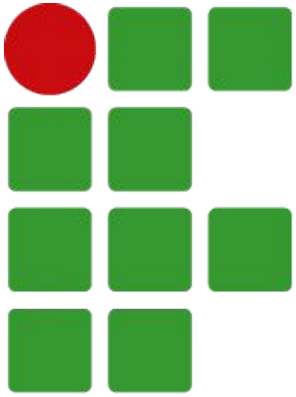


INSTITUTO FEDERAL
Paraná
Campus Paranaguá

Activities

Dispositivos Móveis

Prof. Diego Stiehl

INSTITUTO FEDERAL

Activity

- Representa uma tela da aplicação
  - Parte do app que o usuário vê e interage
- Uma aplicação normalmente é composta por uma ou mais Activities
- No código, temos uma classe Activity
  - Controla os eventos da tela
  - Define quais views serão desenhadas
  - …

INSTITUTO FEDERAL

Activity

- Para termos uma activity, devemos herdar:
  - `android.app.Activity`
    - Ou alguma subclasse dela
- Precisamos sobrescrever o método:
  - `onCreate(bundle)`
- Normalmente teremos um arquivo de layout:
  - res/layout/activity_xxxx.xml
- Deve constar no arquivo AndroidManifest.xml
  - `<activity android:name=".MinhaActivity" />`

INSTITUTO FEDERAL

Analogia com Web

- Se compararmos com um website
  - Um site possui várias páginas
    - Uma aplicação Android apresenta várias Activities
  - O site tem uma homepage (index)
    - Uma aplicação tem uma “main activity”
  - Páginas de sites têm links para páginas (dele e de outros sites)
    - Activities de aplicações podem iniciar outras Activities (de dentro e fora da aplicação)

APIs de Compatibilidade

- O Android é distribuído com bibliotecas de compatibilidade
  - Biblioteca v4
    - Traz retrocompatibilidade de novos recursos do Android até API Level 4 (Android 1.6)
  - Biblioteca v7
    - Traz retrocompatibilidade de novos recursos do Android até API Level 7 (Android 2.3)
- Ambas contém uma série de classes que podem ser estendidas
- Ver build.gradle (app)

