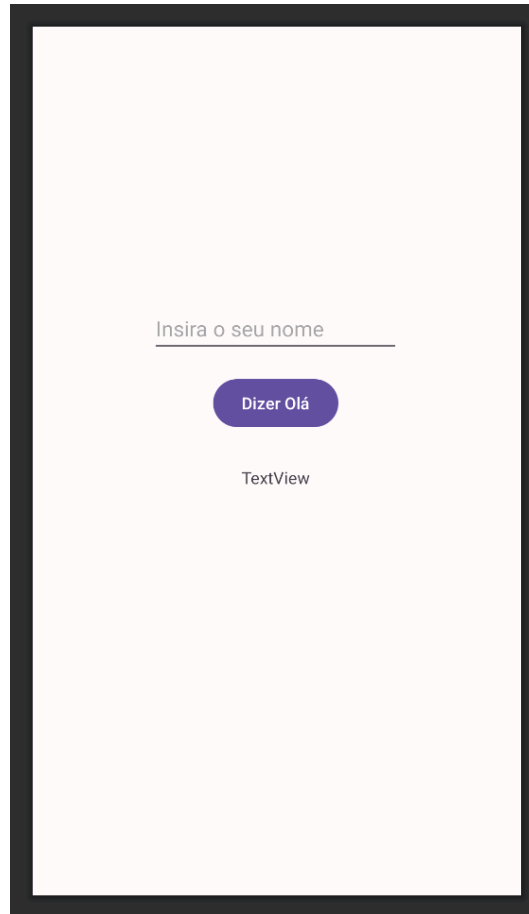
Primeira APP



Primeira APP

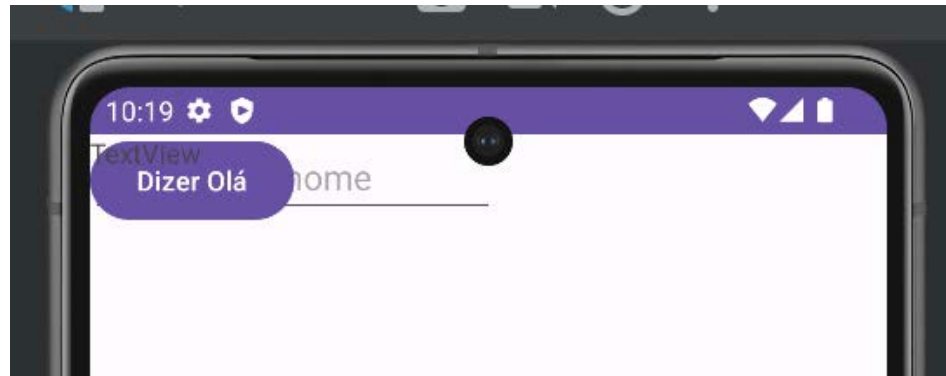
- Seleccionamos o editText e na janela “Attributes” à direita vamos alterar os parâmetros:
 - ID: edit_nome
 - hint: Insira o seu nome
 - text: <vazio>
- Vamos alterar, também, os valores dos restantes elementos:
- Button:
 - ID: button_ola
 - text: Dizer Olá
- TextView:
 - ID: text_resultado

Primeira APP



Primeira APP

- Se executarmos a aplicação a mesma aparecerá desconfigurada.

