



Reskilling 4Employment Software Developer

Acesso móvel a sistemas de informação

Bruno Santos

bruno.santos.mcv@msft.cesae.pt

Tópicos

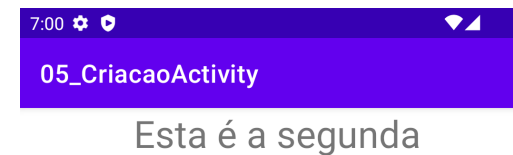
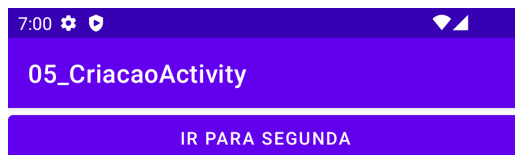
- Intent
 - Extras
 - Retorno a Activity

Intent (revisão)

- Os Intent são usados para navegar entre Activity.

```
startActivity(Intent(this, MainActivity2::class.java))
```

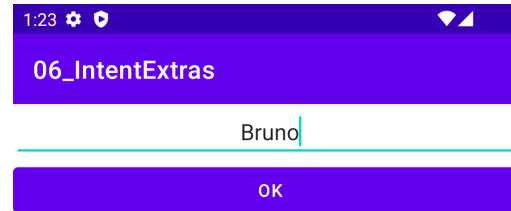
Intent (revisão)



