

Programação para Dispositivos Móveis

Actividades e *Intents*

2º Ano 1º Semestre
2020-2021



Sumário

- O que são atividades ?
- Ciclo de Vida de uma Atividade
 - Funções (Métodos)
 - Persistência de Estados
 - Sistema de Logs – LogCat
 - Mensagens no ecrã – Toasts
 - Abrir uma Atividade
 - *Intents* Explícitos e Implícitos

Componentes de uma aplicação Android

- As aplicações Android são compostas pelos seguintes componentes:
 - activities
 - intents
 - services
 - (content providers)
 - (broadcast receivers)

O que são Atividades ?

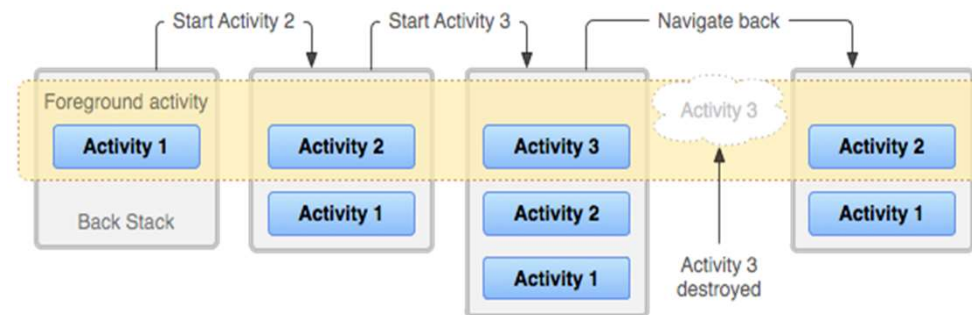
- Componente de uma aplicação Android que fornece ao utilizador um ecrã. Este ecrã representa uma atividade, como por exemplo, preencher um formulário.
- Ligação da parte gráfica (layout do ecrã) com a parte lógica (ações desencadeadas pelo utilizador ao interagir com a parte gráfica)
- Normalmente está associada a uma interface gráfica em que os utilizadores podem interagir de forma a executarem uma determinada ação:
 - Tirar fotos
 - Enviar mensagens ...

O que são Atividades ?

- As atividades podem ser reutilizáveis, mesmo em aplicações diferentes.
 - Por exemplo, podemos ter uma aplicação que num dado ponto é necessário enviar um email. Não será necessário desenvolver essa funcionalidade, pois ela já existe. Apenas temos que chamar a atividade da aplicação de e-mail.
- Mesmo que as atividades sejam de aplicações diferentes, o Sistema Android garante que uma experiência contínua ao utilizador mantendo as atividades relacionados na mesma tarefa. Assim, a tarefa é um conjunto de atividades relacionadas para atingirem um dado objetivo.

O que são Atividades ?

- O sistema Android utiliza uma fila de prioridades para a gestão das atividades em execução num dispositivo.



Fonte: <https://developer.android.com/guide/components/activities/tasks-and-back-stack>

- Quando o utilizador pressiona o botão “Back”, a atividade atual é destruída, não mantendo o estado. E a última atividade a entrar, é a primeira a sair (LIFO – *Last in, First out*)

O que são Atividades ?

- Cada “activity” está listada no ficheiro “AndroidManifest.xml”
 - `<activity android:name=“xxx” android:label=“xxx”>`
- Estende a classe Activity (ou outras semelhante - como por exemplo ActionBarActivity)

O que são Atividades ?

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="pt.xpto.xpto"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="15" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".HelloWorldActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