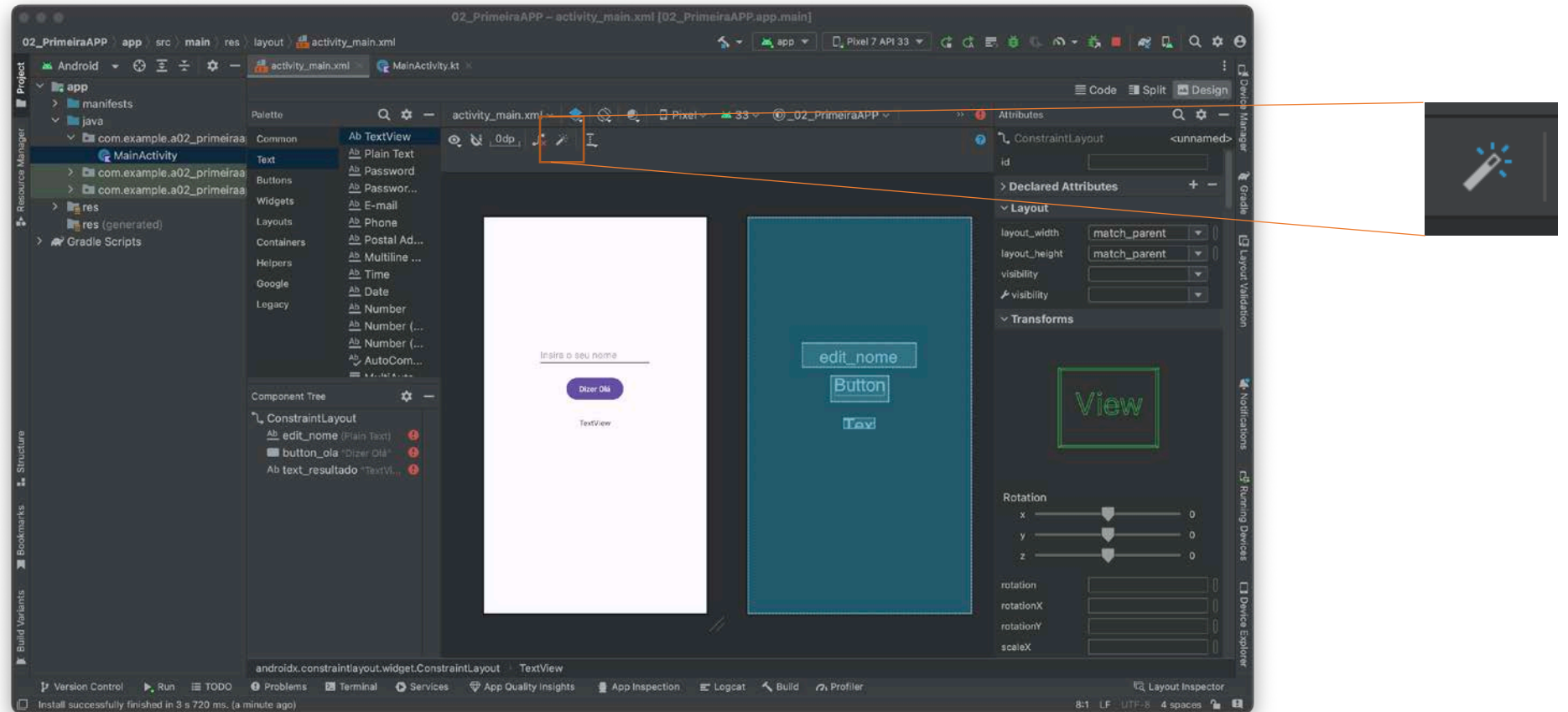


- Para resolver este problema temos de criar as ligações entre componentes do layout.

