

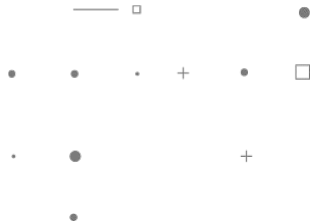


SISTEMAS DE INFORMAÇÃO
DESENVOLVIMENTO ANDROID



MOBILE DEVELOPMENT

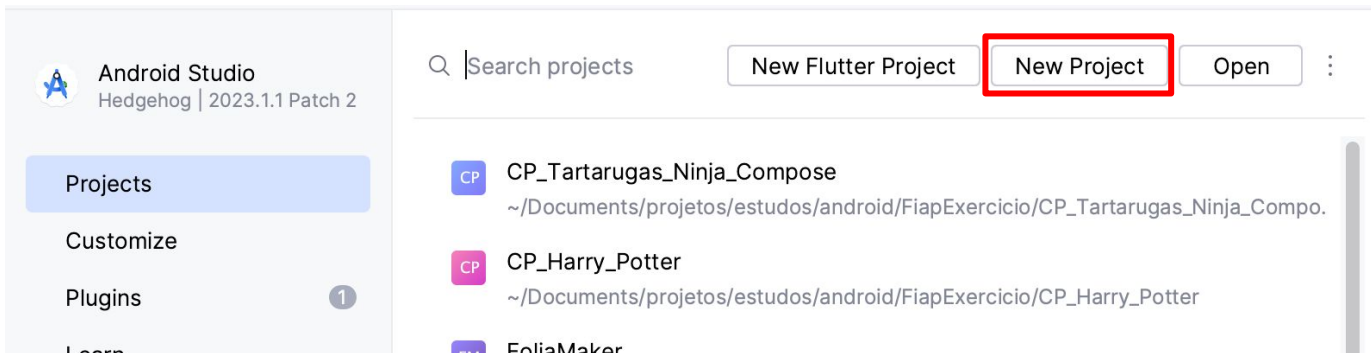




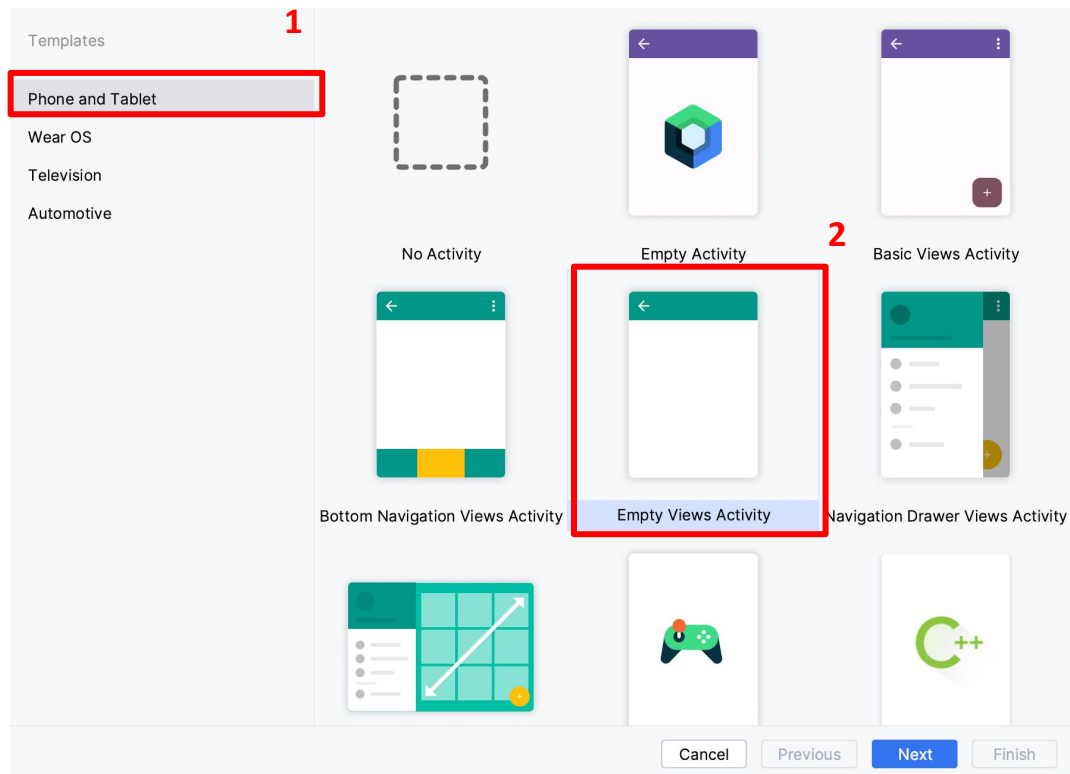
HORA DE COMEÇAR:
QUAL VAI SER O APP DE HOJE?