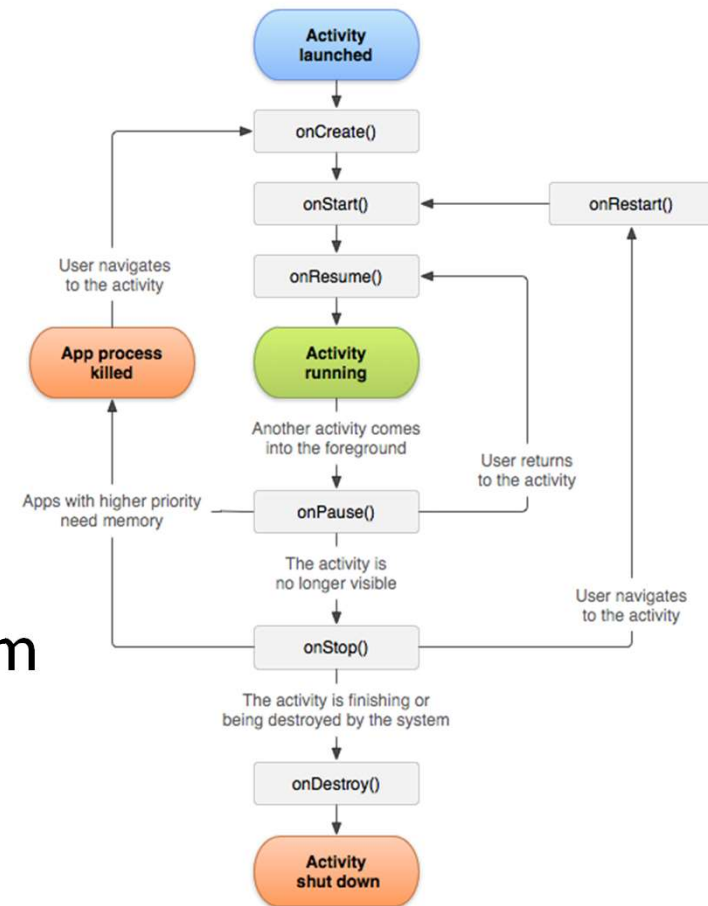
</manifest>
```

AndroidManifest.xml



Ciclo de Vida de uma Atividade

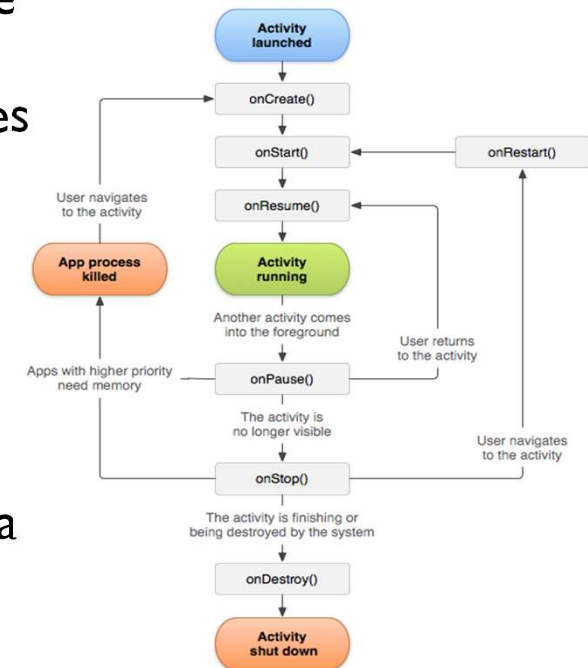
- Uma “Activity” passa por vários estados durante o seu ciclo de vida. A Figura seguinte mostra as funções chamadas na “Activity” quando existem alterações de estado.
- Ao desenvolver uma aplicação é importante o programador ter em conta estes estados e reagir apropriadamente a cada um, caso necessário.



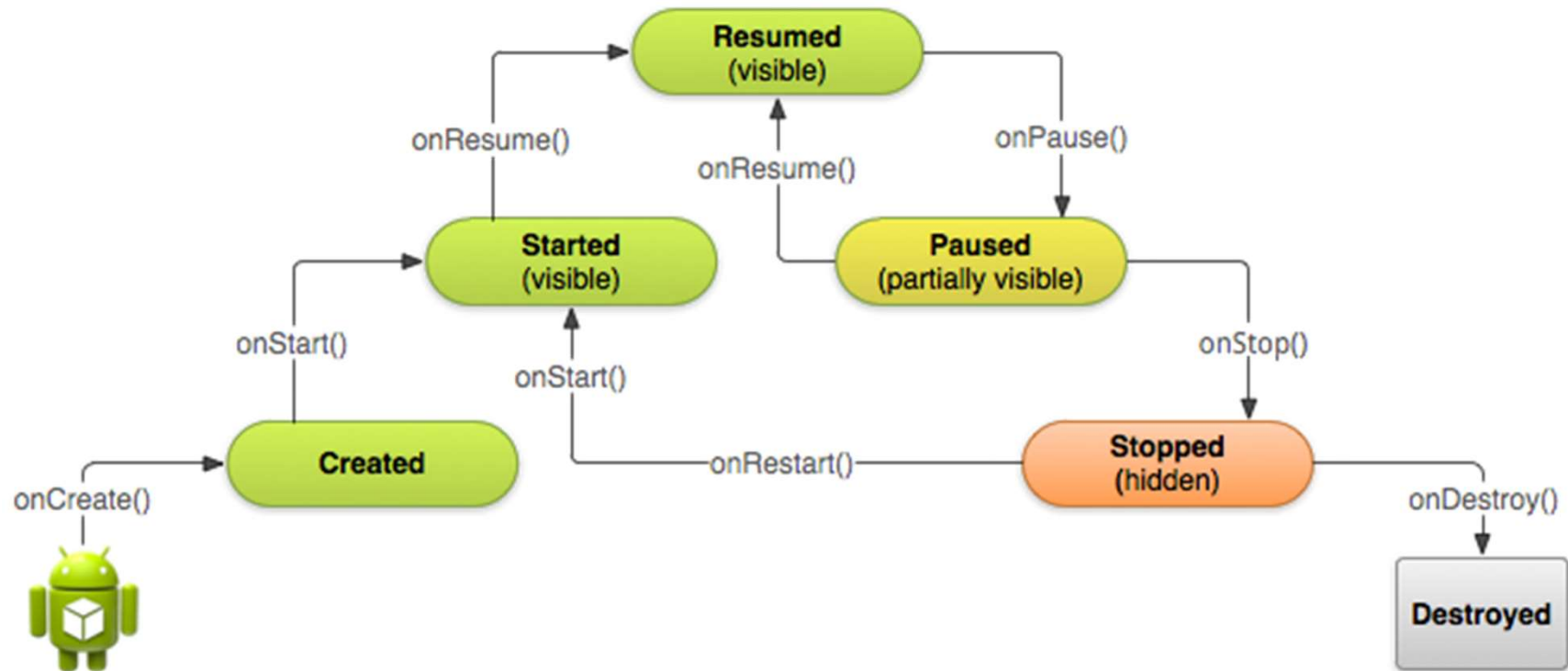