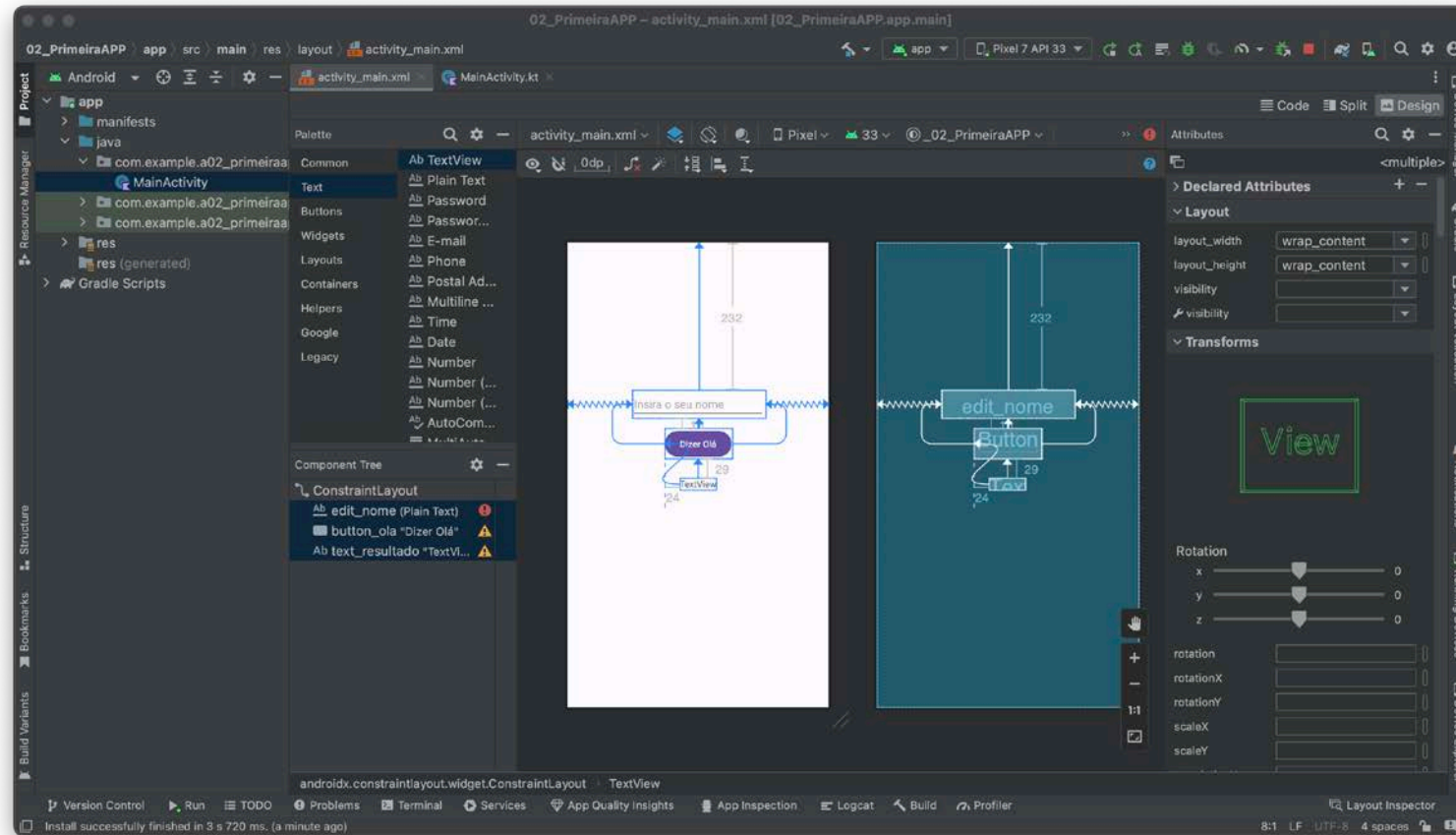
Primeira APP

- Após selecionar todos os elementos do layout deve ser clicado o botão “Infer Constraints”

Primeira APP

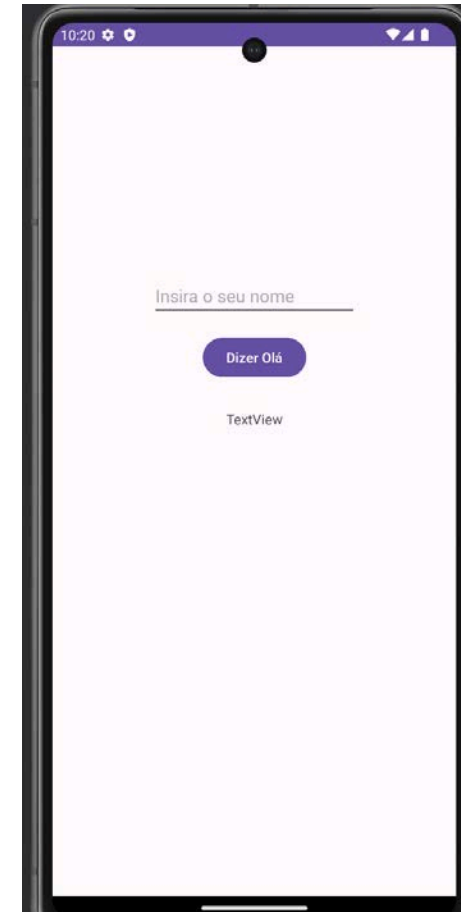


Primeira APP



Primeira APP

- As ligações são criadas de forma automática e executando a aplicação os elementos ficam alinhados de forma semelhante ao apresentado na pré-visualização.
- Recarregue a aplicação



Primeira APP

- Vamos passar para a programação de eventos no ficheiro MainActivity.kt
- Em primeiro lugar temos de criar uma ligação entre o ficheiro MainActivity.kt e os elementos gráficos presentes no layout activity_main.xml

ViewBinding

“A vinculação de visualizações é um recurso que facilita a programação de códigos que interagem com visualizações. Quando a vinculação de visualizações é ativada em um módulo, ela gera uma classe de vinculação para cada arquivo de layout XML presente nesse módulo. A instância de uma classe de vinculação contém referências diretas a todas as visualizações que têm um código no layout correspondente.”