CRIANDO O PROJETO



CRIANDO O PROJETO



CRIANDO O PROJETO

Empty Views Activity

Creates a new empty activity


Name


Package name

Save location

Language

Minimum SDK

 Your app will run on approximately 81,2% of devices.
[Help me choose](#)

Build configuration language 

Cancel

Previous

Next

Finish

APP: GERADOR DE SENHAS



GERADOR DE SENHA

Começar

Tamanho da senha: 8



Confirmar

Senha aqui



HORA DE COMEÇAR



build.gradle

Ativando o ViewBinding

Para fazer o bind iremos utilizar o **viewBinding**.

Abra o arquivo **build.gradle (app)** e adicione-o:

```
android {  
    namespace = "br.com.heiderlopes.geradordesenha"  
    compileSdk = 35  
  
    viewBinding {  
        enable = true  
    }  
}
```



CRIANDO A PRIMEIRA TELA



LOTTIE

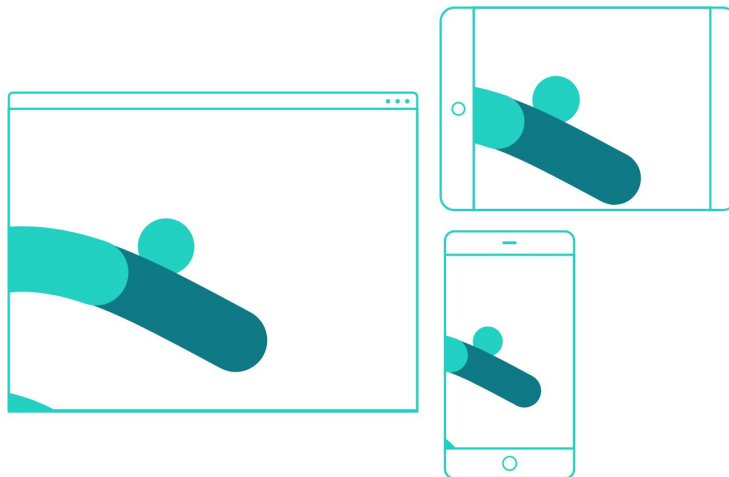
Lottie

Easily add high-quality animation to any native app.

Lottie is an iOS, Android, and React Native library that renders After Effects animations in real time, allowing apps to use animations as easily as they use static images.

Get Started

[Learn more >](#)



Saiba mais: <https://airbnb.design/lottie/>
<https://www.lottiefiles.com>

LOTTIE

Abra o arquivo **build.gradle (app)** e adicione à seguinte dependência:

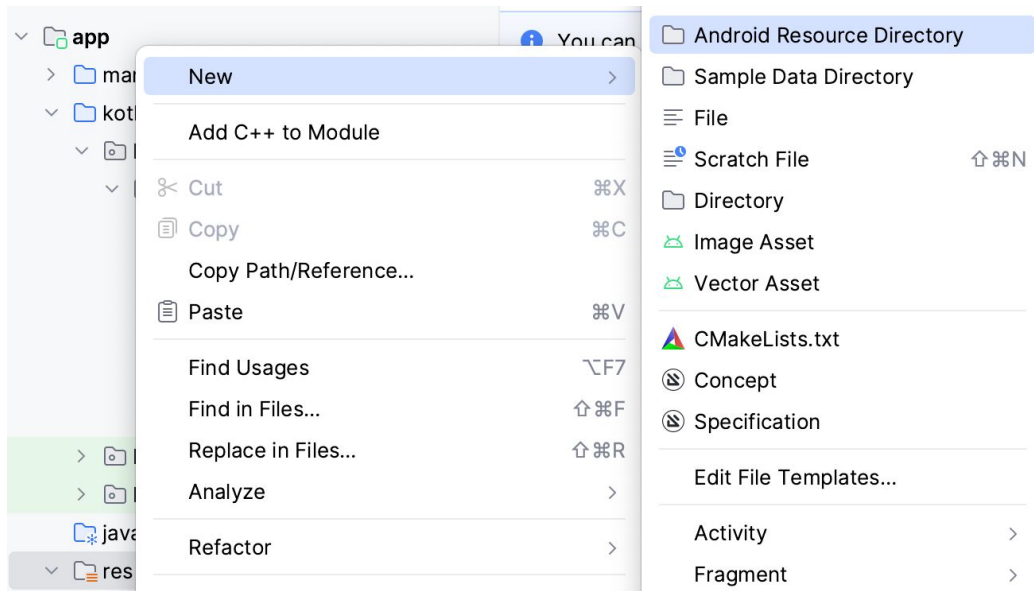
```
implementation ("com.airbnb.android:lottie:6.4.0")
```