Fonte: <https://developer.android.com/guide/components/activities?hl=pt-br>

Ciclo de Vida de uma Atividade

- **onCreate:** chamado quando a atividade é criada. Local ideal para executar grande parte das tarefas de inicialização.
- **onRestart:** chamado quando a atividade está prestes a reiniciar depois de ter sido parado pelo sistema.
- **onStart:** indica à atividade que está prestes a ser visível para o utilizador.
- **onResume:** quando vai dar início à interação com o utilizador.
- **onPause:** deve-se armazenar dados persistentes exigidos pela atividade e parar tarefas intensivas para o CPU (animação).
- **onStop:** deixa de estar visível para o utilizador.
- **onDestroy:** prestes a ser destruída. Voluntária ou pelo sistema por motivos de libertar memória ou alteração de configuração (mudança de orientação do dispositivo).

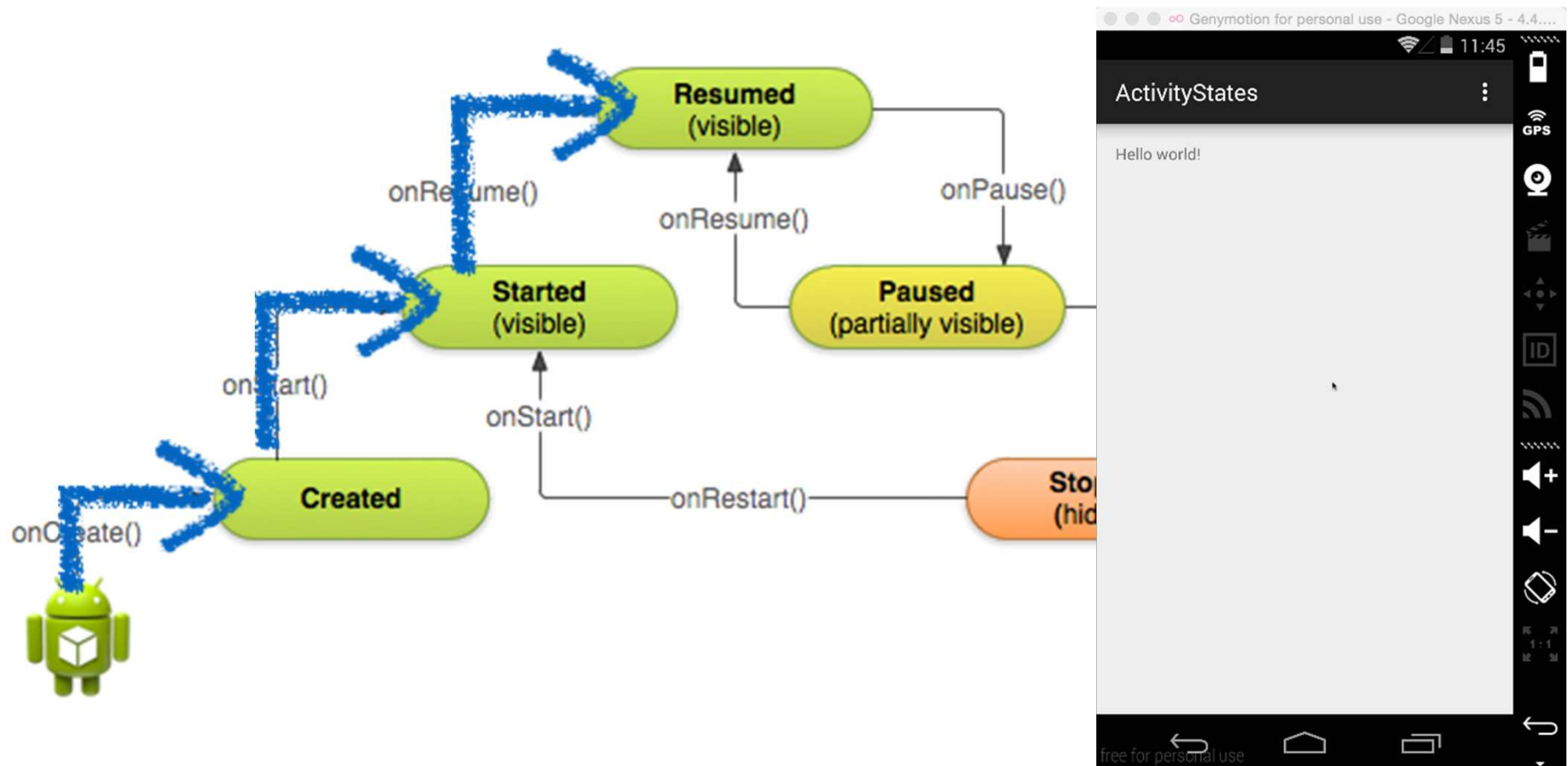


Ciclo de Vida de uma Atividade



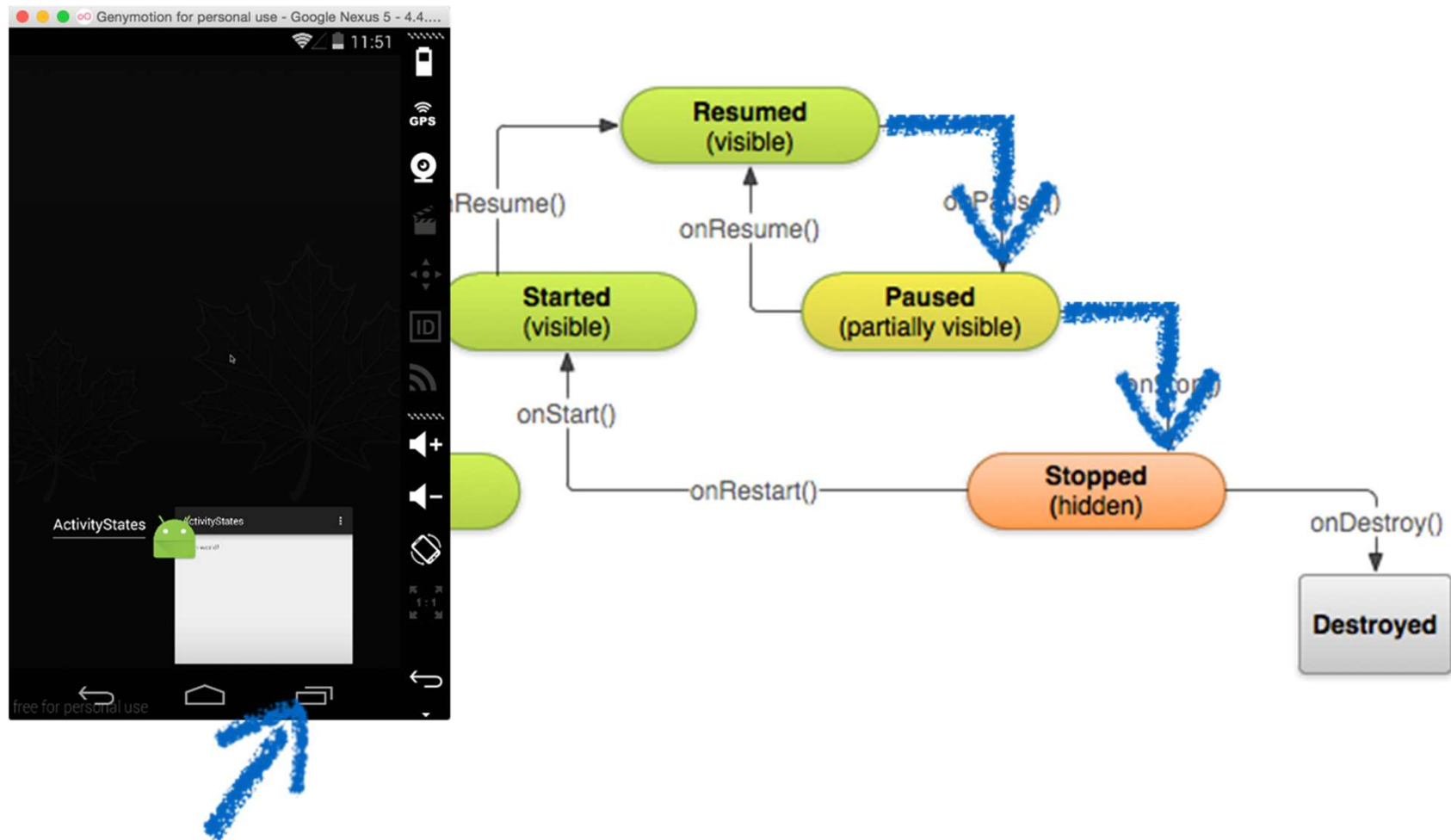
Ciclo de Vida de uma Atividade

- Exemplo I:



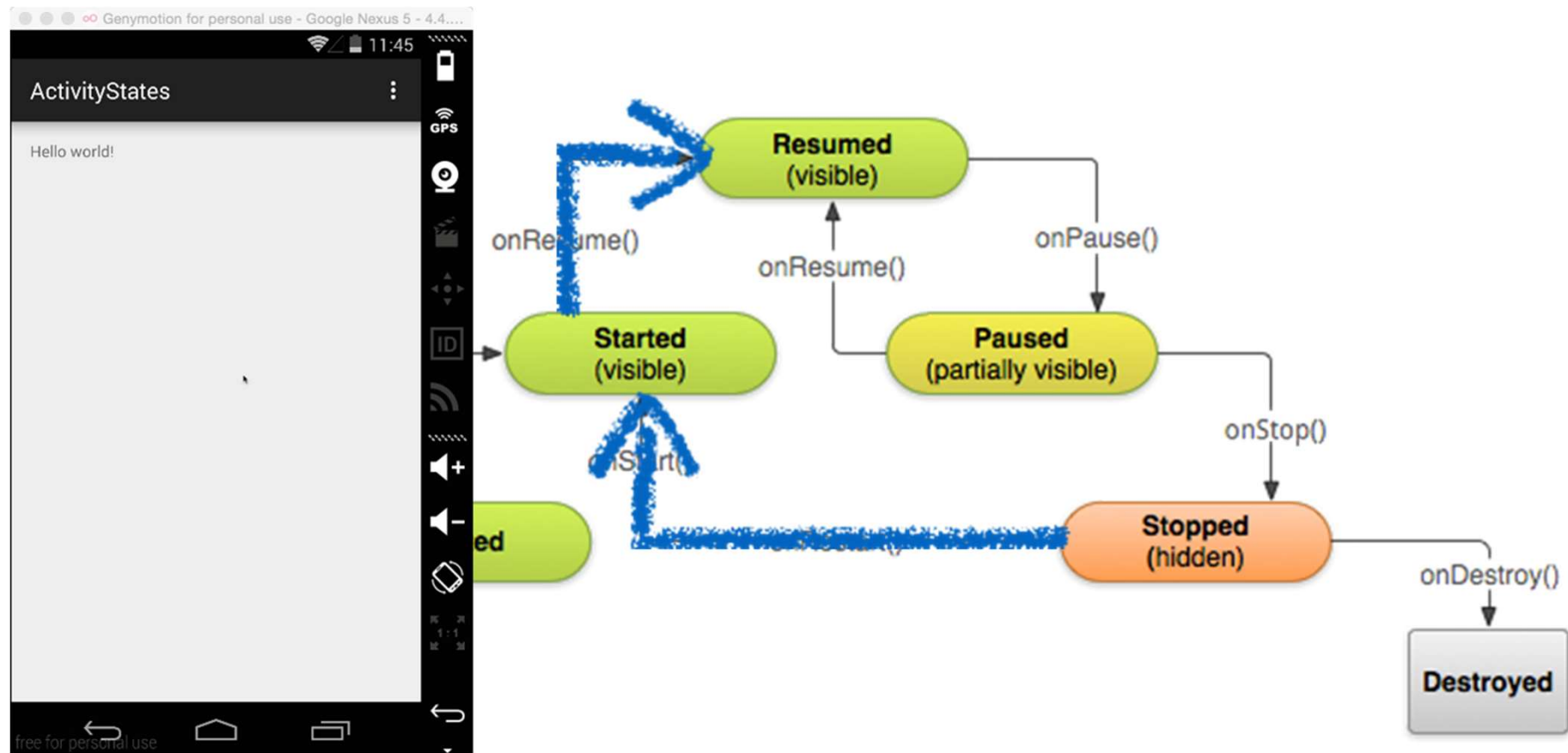
Ciclo de Vida de uma Atividade

Exemplo 2:



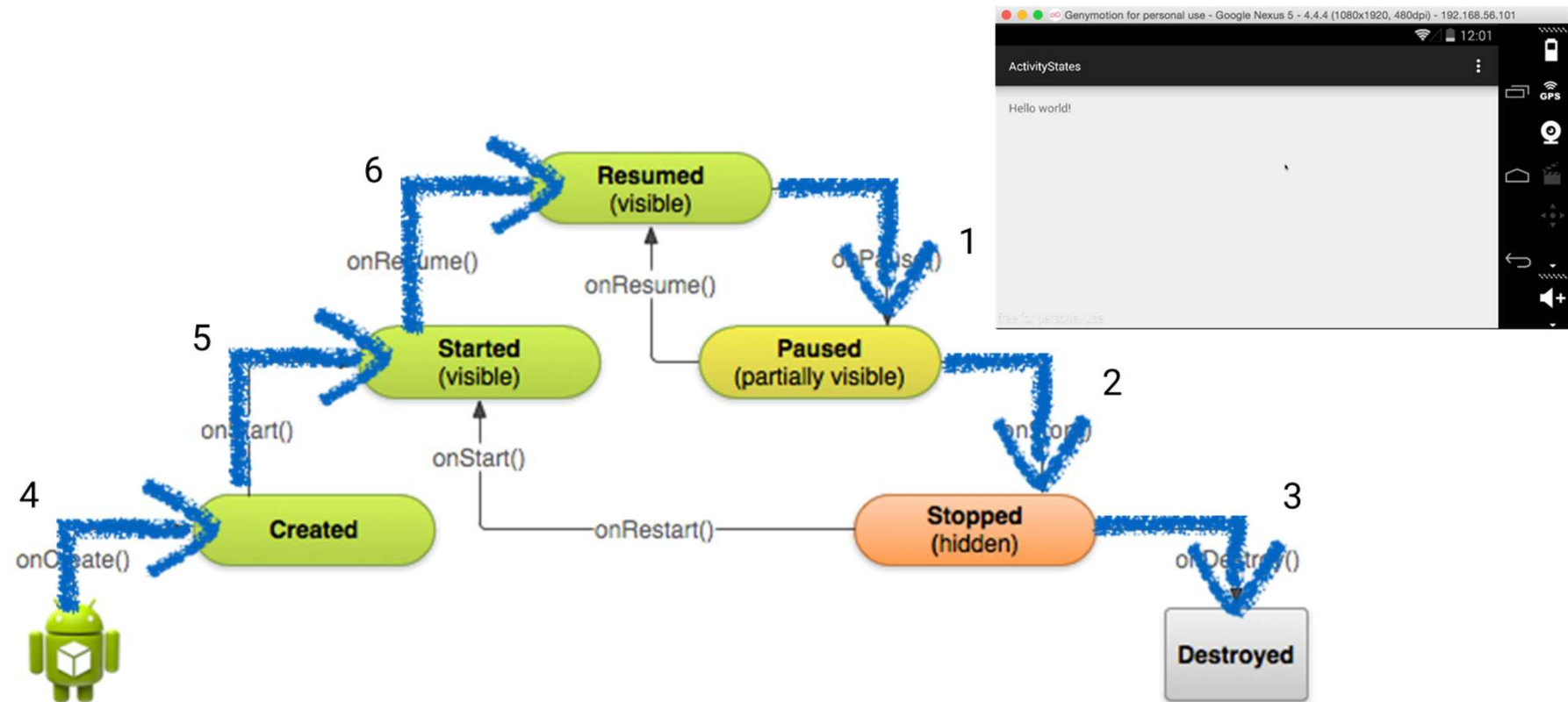