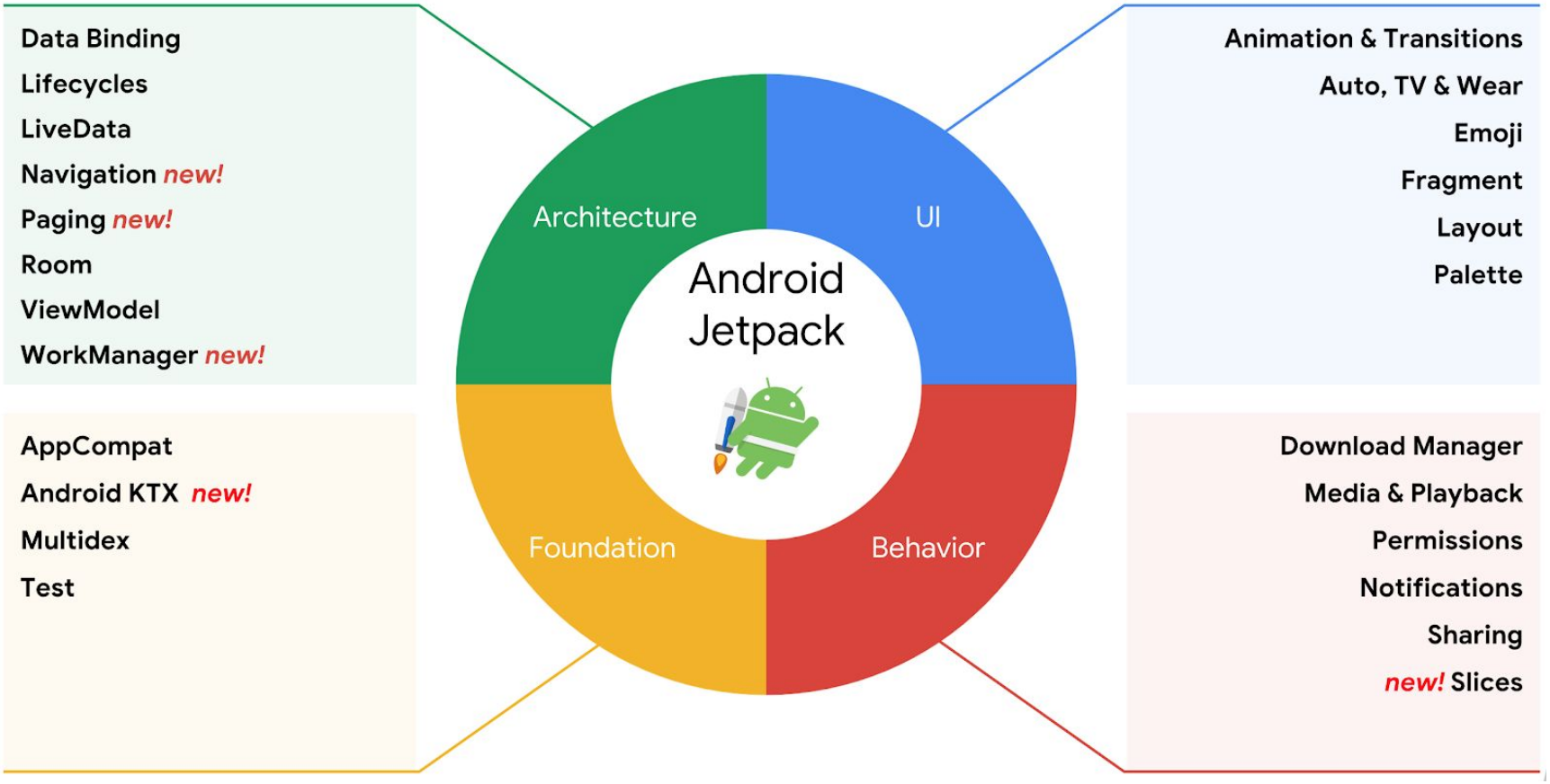
INSTITUTO FEDERAL

Android Jetpack (NOVO)

- Recentemente, as APIs de Compatibilidade foram substituídas pelo Android Jetpack
  - Biblioteca que padroniza, agrupa e cria muitos recursos do Android
- Sugestão: migrar projetos para usar Jetpack
- Pacote:
  - `androidx.*`

INSTITUTO FEDERAL

Android Jetpack



https://medium.com/@victorbrunobotelho/entendendo-o-android-jetpack-parte-1-69f40b95a675

INSTITUTO FEDERAL

Qual Classe Estender?

- Não é indicado estender diretamente a classe Activity
- Devemos usar uma implementação que considere a retrocompatibilidade
  - ~~Biblioteca v7~~ Android Jetpack
- Classe indicada:
  - `androidx.appcompat.app.AppCompatActivity`
    - Leva suporte da ActionBar aos sistemas antigos