“Na maioria dos casos, a vinculação de visualizações substitui findViewById.”

<https://developer.android.com/topic/libraries/view-binding>

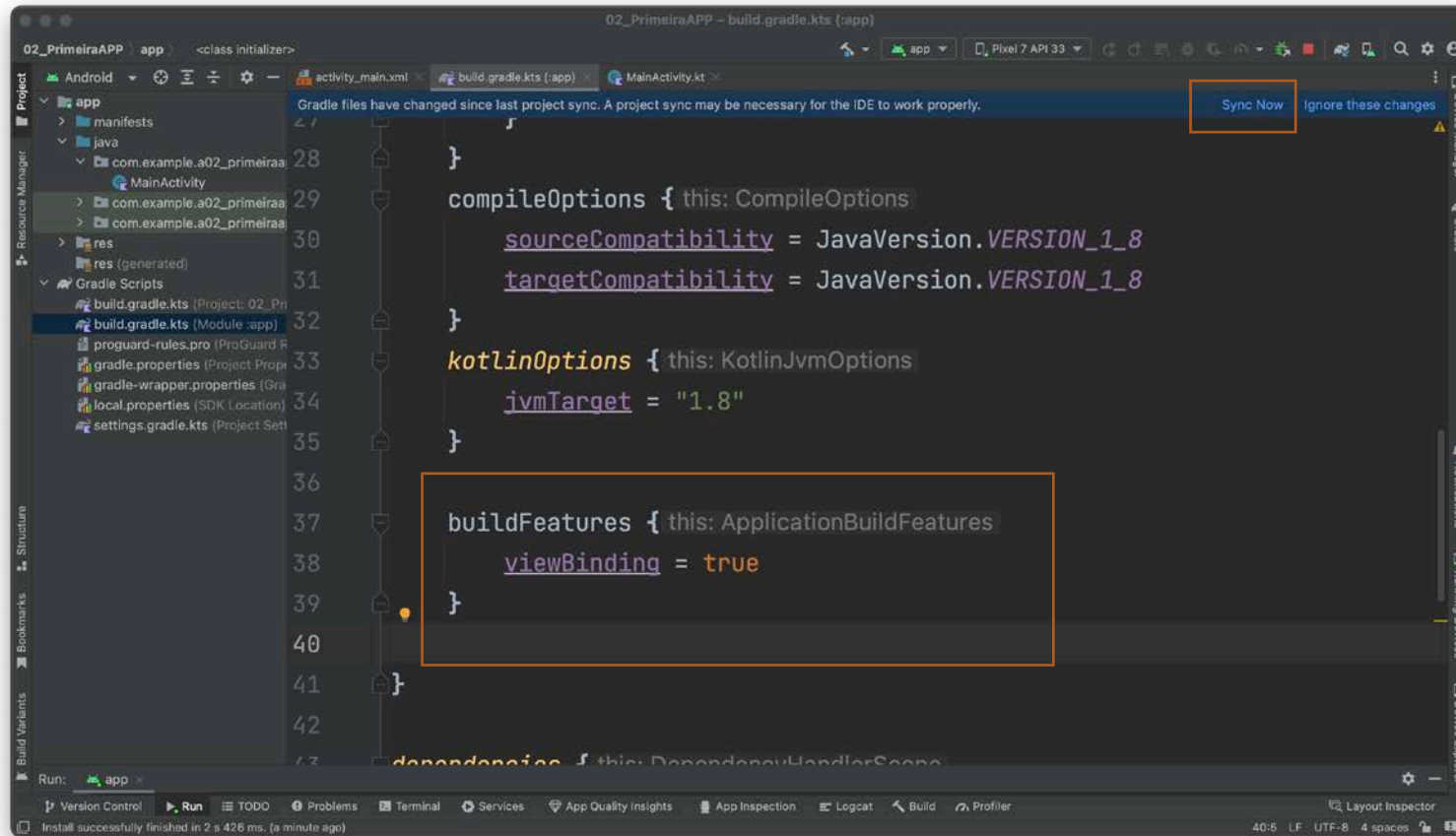
ViewBinding

- À esquerda, nos ficheiros do projeto vamos abrir:
 1. Gradle Scripts → build.gradle (Module: ...)
 2. Após a chaveta de fecho do kotlinOptions adicionar:

```
buildFeatures {  
    viewBinding = true  
}
```