E clique em:

Sync Now

LOTTIE

Crie um diretório dentro de **res** chamado **raw** (botão direito sobre **res** → **New** → **Android Resource Directory**)



LOTTIE

Crie um diretório dentro de **res** chamado **raw** (botão direito sobre **res** → **New** → **Android Resource Directory**)

Directory name:

Resource type:

Source set:

Available qualifiers:

- Country Code
- Network Code
- Locale
- Layout Direction
- Smallest Screen Width
- Screen Width
- Screen Height
- Size
- Ratio
- Orientation
- UI Mode

Chosen qualifiers:

Nothing to show

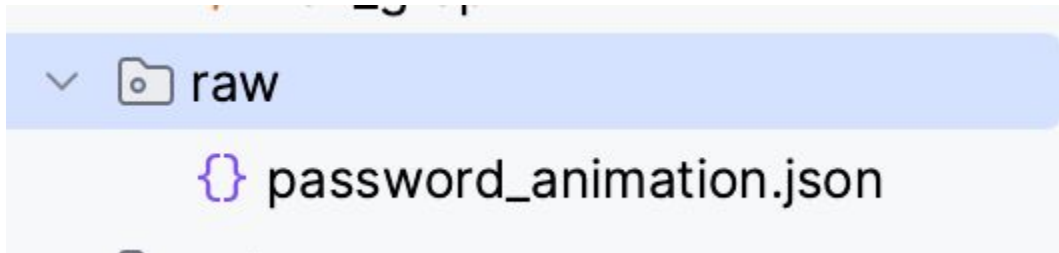
>> <<

Cancel OK

LOTTIE

Baixe as animações desejada no lottie files (<https://lottiefiles.com/>) e adicione na pasta **raw** com o nome:

password_animation.json:



CRIANDO A PRIMEIRA TELA

Abra o arquivo **activity_main.xml** e adicione o seguinte código:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:lottie="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="16dp">

    <com.airbnb.lottie.LottieAnimationView
        android:id="@+id/lottieAnimationView"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:layout_gravity="center"
        lottie:lottie_rawRes="@raw/password_animation"
        lottie:lottie_autoPlay="true"
        lottie:lottie_loop="true"/>
```


CRIANDO A PRIMEIRA TELA

Abra o arquivo **activity_main.xml** e adicione o seguinte código:

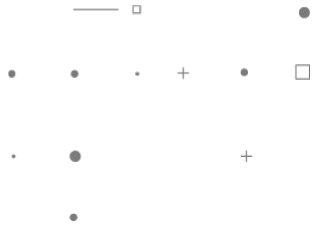
```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="GERADOR DE SENHA"
    android:textAlignment="center"
    android:textSize="20sp" />

<Button
    android:id="@+id/btIniciar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginVertical="16dp"
    android:text="Começar" />
</LinearLayout>
```

CRIANDO A PRIMEIRA TELA

Abra o arquivo **MainActivity.kt** e adicione o código referente ao viewbinding das views:

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        val binding = ActivityMainBinding.inflate(layoutInflater)  
        setContentView(binding.root)  
  
        binding.btIniciar.setOnClickListener {  
  
        }  
    }  
}
```