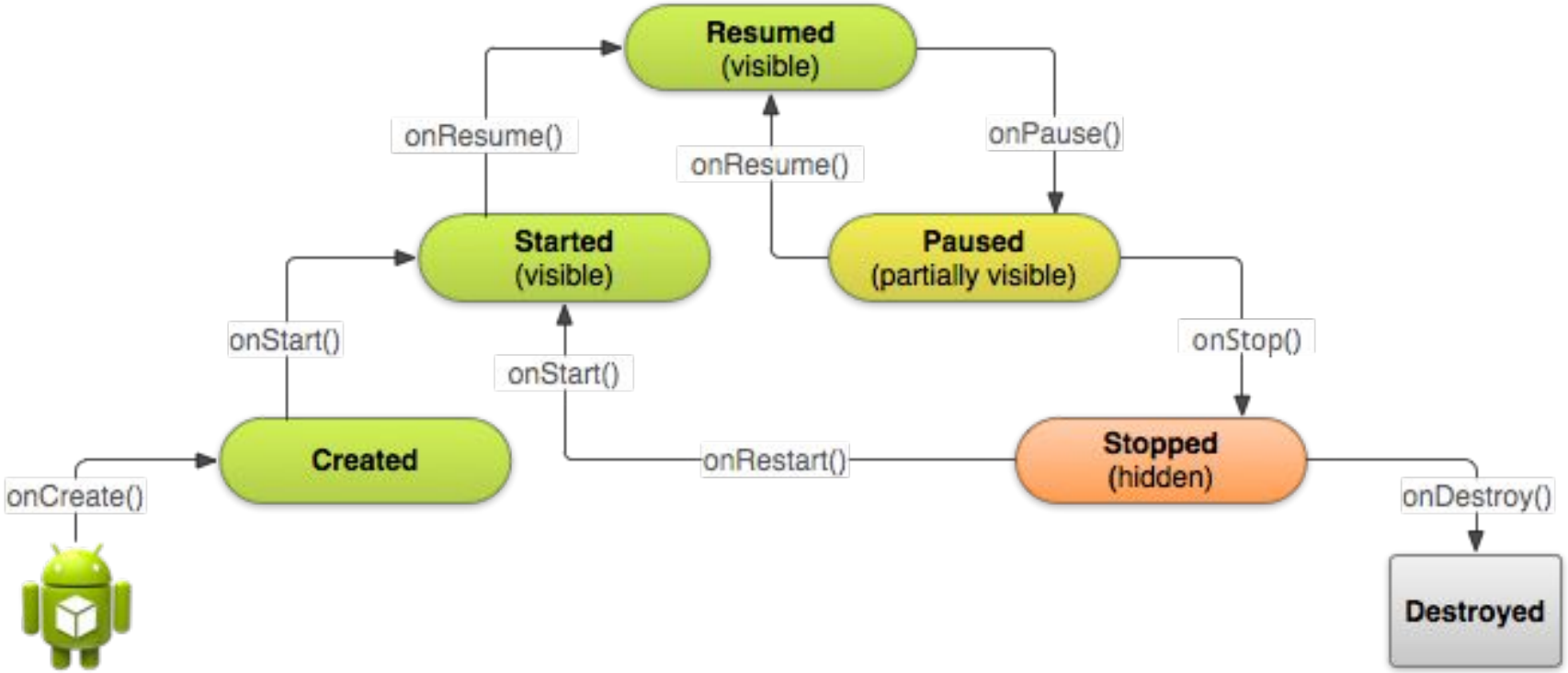
INSTITUTO FEDERAL

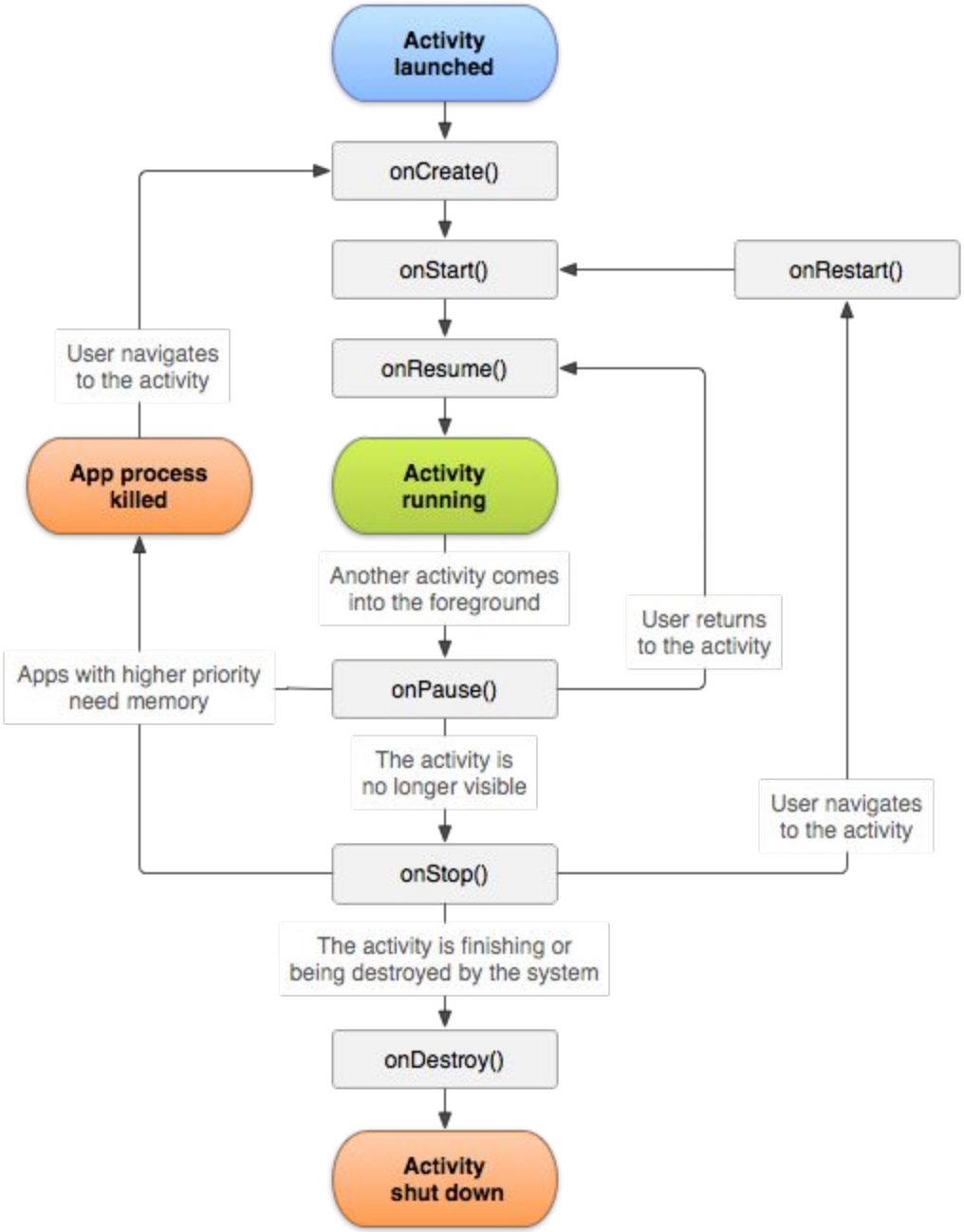
Ciclo de Vida

- O Programador não possui total controle sobre o fluxo e a existência de uma Activity
  - O Activity Manager faz isso
- Uma Activity pode assumir diversos estados
  - Criada, iniciada, resumida, pausada, parada e destruída
  - O programador é avisado de alterações de estado
    - Pode tomar decisões em cada evento

INSTITUTO FEDERAL

Ciclo de Vida




Activity launched
onCreate()
onStart()
onRestart()
onResume()
User navigates to the activity
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

INSTITUTO FEDERAL

Estado Visual

- Os **estados** dizem respeito às propriedades visuais das Activities (foco)
- Uma Activity resumida (resumed)
  - Pode não estar interagindo com o usuário
- Uma Activity parada (stopped)
  - Pode estar realizando algum processamento

INSTITUTO FEDERAL

onCreate()