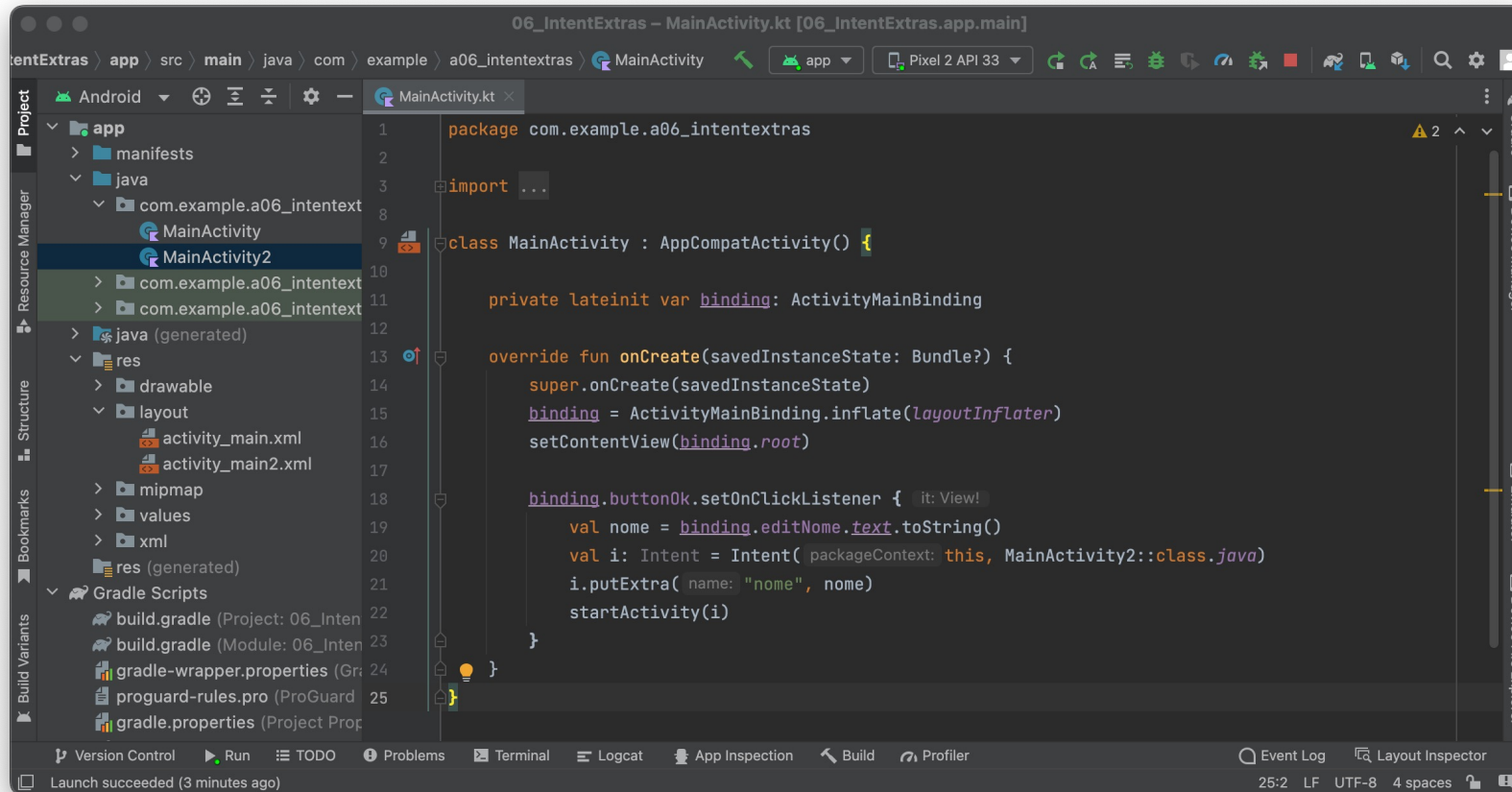
Intent - Extras

- Agora queremos passar valores entre Activity, para isso vamos usar os extras que estão associados aos Intent.
- Temos duas Activity, sendo que a primeira tem um campo para o utilizador inserir o seu nome e um botão que ao ser clicado irá passar para a nova Activity e irá levar o nome do utilizador.

Intent - Extras



Intent - Extras

A screenshot of the Android Studio IDE. The main editor window displays the MainActivity.kt file. The code defines a package, imports necessary classes, and implements the MainActivity class which extends AppCompatActivity. It includes an onCreate method that inflates a binding, sets the content view, and starts MainActivity2 with an Intent that carries an extra named 'nome'. The left sidebar shows the Project, Resource Manager, and Structure views. The bottom status bar indicates 'Launch succeeded (3 minutes ago)' and the file encoding is UTF-8 with 4 spaces.

```
06_IntentExtras - MainActivity.kt [06_IntentExtras.app.main]
06_IntentExtras > app > src > main > java > com > example > a06_intentextras > MainActivity
MainActivity
Pixel 2 API 33

1 package com.example.a06_intentextras
2
3 import ...
4
5 class MainActivity : AppCompatActivity() {
6
7     private lateinit var binding: ActivityMainBinding
8
9     override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11         binding = ActivityMainBinding.inflate(layoutInflater)
12         setContentView(binding.root)
13
14         binding.buttonOk.setOnClickListener { it: View!
15             val nome = binding.editNome.text.toString()
16             val i: Intent = Intent( packageContext: this, MainActivity2::class.java)
17             i.putExtra( name: "nome", nome)
18             startActivity(i)
19         }
20     }
21 }
22
23
24
25
```

Intent - Extras

- Foi acrescentado um extra para ser passado o valor do nome inserido pelo utilizador

```
val nome = binding.editNome.text.toString()
val i: Intent = Intent(packageContext: this, MainActivity2::class.java)
i.putExtra(name: "nome", nome)
startActivity(i)
```


