



POLITÉCNICO  
DE LEIRIA  
ESCOLA SUPERIOR  
DE TECNOLOGIA  
E GESTÃO

**CTeSP de Programação de Sistemas de Informação**

**Acesso Móvel a Sistemas de Informação**

# Activities e Intents na Criação de Aplicações Android

Sónia Luz, [sonia.luz@ipleiria.pt](mailto:sonia.luz@ipleiria.pt)

David Safadinho, [david.safadinho@ipleiria.pt](mailto:david.safadinho@ipleiria.pt)

Departamento de Engenharia Informática

Escola Superior de Tecnologia e Gestão

Instituto Politécnico de Leiria

1º Semestre - 2021/2022

# Aplicação Android

- Normalmente, uma aplicação é composta por um conjunto de atividades que estão interligadas
- Pelo menos uma das atividades é especificada (*AndroidManifest.xml*) como a atividade principal (*main*)
  - é iniciada pelo *launcher* da plataforma;
  - apresentada ao utilizador quando este pressiona o ícone da aplicação;
  - corresponde ao ponto de entrada da aplicação.

# Atividade - Definição

- É um componente de uma aplicação que fornece um ecrã com o qual os utilizadores podem interagir:
  - Ex: realizar uma chamada, tirar uma fotografia, enviar um email, ver um mapa, etc.;
- Cada atividade corresponde, tipicamente, a um par Código Java + XML
  - tem acesso a uma janela:
    - que ocupa todo o ecrã do dispositivo;
    - e que deve conter os itens da GUI.

# Atividade – Funcionamento

- Cada atividade **pode iniciar outra atividade**
  - de modo a executar ações diferentes;
  - disponibilizar todas as funcionalidades pretendidas na App;
- Cada nova atividade iniciada:
  - é apresentada ao utilizador e internamente é colocada (*push*) na pilha de atividades (*“back stack”*);
  - sobrepondo-se à atividade que se encontrava visível imediatamente antes e que foi a responsável pelo seu início;
  - esta pilha assume o comportamento LIFO típico
    - quando a atividade atual termina ou é terminada pelo utilizador, é retirada (*pop*) da pilha e a atividade imediatamente antes torna-se visível.



# Atividade – Funcionamento

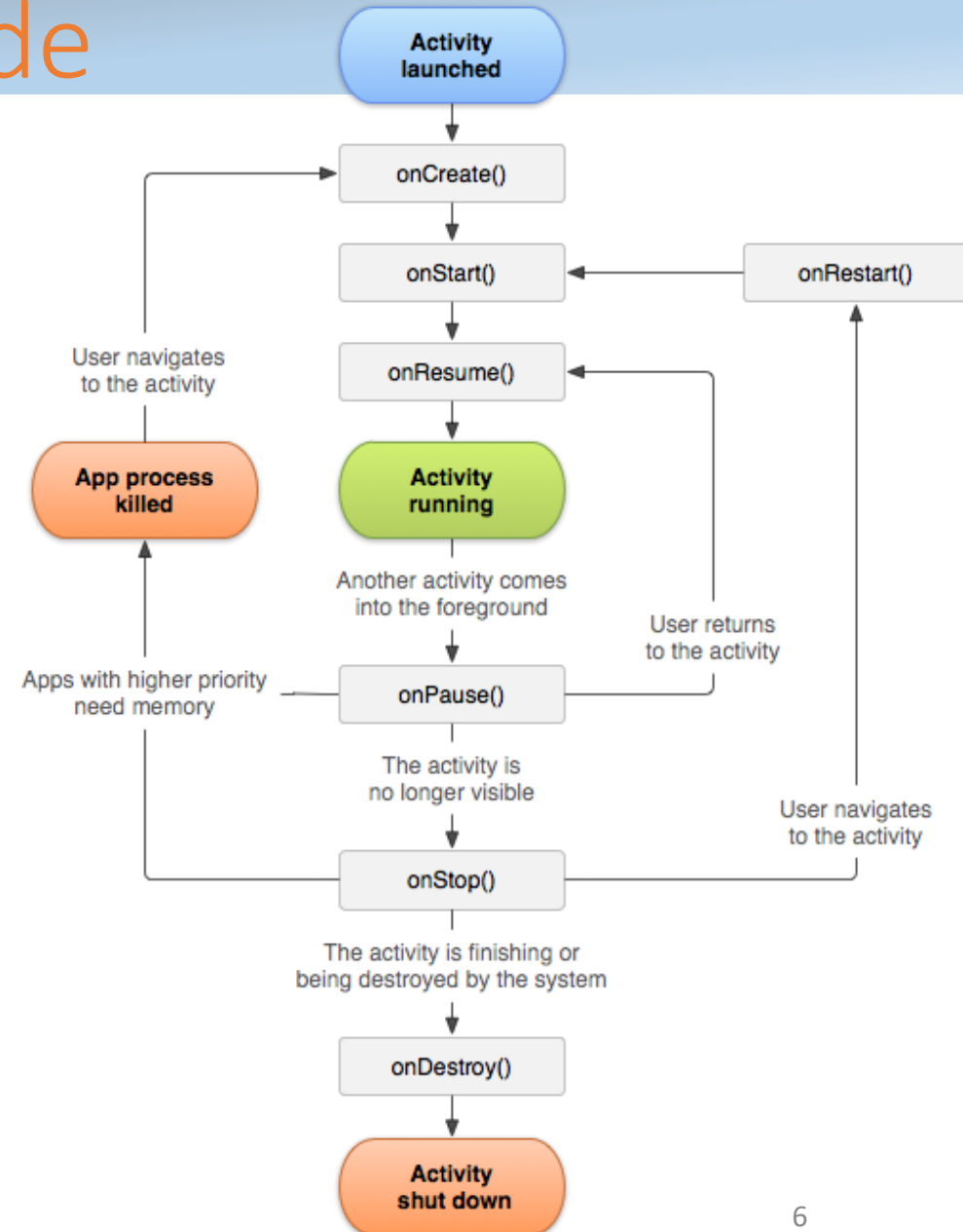
- Para criar uma atividade:
  - é necessário criar uma subclasse de *Activity*;
  - ou uma subclasse existente como uma *AppCompatActivity*;
- Sempre que uma atividade pára porque uma nova se inicia
  - é notificada desta mudança no seu estado;
  - recorrendo aos métodos de *callback*
    - que estão definidos para o ciclo de vida das atividades;
    - estes métodos dependem da ação do sistema sobre a atividade:
      - a **criar**;
      - a **parar**;
      - a **retomar /reiniciar**;
      - a **destruir**.