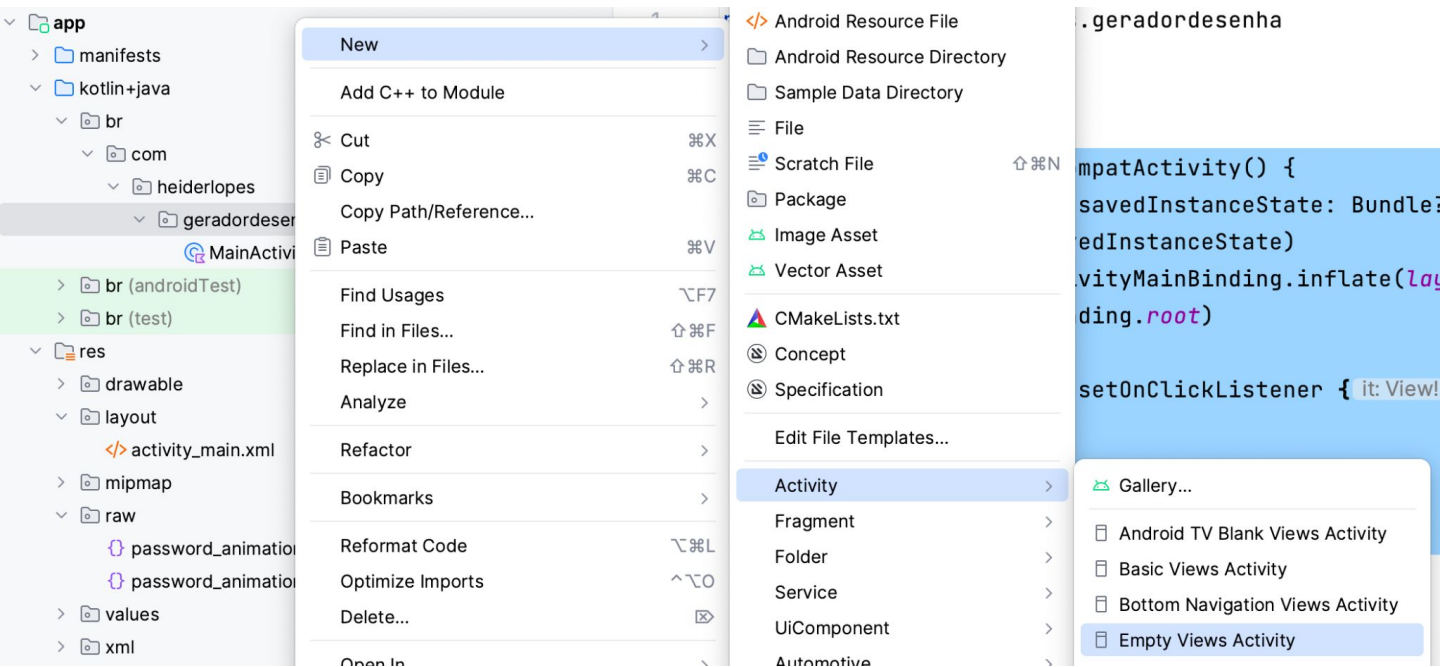


CRIANDO A SEGUNDA TELA



CRIANDO A SEGUNDA TELA

Clique com o botão direito sobre o pacote principal → **New** → **Activity** → **Empty View**



CRIANDO A SEGUNDA TELA

Defina o nome da **Activity** e clique em **Finish**

Empty Views Activity
Creates a new empty activity

Activity Name
SecondActivity

☒ Generate a Layout File

Layout Name
activity_second

☐ Launcher Activity

Package name
br.com.heiderlopes.geradordesenha

Source Language
Kotlin

Cancel Previous Next Finish

CRIANDO A SEGUNDA TELA

Abra o arquivo **activity_second.xml** e adicione o seguinte código:

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:id="@+id/tvTamanho"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Tamanho da senha: 8" />
```

CRIANDO A SEGUNDA TELA

Abra o arquivo **activity_second.xml** e adicione o seguinte código:

```
<SeekBar
    android:layout_marginVertical="32dp"
    android:id="@+id/seekBar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:max="20"
    android:progress="8" />

<Button
    android:id="@+id/btnConfirmar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Confirmar" />
</LinearLayout>
```

CRIANDO A SEGUNDA TELA

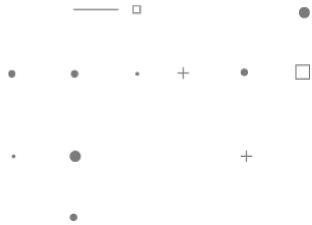
Abra o arquivo **SecondActivity.kt** e adicione o código referente ao viewbinding das views:

```
class SecondActivity : AppCompatActivity() {  
  
    private lateinit var binding: ActivitySecondBinding  
    private var tamanho = 8  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
  
        binding = ActivitySecondBinding.inflate(layoutInflater)  
        setContentView(binding.root)  
    }  
}
```


CRIANDO A SEGUNDA TELA

Abra o arquivo **SecondActivity.kt** e adicione o código referente ao viewbinding das views:

```
binding.seekBar.setOnSeekBarChangeListener( object :  
SeekBar.OnSeekBarChangeListener {  
    override fun onProgressChanged (seekBar: SeekBar?,  
progress: Int, fromUser: Boolean) {  
        tamanho = progress.coerceAtLeast(4)  
        binding.tvTamanho.text = "Tamanho da senha: $tamanho"  
    }  
  
    override fun onStartTrackingTouch (seekBar: SeekBar?) {}  
    override fun onStopTrackingTouch (seekBar: SeekBar?) {}  
})  
  
binding.btnConfirmar.setOnClickListener {  
    }  
}
```

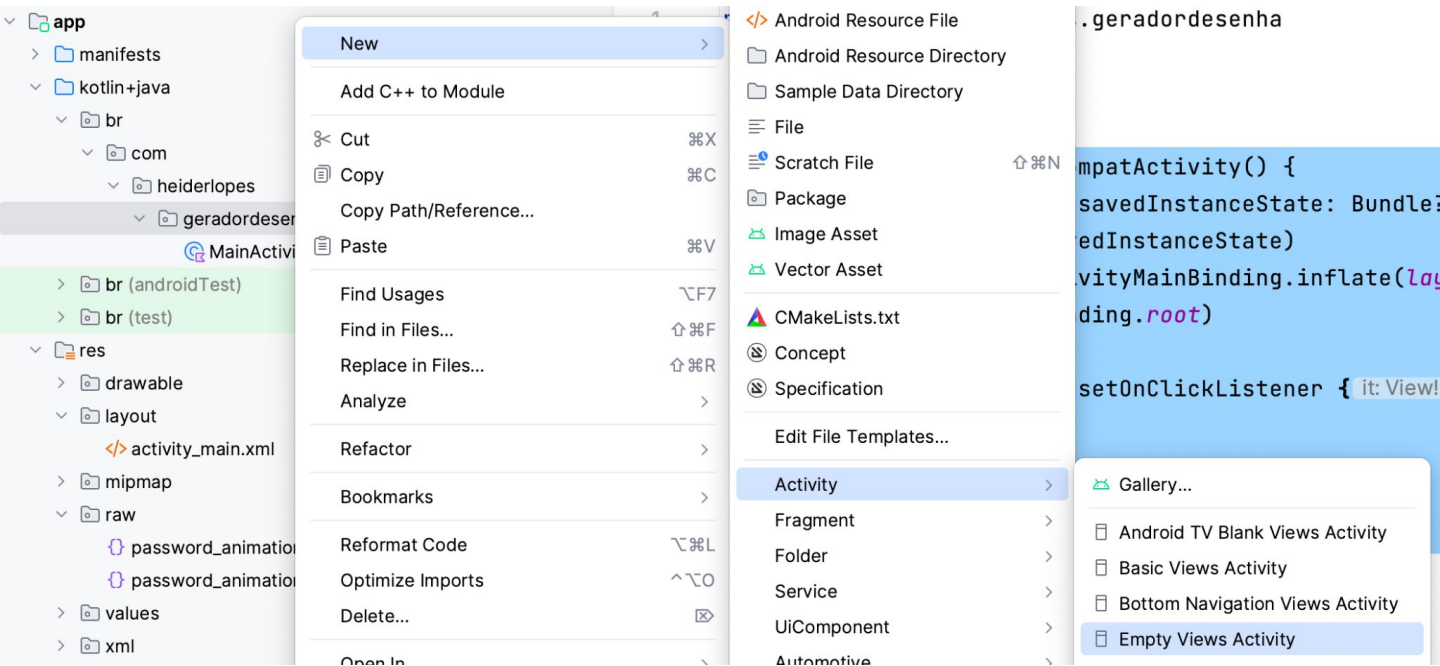


CRIANDO A TERCEIRA TELA



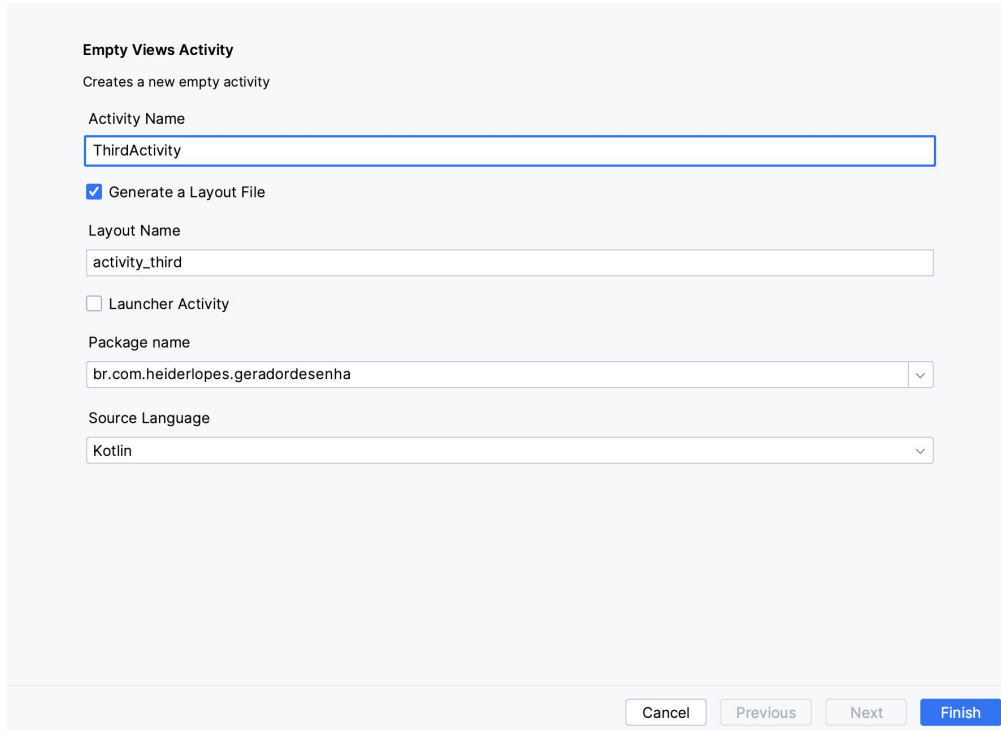
CRIANDO A TERCEIRA TELA

Clique com o botão direito sobre o pacote principal → **New** → **Activity** → **Empty Views**
Activity (Blank)



CRIANDO A TERCEIRA TELA

Defina o nome do **Fragment** e clique em **Finish**



Empty Views Activity
Creates a new empty activity

Activity Name
ThirdActivity

☒ Generate a Layout File

Layout Name
activity_third

☐ Launcher Activity

Package name
br.com.heiderlopes.geradordesenha

Source Language
Kotlin

Cancel Previous Next Finish

CRIANDO A TERCEIRA TELA

Abra o arquivo **activity_third.xml** e adicione o seguinte código:

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:textAlignment="center"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/tvSenha"
        android:text="Senha aqui"
        android:textSize="20sp" />

</LinearLayout>
```