Intent - Extras

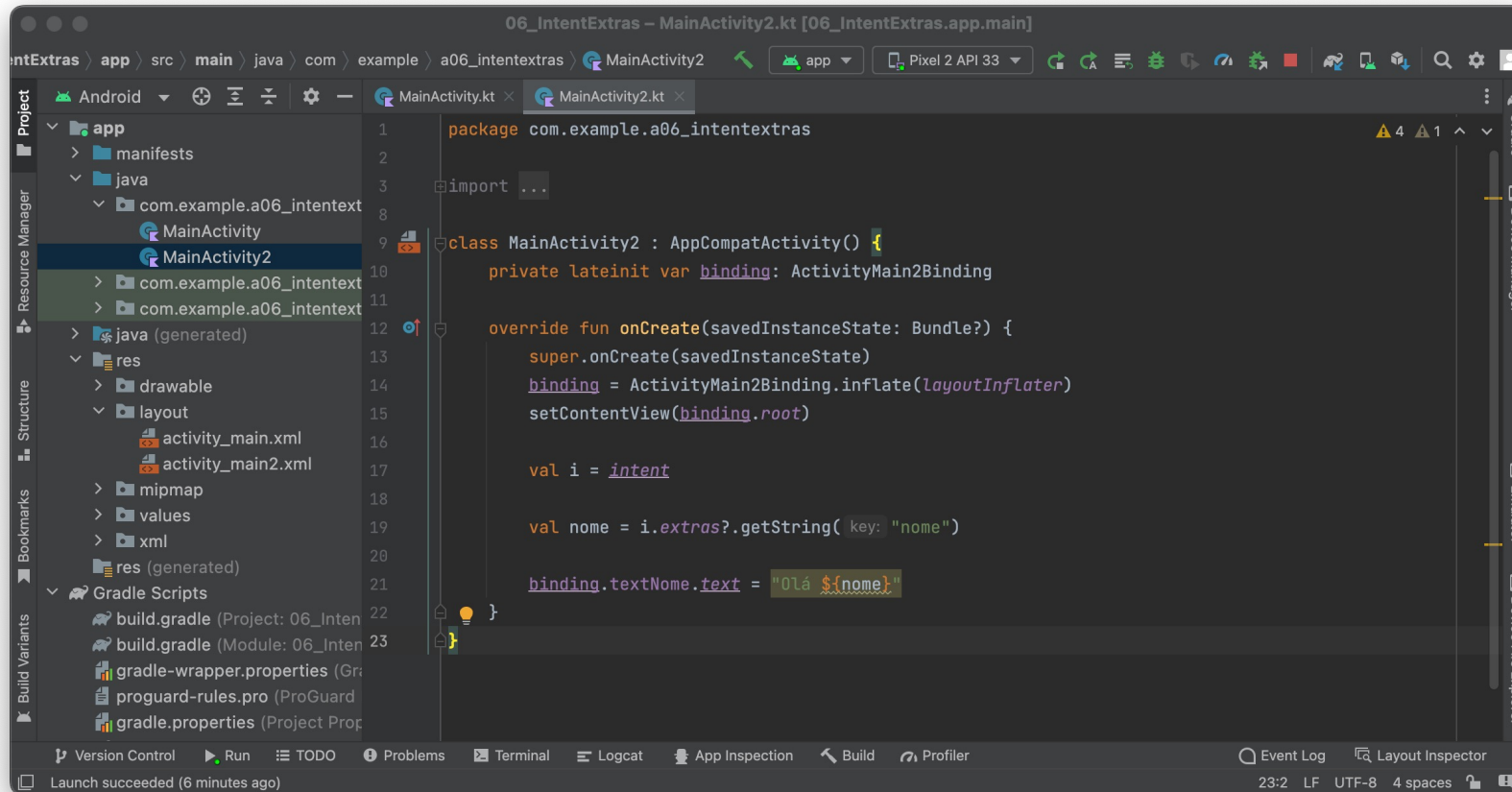
- “i” é o nome dado ao Intent
- “.putExtra” é o método chamado associado ao Intent
- “nome” é a chave que vamos dar ao elemento que queremos passar, será apenas para que do outro lado consigamos ir buscar o valor
- nome é o valor inserido na EditText

```
val nome = binding.editNome.text.toString()
val i: Intent = Intent(packageContext: this, MainActivity2::class.java)
i.putExtra(name: "nome", nome)
startActivity(i)
```

Intent - Extras

- Na segunda Activity temos apenas uma TextView com o texto “Olá” e que será complementado pelo nome passado da primeira Activity.
- Temos de carregar o Intent que vem da primeira e aceder a todos os Extras enviados pela mesma.
- Nota: se não precisarmos de aceder aos extras não necessitamos de carregar o Intent.