- Normalmente utiliza-se o onCreate() para fazer as definições iniciais de uma Activity
  - Carregar views (interface gráfica)

```
package br.edu.ifpr.paranagua.novokotlin

import android.support.v7.app.AppCompatActivity
import android.os.Bundle

class MainActivity : AppCompatActivity() {

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

INSTITUTO FEDERAL

Prática - Activity e LogCat

- Implemente (sobrescreva) **TODOS** os métodos do ciclo de vida de uma Activity
  - Use um Log de Info para alertar no Android Monitor sobre a execução de cada método
  - Execute e use a aplicação
  - Analise os resultados impressos
  - Compare com os diagramas de estado apresentados anteriormente

INSTITUTO FEDERAL

Inflando Layout

- Uma Activity normalmente vai inflar um layout XML definido para ela
  - No `onCreate()`
- Esta é a função do método:
  - `setContentView(R.layout.activity_main)`
- Setar a view de conteúdo padrão da Activity

INSTITUTO FEDERAL

Elementos

- Se o layout XML definir elementos internos:
  - Podemos inflar estes como objetos Kotlin
- Os elementos precisam ter um **id** válido

```
<TextView
    android:id="@+id/txtHello"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!" />
```

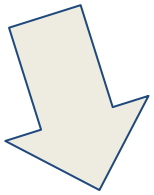
```
val txtHello = findViewById<TextView>(R.id.txtHello)
txtHello.text = "Mudei o texto!"
```

Evoluindo o OnClickListener

```
class MainActivity : AppCompatActivity(), View.OnClickListener {
    override fun onClick(v: View) {
        if (v == btOK) {
            btOK.text = "Clicou"
        }
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
//          ...

        val btOK = findViewById<Button>(R.id.btOK)
        btOK.setOnClickListener(this)
    }
}
```



```
btOK.setOnClickListener { btOK.text = "Clicou" }
```

INSTITUTO FEDERAL

Iniciando Outra Activity

- Precisamos dizer que temos a intenção de iniciar determinada Activity
  - Usamos uma Intent (intenção)
- A outra Activity terá seu próprio ciclo de vida

```
val intent = Intent(this, SomeActivity::class.java)
startActivity(intent)
```

INSTITUTO FEDERAL

Parâmetros

```
val bundle = Bundle()
bundle.putString("nome", "Diego")

val intent = Intent(this, SomeActivity::class.java)
intent.putExtras(bundle)

startActivity(intent)
```

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

    val nome = intent.extras.getString("nome")
    Toast.makeText(this, nome, Toast.LENGTH_SHORT).show()
}
```

INSTITUTO FEDERAL

Prática - Ciclos de Vida

- Gere Log para as duas Activities (ciclo de vida)
  - Identifique cada uma nas mensagens
- Use a aplicação
  - Navegue entre elas e retorne
- Analise o resultado no LogCat

INSTITUTO FEDERAL

Prática - Login

- Aplicação com Login
  - Reaproveitar XML da atividade de Login
    - Vai ser a tela inicial
  - Se logar com sucesso (verificação estática)
    - Vai para uma segunda tela e mostra os dados do usuário em campos de texto
  - Senão
    - Fica na mesma tela e mostra um Toast de “Login ou Senha inválidos”

INSTITUTO FEDERAL

O Problema de Rotacionar

- O método onCreate() é chamado apenas uma vez na "vida" da Activity
  - Porém, quando rotacionamos o dispositivo, toda a Activity é recriada do zero
    - Necessidade de recarregamento de layout
      - Novas views
      - Diferente orientação
- Perdermos o estado da Activity
  - Variáveis, objetos, dentre outros devem ser salvos

INSTITUTO FEDERAL

Salvando o Estado

```
class SomeActivity : AppCompatActivity() {
    private lateinit var cidade: String

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

        if (savedInstanceState != null)
            cidade = savedInstanceState.getString("cidade")
        else
            cidade = "Paranaguá"
    }

    override fun onSaveInstanceState(outState: Bundle) {
        super.onSaveInstanceState(outState)
        outState.putString("cidade", cidade)
    }
}
```