Ciclo de Vida de uma Atividade

Exemplo 3:



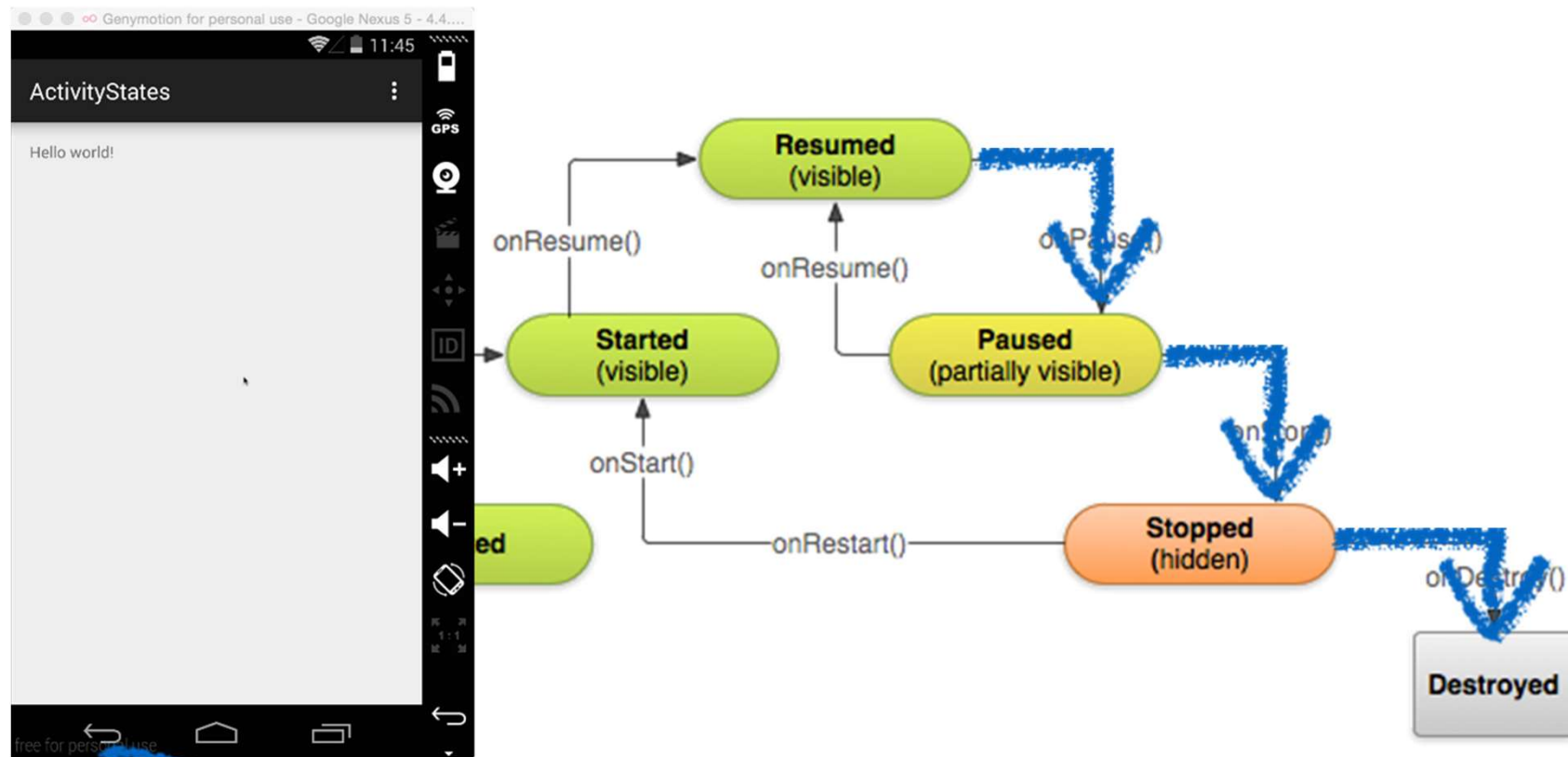
Ciclo de Vida de uma Atividade

Exemplo 4:



Ciclo de Vida de uma Atividade

Exemplo 5:

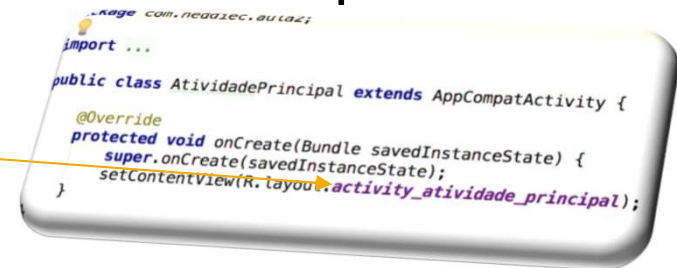


Ciclo de Vida de uma Atividade

- Persistência de Estado

- Quando a atividade é interrompida ou destruída é possível recuperar o estado da atividade:

- Armazenar valores (**Bundle**)
 - Criar uma classe para armazenamento



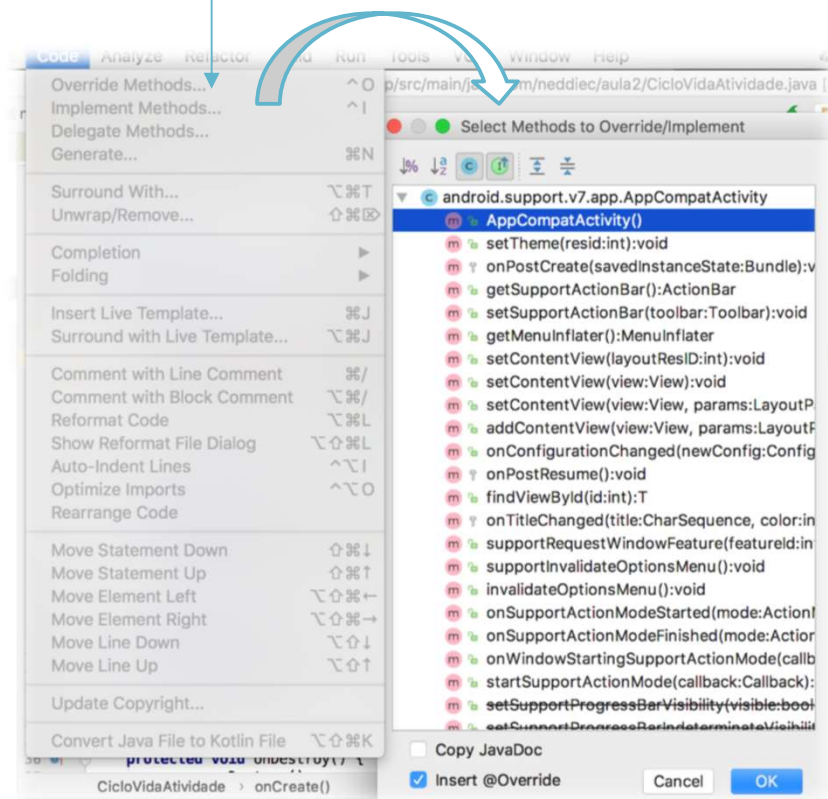
```
package com.neqazec.aulaz;  
  