Intent - Extras



```
06_IntentExtras - MainActivity2.kt [06_IntentExtras.app.main]
Android > app > src > main > java > com > example > a06_intentextras > MainActivity2
Pixel 2 API 33
MainActivity.kt x MainActivity2.kt x

Project
  app
    manifests
    java
      com.example.a06_intentextras
        MainActivity
        MainActivity2
      com.example.a06_intentextras
      com.example.a06_intentextras
    java (generated)
    res
      drawable
      layout
        activity_main.xml
        activity_main2.xml
      mipmap
      values
      xml
    res (generated)
  Gradle Scripts
    build.gradle (Project: 06_IntentExtras)
    build.gradle (Module: 06_IntentExtras)
    gradle-wrapper.properties (Gradle)
    proguard-rules.pro (ProGuard)
    gradle.properties (Project Properties)

Resource Manager
Structure
Bookmarks
Build Variants
  build.gradle (Project: 06_IntentExtras)
  build.gradle (Module: 06_IntentExtras)
  gradle-wrapper.properties (Gradle)
  proguard-rules.pro (ProGuard)
  gradle.properties (Project Properties)

1 package com.example.a06_intentextras
2
3 import ...
4
5 class MainActivity2 : AppCompatActivity() {
6     private lateinit var binding: ActivityMain2Binding
7
8     override fun onCreate(savedInstanceState: Bundle?) {
9         super.onCreate(savedInstanceState)
10        binding = ActivityMain2Binding.inflate(layoutInflater)
11        setContentView(binding.root)
12
13        val i = intent
14
15        val nome = i.extras?.getString(key: "nome")
16
17        binding.textNome.text = "Olá ${nome}"
18    }
19
20
21
22
23
Launch succeeded (6 minutes ago)
23:2 LF UTF-8 4 spaces
```