CRIANDO A TERCEIRA TELA

Abra o arquivo **ThirdActivity.kt** e adicione o código referente ao **viewbinding** das views:

```
class ThirdActivity : AppCompatActivity() {  
  
    private lateinit var binding: ActivityThirdBinding  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        binding = ActivityThirdBinding.inflate(layoutInflater)  
        setContentView(binding.root)  
    }  
}
```



CRIANDO A NAVEGAÇÃO



NAVEGANDO DA PRIMEIRA TELA PARA A SEGUNDA

Abra o arquivo **MainActivity.kt** e adicione o seguinte código em negrito ao clique do botão:

```
binding.btIniciar.setOnClickListener {  
    val secondActivityIntent = Intent(this, SecondActivity::class.java)  
    startActivity(secondActivityIntent)  
}
```



PASSANDO PARÂMETROS

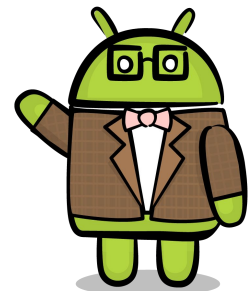
Abra o arquivo **ThirdActivity.kt** e adicione o chave que será responsável pelo parâmetro:

```
class ThirdActivity : AppCompatActivity() {  
  
    private lateinit var binding: ActivityThirdBinding  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        binding = ActivityThirdBinding.inflate(layoutInflater)  
        setContentView(binding.root)  
    }  
  
    companion object {  
        const val EXTRA_TAMANHO_SENHA = "EXTRA_TAMANHO_SENHA"  
    }  
}
```

PASSANDO PARÂMETROS

companion object

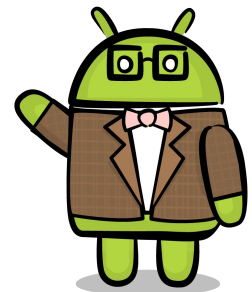
- Permite declarar membros (funções ou variáveis) que pertencem à classe, e não a uma instância dela.
- Funciona como o **static** em Java.
- Assim, você pode acessar **EXTRA_TAMANHO_SENHA** assim:
ThirdActivity.EXTRA_TAMANHO_SENHA, sem precisar criar um objeto da **ThirdActivity**.