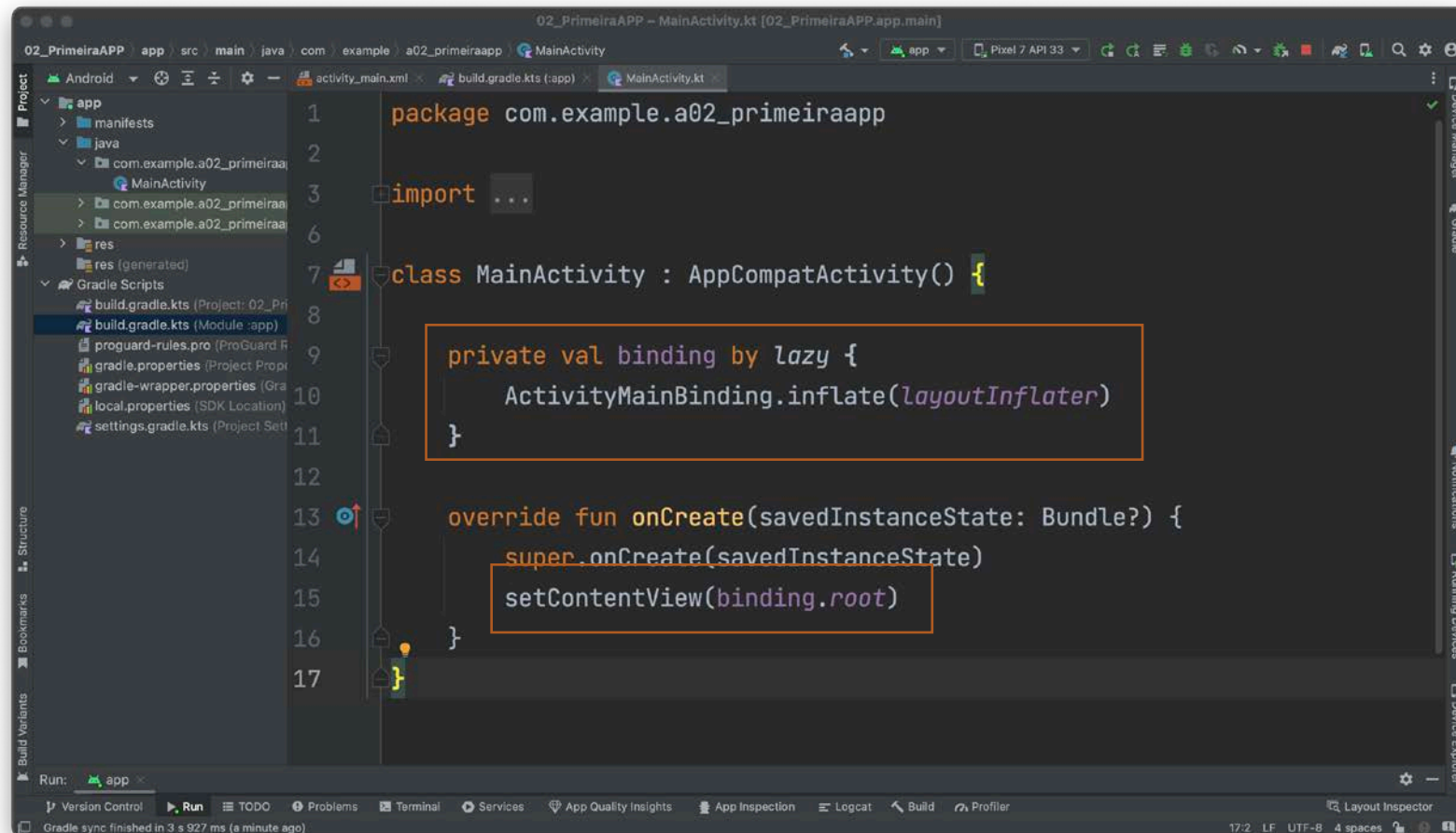
3. Clicar no link Sync Now do topo do ficheiro para sincronizar e importar o ViewBinding.

ViewBinding



Primeira APP

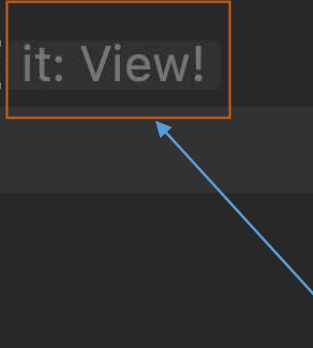
- No ficheiro MainActivity.kt vamos implementar o ViewBinding



Primeira APP

- De seguida vamos criar e configurar o evento de clique no botão. Para isso devemos escrever:

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(binding.root)  
  
    binding.button01a.setOnClickListener { it: View!  
  