Intent - Extras

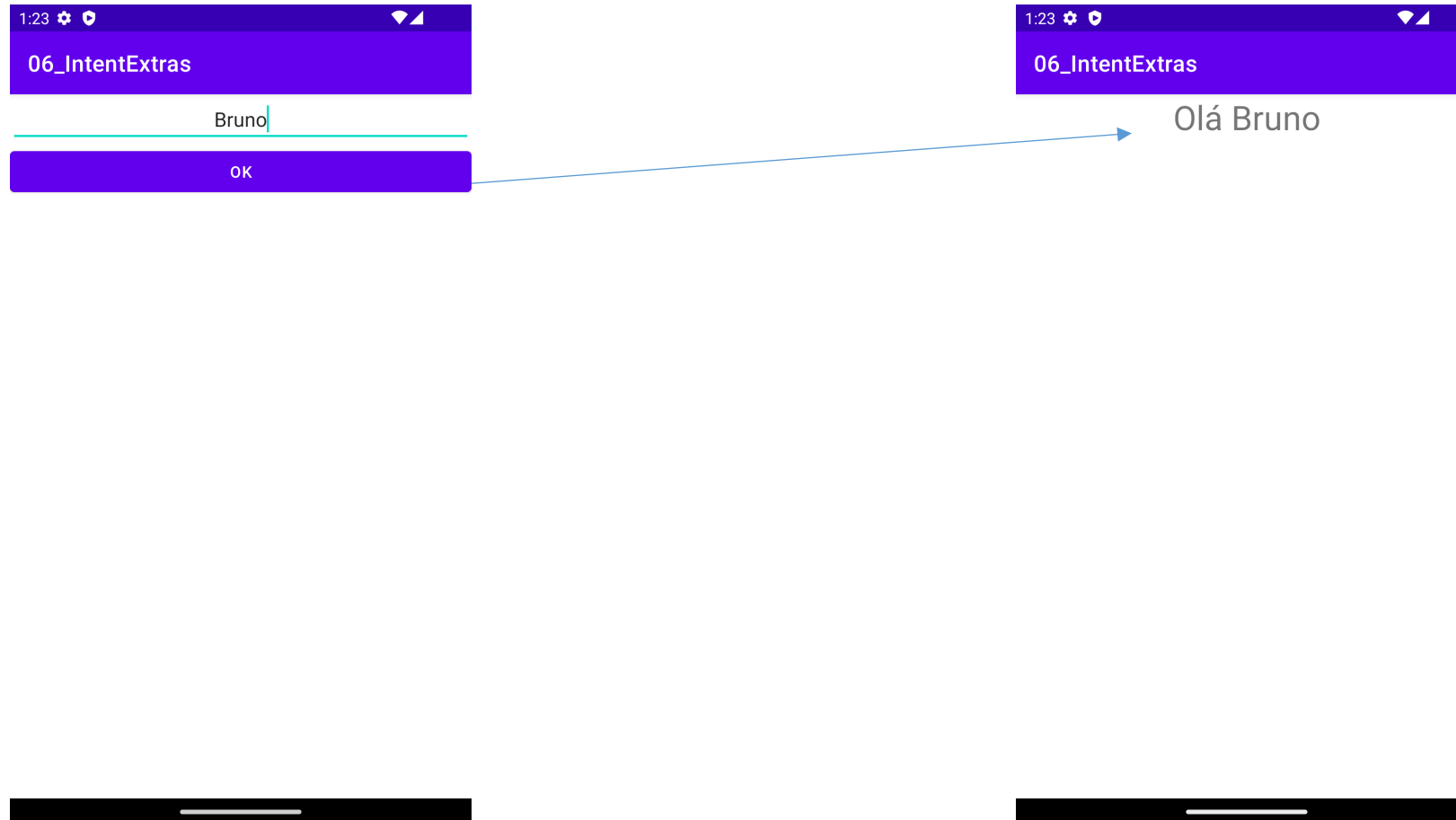
- “val i = intent” – criamos um intent com o nome i e associamos o que vem da Activity anterior
- “i.extras?” – permite aceder a todos os extras enviados da Activity anterior. O ? simboliza a possibilidade de vir a nulo, tratando essa exceção.
- “.getString(“nome”) – carrega o valor do nome

```
val i = intent

val nome = i.extras?.getString( key: "nome")

binding.textNome.text = "Olá ${nome}"
```

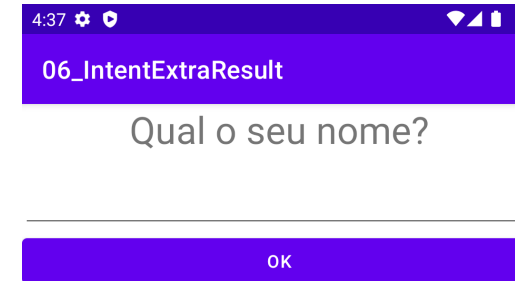
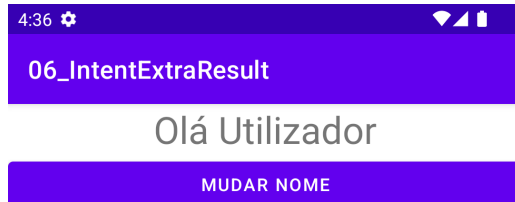
Intent - Extras



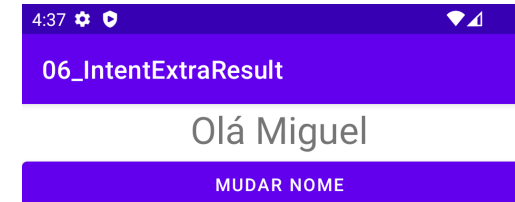
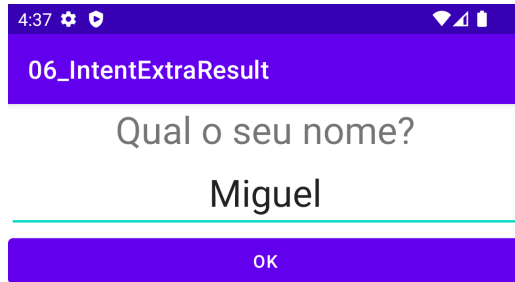
Intent - Retorno de Extras