PASSANDO PARÂMETROS

const val EXTRA_TAMANHO_SENHA = "EXTRA_TAMANHO_SENHA"

- **const val** define uma constante em tempo de compilação.
- **val** = imutável (como final em Java).
- **const** = o valor é conhecido em tempo de compilação, e só pode ser usado com tipos primitivos e String.
- **EXTRA_TAMANHO_SENHA** é o nome da chave usada para passar o dado via Intent.



PASSANDO PARÂMETROS

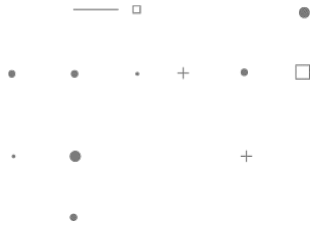
Abra o arquivo **ThirdActivity.kt** e recupere o valor do tamanho da senha e gere uma nova. Fique atento só adicione o código em negrito:

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    binding = ActivityThirdBinding.inflate(layoutInflater)  
    setContentView(binding.root)  
  
    val length = intent.getIntExtra(EXTRA_TAMANHO_SENHA, 0)  
    val senha = gerarSenha(length)  
    binding.tvSenha.text = "Senha gerada:\n$senha"  
}  
  
private fun gerarSenha(tamanho: Int): String {  
    val chars =  
    "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#\$%&*"   
    return (1..tamanho).map { chars.random() }.joinToString("")  
}
```

PASSANDO PARÂMETROS

Abra o arquivo **SecondActivity.kt** e envie o tamanho da senha que será criada no clique do botão confirmar:

```
binding.btnConfirmar.setOnClickListener {  
    val thirdActivityIntent = Intent(this,  
ThirdActivity::class.java).apply {  
        putExtra(ThirdActivity.EXTRA_TAMANHO_SENHA, tamanho)  
    }  
    startActivity(thirdActivityIntent)  
}
```

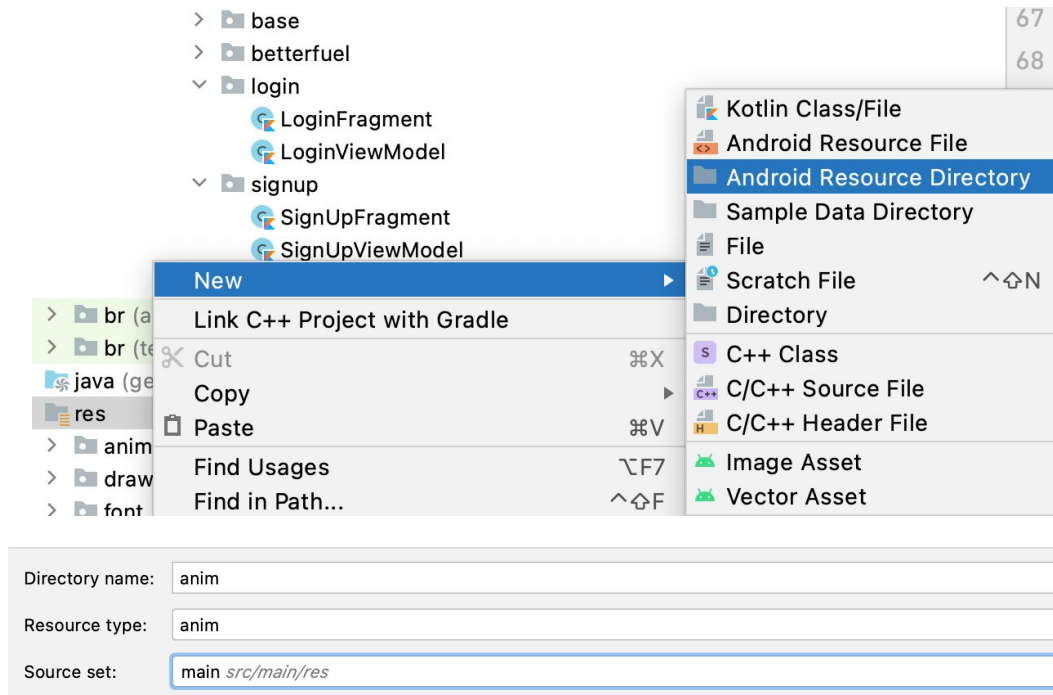


CRIANDO ANIMAÇÕES NAS TRANSIÇÕES DE TELAS



CRIANDO ANIMAÇÕES NAS TRANSIÇÕES DE TELAS

Crie uma pasta chamada **anim** dentro de **res**



CRIANDO ANIMAÇÕES NAS TRANSIÇÕES DE TELAS

Dentro da pasta **anim** crie um arquivo chamado **slide_in_left** e adicione o seguinte código:

```
<?xml version="1.0" encoding="utf-8"?>
<translate xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="@android:integer/config_mediumAnimTime"
    android:fromXDelta="-100%p"
    android:toXDelta="0" />
```

CRIANDO ANIMAÇÕES NAS TRANSIÇÕES DE TELAS

Dentro da pasta **anim** crie um arquivo chamado **slide_in_right** e adicione o seguinte código:

```
<?xml version="1.0" encoding="utf-8" ?>
<translate xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="@android:integer/config_mediumAnimTime"
    android:fromXDelta="100%p"
    android:toXDelta="0" />
```

CRIANDO ANIMAÇÕES NAS TRANSIÇÕES DE TELAS

Dentro da pasta **anim** crie um arquivo chamado **slide_out_left** e adicione o seguinte código:

```
<?xml version="1.0" encoding="utf-8"?>
<translate xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="@android:integer/config_mediumAnimTime"
    android:fromXDelta="0"
    android:toXDelta="-100%p" />
```

CRIANDO ANIMAÇÕES NAS TRANSIÇÕES DE TELAS

Dentro da pasta **anim** crie um arquivo chamado **slide_out_right** e adicione o seguinte código:

```
<?xml version="1.0" encoding="utf-8"?>
<translate
xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="@android:integer/config_mediumAnimTime"
    android:fromXDelta="0"
    android:toXDelta="100%p" />
```



APLICANDO A ANIMAÇÃO NA IDA



CRIANDO ANIMAÇÕES NAS TRANSIÇÕES DE TELAS

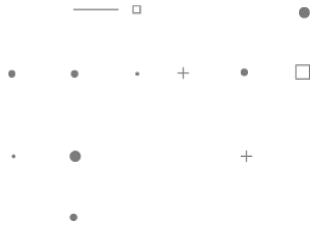
Abra a **MainActivity.kt** e altere o código para passar a animação:

```
binding.btIniciar.setOnClickListener {  
    val secondActivityIntent = Intent(this, SecondActivity::class.java)  
    val options = ActivityOptions.makeCustomAnimation(  
        this,  
        R.anim.slide_in_right,  
        R.anim.slide_out_left  
    )  
  
    startActivity(secondActivityIntent, options.toBundle())  
}
```

CRIANDO ANIMAÇÕES NAS TRANSIÇÕES DE TELAS

Abra a **SecondActivity.kt** e altere o código para passar a animação:

```
binding.btnConfirmar.setOnClickListener {  
    val thirdActivityIntent = Intent(this,  
ThirdActivity::class.java).apply {  
        putExtra(ThirdActivity.EXTRA_TAMANHO_SENHA, tamanho)  
    }  
  
    val options = ActivityOptions.makeCustomAnimation(  
        this,  
        R.anim.slide_in_right,  
        R.anim.slide_out_left  
    )  
  
    startActivity(thirdActivityIntent, options.toBundle())  
}
```



APLICANDO A ANIMAÇÃO NA VOLTA (BACKPRESSED)



CRIANDO ANIMAÇÕES NAS TRANSIÇÕES DE TELAS

Abra a **SecondActivity.kt** e adicione o seguinte código dentro do **onCreate**:

```
onBackPressedDispatcher.addCallback(this) {  
    finish()  
    overridePendingTransition(R.anim.slide_in_left,  
R.anim.slide_out_right)  
}
```

CRIANDO ANIMAÇÕES NAS TRANSIÇÕES DE TELAS

Abra a **ThirdActivity.kt** e adicione o seguinte código dentro do **onCreate**:

```
onBackPressedDispatcher.addCallback(this) {  
    finish()  
    overridePendingTransition(R.anim.slide_in_left,  
R.anim.slide_out_right)  
}
```

OBRIGADO



/heider.lopes



/in/heider-lopes-a06b2869/

FIAP

Copyright © 2019 | Professor (a) Heider Lopes

Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.

FIAP