import ...  
  
public class AtividadePrincipal extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_atividade_principal);  
    }  
}
```

A yellow arrow points from the word "Bundle" in the list to the `Bundle savedInstanceState` parameter in the `onCreate` method of the code snippet.

- Uma atividade tem dois métodos para o armazenamento e recuperação do estado:
 - **onSaveInstanceState** – chamado quando está a ser destruída.
 - **onRestoreInstanceState** – chamado após o **onCreate**.
 - O uso destes métodos permite guardar dados temporariamente de forma a ressalvar situações em que a atividade é interrompida ou destruída. (Exemplo: após preencher alguns dados de um longo formulário, o utilizador muda a orientação do dispositivo móvel)

Aplicação "Ciclo de Vida"

- Incluir os métodos do ciclo de vida através do menu principal **Code > Override Methods ...**
 - **onCreate** (já implementado)
 - **onStart**
 - **onRestart**
 - **onResume**
 - **onPause**
 - **onStop**
 - **onDestroy**
 - **onSaveInstanceState**
 - **onRestoreInstanceState**



Aplicação "Ciclo de Vida"

- Para assinalar a passagem pelo estado devemos utilizar um sistema de *logs* ou através de mensagens visualizadas no ecrã (existem estados em que estas mensagens não são apresentadas !!! Porquê ?).
- Classe **Log** (através do pacote `android.util`) – registo de eventos baseado em níveis. Para a visualização destes eventos é utilizado a ferramenta **LogCat** do Android Studio.