- Vamos supor que temos agora um cenário diferente:
- A primeira Activity tem uma TextView com a saudação Olá <nome da pessoa> e um botão no qual é possível passar para a segunda Activity onde podemos alterar o nome da pessoa.
- A segunda Activity tem uma EditText que deve receber o nome passado da primeira Activity e quando clicado no botão alterar deve voltar para a primeira Activity e alterar o nome na saudação.
- O processo deve repetir-se sempre que é passado de uma Activity para outra.

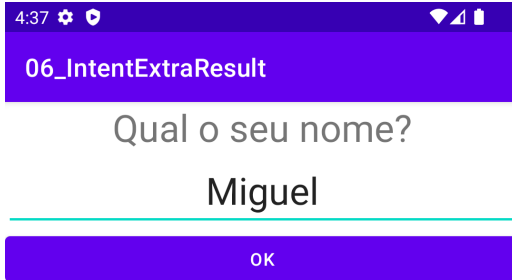
Intent - Retorno de Extras



Intent - Retorno de Extras



Intent - Retorno de Extras



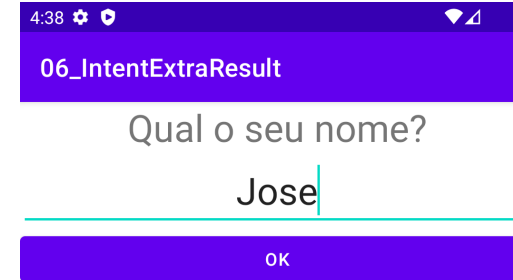
4:37

06_IntentExtraResult

Qual o seu nome?

Miguel

OK



4:38

06_IntentExtraResult

Qual o seu nome?

Jose

OK

Intent - Retorno de Extras



Intent - Retorno de Extras

- Na primeira Activity vamos criar uma variável para guardar o valor do nome do utilizador e criar o evento de clique no botão. Aqui vamos fazer uma alteração ao Intent, não vamos fazer startActivity...

Intent - Retorno de Extras

```
private lateinit var binding: ActivityMainBinding
private lateinit var result: ActivityResultLauncher<Intent>
private var nome = ""

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    binding = ActivityMainBinding.inflate(layoutInflater)
    setContentView(binding.root)

    binding.buttonMudarNome.setOnClickListener { it: View!
        val i = Intent(packageContext: this, MainActivity2::class.java)
        i.putExtra(name: "nome", nome)
        result.launch(i)
    }
```