    }  
}
```



Criado automaticamente, não é escrito, apenas placeholder

Primeira APP

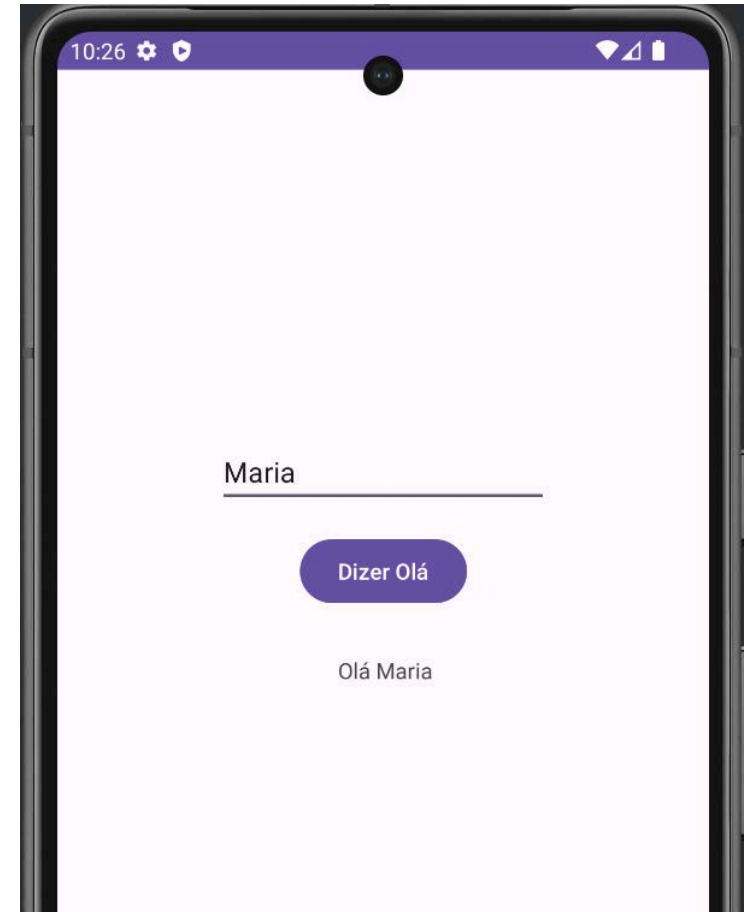
- Dentro das chavetas vamos ler o valor inserido na EditText e apresentar o valor concatenado com “Olá” no TextView

Primeira APP

```
binding.buttonOla.setOnClickListener { it: View!  
    var nome: String = binding.editNome.text.toString()  
    binding.textResultado.text = "Olá $nome"  
}
```

Variáveis não alteradas podem ser definidas como constantes trocando var por val

Não é obrigatório definir :String pois o tipo de dados será inferido automaticamente



Exercício 1

- Crie uma aplicação que peça ao utilizador um valor em euros e o converta para um valor em dólares.
- Pode dar um valor arbitrário à taxa de conversão.
- Considere que os valores inseridos serão decimais (double).

Nota: para ler o valor como double deve utilizar código semelhante a

```
val euros: Double = binding.etValorEuros.text.toString().toDouble()
```

Exercício 2

- Altere ou crie uma nova aplicação semelhante à anterior e realize as seguintes operações:
 1. Centre o conteúdo do EditText (ver parâmetro gravity);
 2. Altere, na janela de “Attributes” o tamanho do texto da TextView onde será apresentado o resultado;
 3. Arredonde o valor apresentado na TextView para aparecerem apenas 2 casas decimais (ver exemplo slide seguinte)
 4. Na TextView, quando é apresentado o valor em dólares, o mesmo, no final deve apresentar o símbolo \$.

Arredondar valor

- Propostas de arredondamento a 2 casas decimais:

```
val dolares: Double = (euros * 0.80 * 100).roundToInt().toDouble() / 100;
```

ou

```
val dolares: Double = String.format("%.2f", euros * 0.8).toDouble()
```

Exercício 3

- Crie uma nova aplicação que permita ao utilizador inserir um valor de temperatura em graus Celsius e mostre o seu equivalente em graus Fahrenheit, considerando que a fórmula de conversão é:

$$\text{Fahrenheit} = \text{Celsius} \times 1,8 + 32$$