```
private static final String TAG = "CicloDeVida";  
...  
Log.d(TAG, "onCreate");
```

- Toasts (através do pacote `android.widget.Toast`) – um pequeno *popup* que é visualizado durante poucos segundos no fundo do ecrã.

```
public class Uteis {  
    public static void MessageToast(String s, Application a) {  
        Toast.makeText(a.getApplicationContext(), s, Toast.LENGTH_LONG).show();  
    }  
}
```

Aplicação "Ciclo de Vida"

- Quais os métodos invocados nos seguintes cenários ?
 - Executar a aplicação pela primeira vez
 - Clicar no botão **Home**
 - Mudar a orientação do ecrã
 - Sair da aplicação, clicando no botão **Back**
 - Voltar à aplicação

Aplicação "Ciclo de Vida"

- Quais os métodos invocados nos seguintes cenários ?

- Executar a aplicação pela primeira vez

```
com.neddiiec.aula2 D/CicloDeVida: onCreate  
com.neddiiec.aula2 D/CicloDeVida: onStart  
com.neddiiec.aula2 D/CicloDeVida: onResume
```

- Clicar no botão **Home**

```
com.neddiiec.aula2 D/CicloDeVida: onPause  
com.neddiiec.aula2 D/CicloDeVida: onSaveInstanceState  
com.neddiiec.aula2 D/CicloDeVida: onStop
```

- Mudar a orientação do ecrã

- Sair da aplicação, clicando no botão **Back**

```
/com.neddiiec.aula2 D/CicloDeVida: onPause  
/com.neddiiec.aula2 D/CicloDeVida: onSaveInstanceState  
/com.neddiiec.aula2 D/CicloDeVida: onStop  
/com.neddiiec.aula2 D/CicloDeVida: onDestroy  
/com.neddiiec.aula2 D/CicloDeVida: onCreate  
/com.neddiiec.aula2 D/CicloDeVida: onStart  
/com.neddiiec.aula2 D/CicloDeVida: onRestoreInstanceState  
/com.neddiiec.aula2 D/CicloDeVida: onResume
```

- Voltar à aplicação

```
com.neddiiec.aula2 D/CicloDeVida: onPause  
com.neddiiec.aula2 D/CicloDeVida: onStop  
com.neddiiec.aula2 D/CicloDeVida: onDestroy
```

```
/com.neddiiec.aula2 D/CicloDeVida: onRestart  
/com.neddiiec.aula2 D/CicloDeVida: onStart  
/com.neddiiec.aula2 D/CicloDeVida: onResume
```

Ver: <https://developer.android.com/studio/debug/am-logcat>



Comunicação entre Atividades

- O mecanismo de comunicação entre atividades é efetuado pelas **Intents** (intenções)
- **Intents**
 - Mensagem assíncrona que permite o acesso a funcionalidades de outros componentes do Android (atividades, serviços ou recetores de *broadcast*)

- **Intents** explícitas

- Indica-se o nome da atividade (classe Java associada).