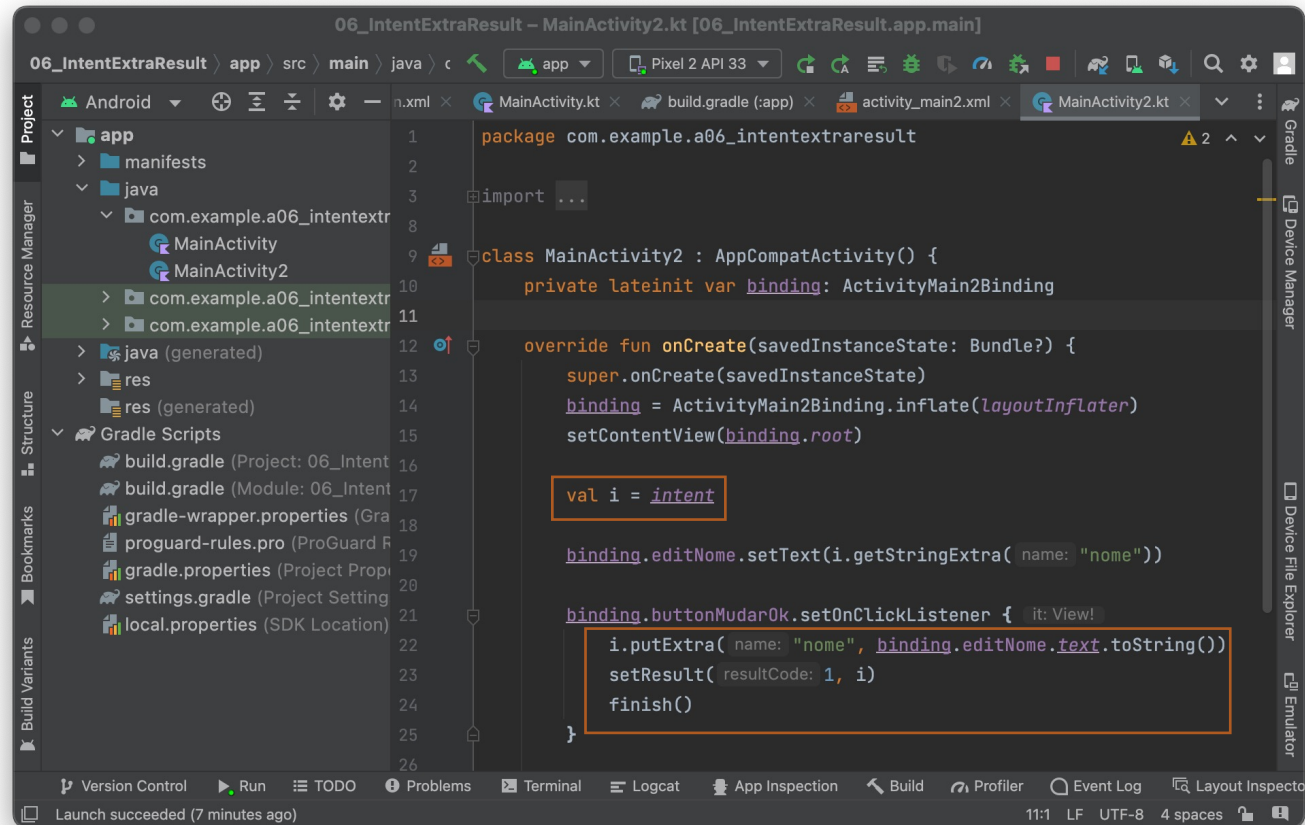
Intent - Retorno de Extras

- “ActivityResultLauncher” é a variável que vai permitir iniciarmos uma nova Activity com a perspetiva de voltar e realizar algum procedimento.
- “result.launch(i)” – não fazemos startActivity uma vez que queremos quando voltar retornar à variável result

Intent - Retorno de Extras

- Na segunda Activity vamos buscar o Intent, atualizar o valor com o Extra e programar o evento de clique no botão.
- Neste caso vamos atualizar o valor do nome dentro do Intent, fazemos setResult e finish.

Intent - Retorno de Extras



```
06_IntentExtraResult - MainActivity2.kt [06_IntentExtraResult.app.main]
06_IntentExtraResult > app > src > main > java > c
Pixel 2 API 33
MainActivity.kt x build.gradle (-app) x activity_main2.xml x MainActivity2.kt x
Project
  app
    manifests
    java
      com.example.a06_intentextr
        MainActivity
        MainActivity2
      com.example.a06_intentextr
      com.example.a06_intentextr
    java (generated)
    res
      res (generated)
    Gradle Scripts
      build.gradle (Project: 06_Intent
      build.gradle (Module: 06_Intent
      gradle-wrapper.properties (Gra
      proguard-rules.pro (ProGuard F
      gradle.properties (Project Prop
      settings.gradle (Project Setting
      local.properties (SDK Location)
Resource Manager
Structure
Bookmarks
Build Variants
1 package com.example.a06_intentextr
2
3 import ...
4
5 class MainActivity2 : AppCompatActivity() {
6     private lateinit var binding: ActivityMain2Binding
7
8     override fun onCreate(savedInstanceState: Bundle?) {
9         super.onCreate(savedInstanceState)
10        binding = ActivityMain2Binding.inflate(layoutInflater)
11        setContentView(binding.root)
12
13        val i = intent
14
15        binding.editNome.setText(i.getStringExtra( name: "nome"))
16
17        binding.buttonMudarOk.setOnClickListener { it: View!
18            i.putExtra( name: "nome", binding.editNome.text.toString())
19            setResult( resultCode: 1, i)
20            finish()
21        }
22    }
23
24
25
26
Launch succeeded (7 minutes ago)
11:1 LF UTF-8 4 spaces
```

Intent - Retorno de Extras

- “setResult(1,i)” informa qual o resultCode, que à semelhança do requestCode serve apenas para controlo da execução, o “i” é o Intent que estávamos a usar e ao qual atualizamos o nome.
- “finish()” é o comando que termina esta Activity e volta para a anterior.

Intent - Retorno de Extras

- Adicionamos ainda informação do que queremos realizar no retorno a primeira Activity

```
result = registerForActivityResult(ActivityResultContracts.StartActivityForResult()) {  
    if (it.data != null && it.resultCode == 1) {  
        nome = it.data?.getStringExtra( name: "nome").toString()  
        binding.textNome.text = "Olá #{nome}"  
    }  
}
```

Exercício 1

- Crie uma aplicação que peça ao utilizador os seus dados pessoais (nome, morada, telefone, email, género) e que ao clicar num botão transporte os mesmos para uma segunda Activity em que apresenta uma mensagem semelhante a:
- O José mora na rua do sobe e desce, tem o telefone 222333444 e o email jose@gmail.com

Exercício 2

- Crie uma aplicação com 3 Activity.
- Na primeira Activity deve ser pedido ao utilizador para inserir um número e clicando no botão OK saltar para a segunda Activity.
- Na segunda deve ser pedido um novo número e clicando no botão OK saltar para a terceira Activity
- Na terceira devem aparecer quatro resultados:
 - A soma dos dois números
 - A diferença dos dois números
 - O produto dos dois números
 - A divisão inteira dos dois números
- Nota: deve aparecer no formato: $4+5=9$

Exercício 3

- Crie uma aplicação que se inicie com um SplashScreen, de seguida são apresentados 3 botões que redirecionam para 3 Activity: Registo, Login e Sobre.
- O Registo deve permitir ao utilizador definir um username e uma password de acesso à sua aplicação (NOTA: só há um login válido nesta aplicação)
- O Login deve permitir ao utilizador tentar validar o username e password que inserir. Caso o login esteja correto é redirecionado para a Activity Sobre, caso contrário aparece um Toast com uma mensagem de erro e os campos de username e password são limpos.
- O Sobre deve conter uma mensagem com o nome do desenvolvedor da aplicação.

Exercício 4

- Crie uma aplicação para seleção de produtos a pedir num restaurante. O utilizador deve selecionar os produtos na primeira Activity (através de marcação em CheckBox) e quando terminar clicar no botão “Efetuar pedido”, deve ser redirecionado para um SplashScreen onde aparece a mensagem “A preparar o seu pedido”, após 4 segundos deve ser redirecionado para uma nova Activity onde é apresentado o pedido (ex: 1 café, 1 gelado).

Exercício 5

- Duplique ou acrescente ao exercício anterior os seguintes elementos:
 - A cada item da lista de produtos deve ser colocada uma pequena imagem do mesmo (sugere-se a pesquisa por ImageView)
 - A cada item da lista de produtos deve ser colocado o preço unitário
 - A cada item da lista de produtos deve ser colocado a quantidade que será pedida
 - Na Activity final deve ser apresentado o valor final do pedido