```
Intent ActCicloVida = new Intent(this, CicloVidaAtividade.class);
startActivity(ActCicloVida);
```

- **Intents** implícitas

- Indica-se a ação a realizar. O Android seleciona a atividade adequada para a dada ação, mesmo sendo de uma aplicação diferente.

```
Intent i = new Intent( Intent.ACTION_VIEW, Uri.parse("http://www.estgv.ipv.pt"));
startActivity(i);
```



Comunicação entre Atividades

- **Intents** explícitas

- Indica-se o nome da atividade (classe Java associada). Estas classes são conhecidas e controladas pelo programador.

```
Intent ActCicloVida = new Intent(this, CicloVidaAtividade.class);
startActivity(ActCicloVida);
```

- Podemos enviar dados (pares chave-valor) para a atividade de destino:

```
Intent i = new Intent(this, AtividadeComValores.class);
i.putExtra("nome", "Marco Bernardo");
i.putExtra("email", "mbernardo@estgv.ipv.pt");
startActivity(i);
```

- Recebendo na atividade de destino:

```
Intent i = getIntent();
Bundle extras = i.getExtras();
if(extras != null){
    tvname.setText(extras.getString("nome"));
    tvemail.setText(extras.getString("email"));
}
```

Comunicação entre Atividades

- **Intents** explícitas
 - Existe a possibilidade também o retorno de dados. Para isso é necessário implementar o método **finish()** na atividade destino e o método **onActivityResult()** na AtividadePrincipal.

```
@Override
public void finish() {
    TextView tvname = (TextView) findViewById(R.id.tv_nome);
    TextView tvemail = (TextView) findViewById(R.id.tv_email);
    Intent data = new Intent();
    data.putExtra("resposta", "Olá " + tvname.getText() + " - " +
    tvemail.getText());
    setResult(RESULT_OK, data);
    super.finish();
}
```

AtividadeComValoresERetorno

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    String AResposta = "";
    if(resultCode == RESULT_OK && requestCode == REQUEST_CODE){
        if(data.hasExtra("resposta")){
            AResposta = data.getExtras().getString("resposta");
        } else AResposta = "Erro no Resultado (2)";
    }else AResposta = "Erro no Resultado (1)";
    Uteis.MessageToast(AResposta, this.getApplication());
}
```

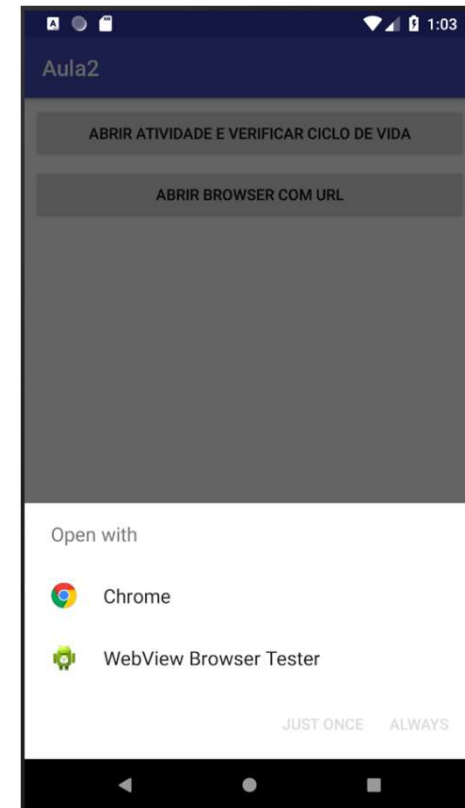
AtividadePrincipal



Comunicação entre Atividades

- **Intents** implícitas
 - Indica-se a ação a realizar e/ou o tipo de dados a serem manipulados pela atividade. O Android seleciona a atividade adequada para a dada ação, mesmo sendo de uma aplicação diferente.
 - Por exemplo, um tipo de ação `ACTION_VIEW` com o URL (objeto `URI`), o Android vai executar uma atividade com a capacidade de navegação web. Neste caso, e como existem várias aplicações que permitem a navegação web, irá perguntar com qual aplicativo deseja executar o URL.

```
Intent i = new Intent(  
    Intent.ACTION_VIEW,  
    Uri.parse("http://www.estgv.ipv.pt")  
);  
startActivity(i);
```



Comunicação entre Atividades

- **Intents** implícitas
 - Mais um exemplo:
 - Enviar uma mensagem eletrónica de um determinado email, com o assunto e texto já predefinido.

```
Intent intent = new Intent(Intent.ACTION_SEND);
intent.setType("text/html");
intent.putExtra(Intent.EXTRA_EMAIL, "mbernardo@estgv.ipv.pt");
intent.putExtra(Intent.EXTRA_SUBJECT, "Aulas de PDM - CTeSP DWDM");
intent.putExtra(Intent.EXTRA_TEXT, "Agendar aula devido a ... (motivo) ...");

startActivity(Intent.createChooser(intent, "Send Email"));
```

Adaptado de:
Nuno Costa