# Estados de uma Atividade

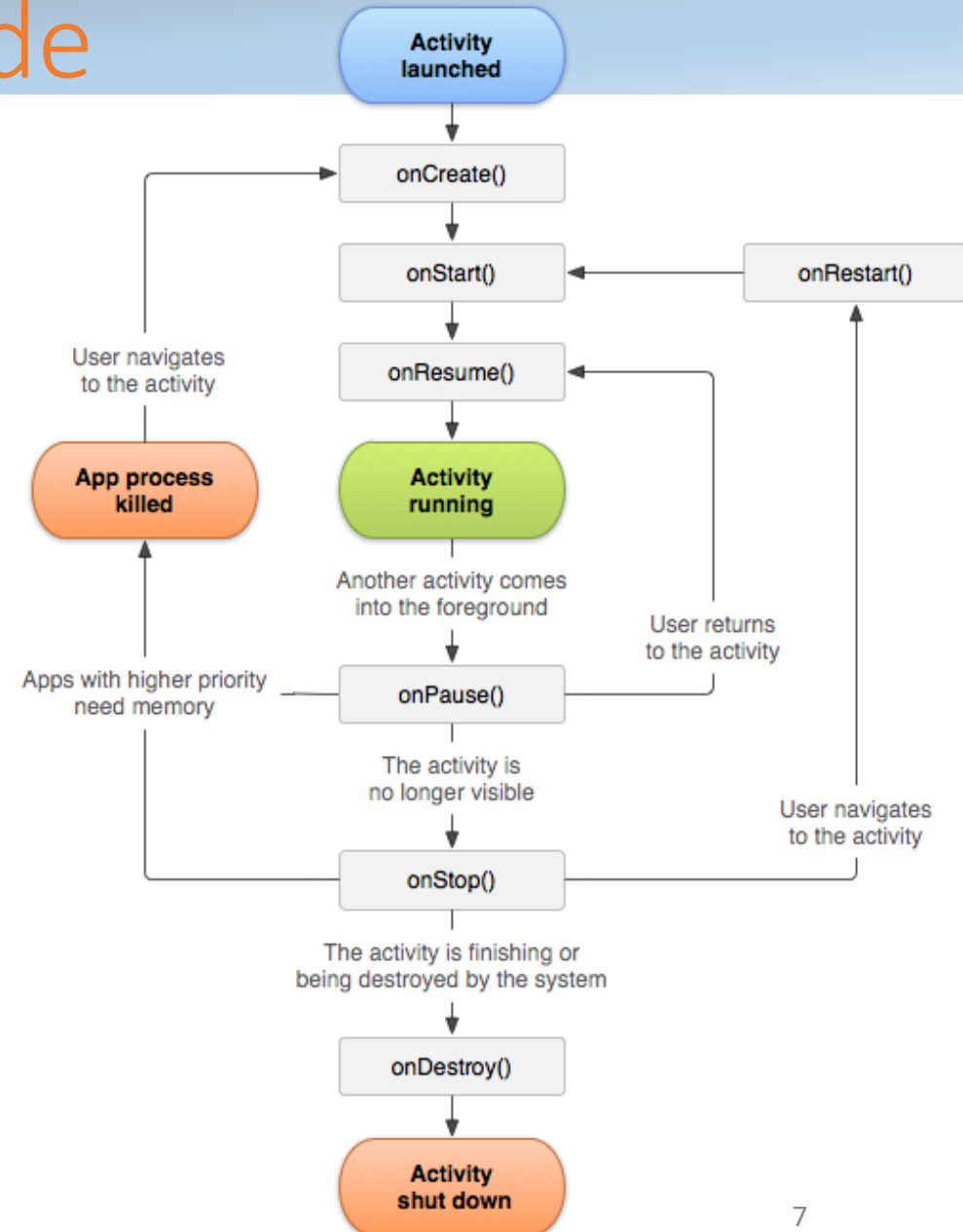
Uma atividade pode existir essencialmente em três estados:

- **Resumed** (retomado/reiniciado)
  - A atividade está em primeiro plano no ecrã e tem o foco do utilizador
    - Em geral, é designado de “*running (em execução)*”



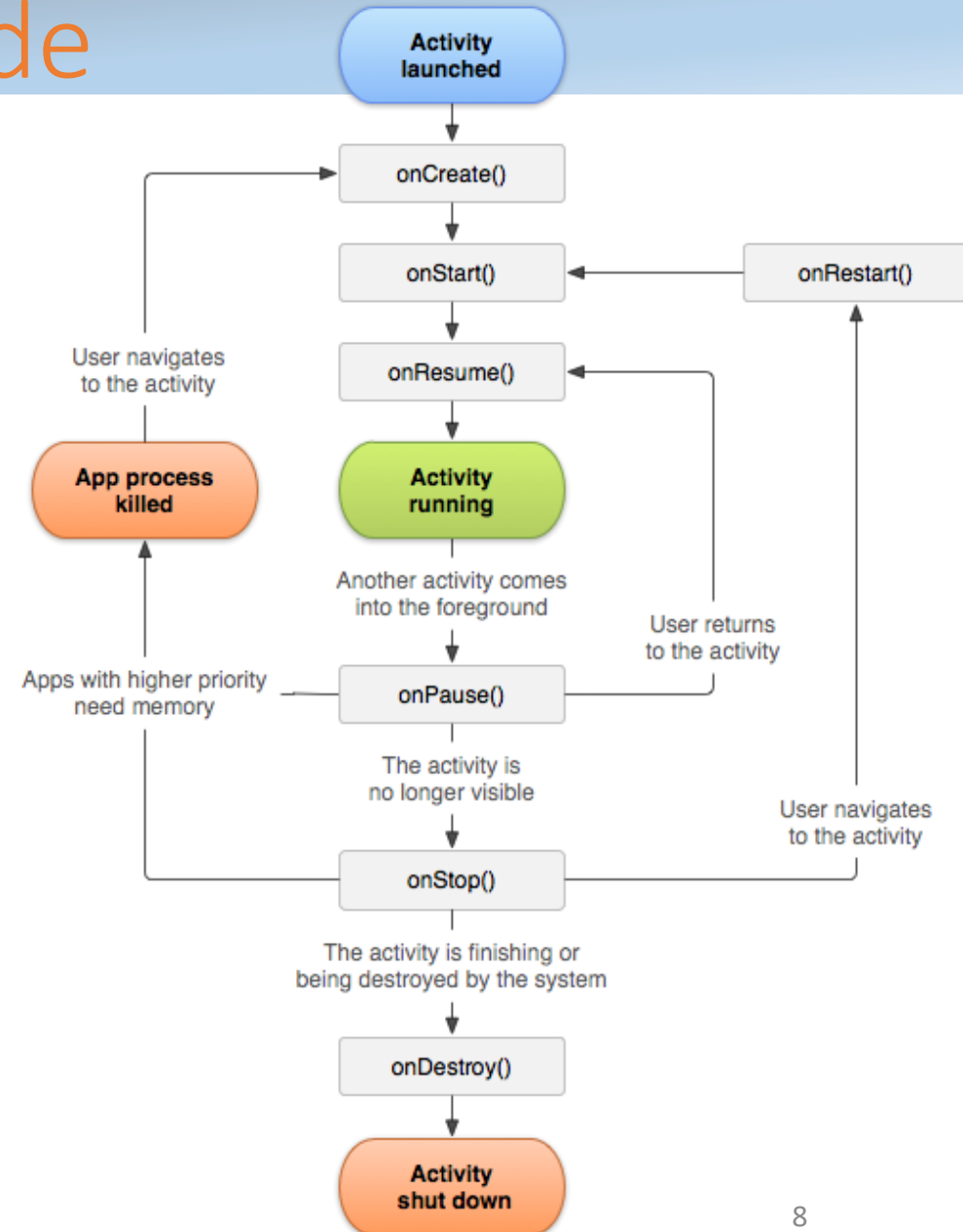
# Estados de uma Atividade

- ***Paused*** (pausado/em pausa)
  - A atividade ainda está visível;
  - Mas outra atividade está em primeiro plano e tem o foco;
    - Ou seja, outra atividade está visível por cima desta
      - mas está parcialmente transparente ou não cobre totalmente o ecrã;
- Uma atividade *pausada* está totalmente ativa
  - mas pode ser eliminada pelo sistema em situações de memória extremamente baixa.



# Estados de uma Atividade

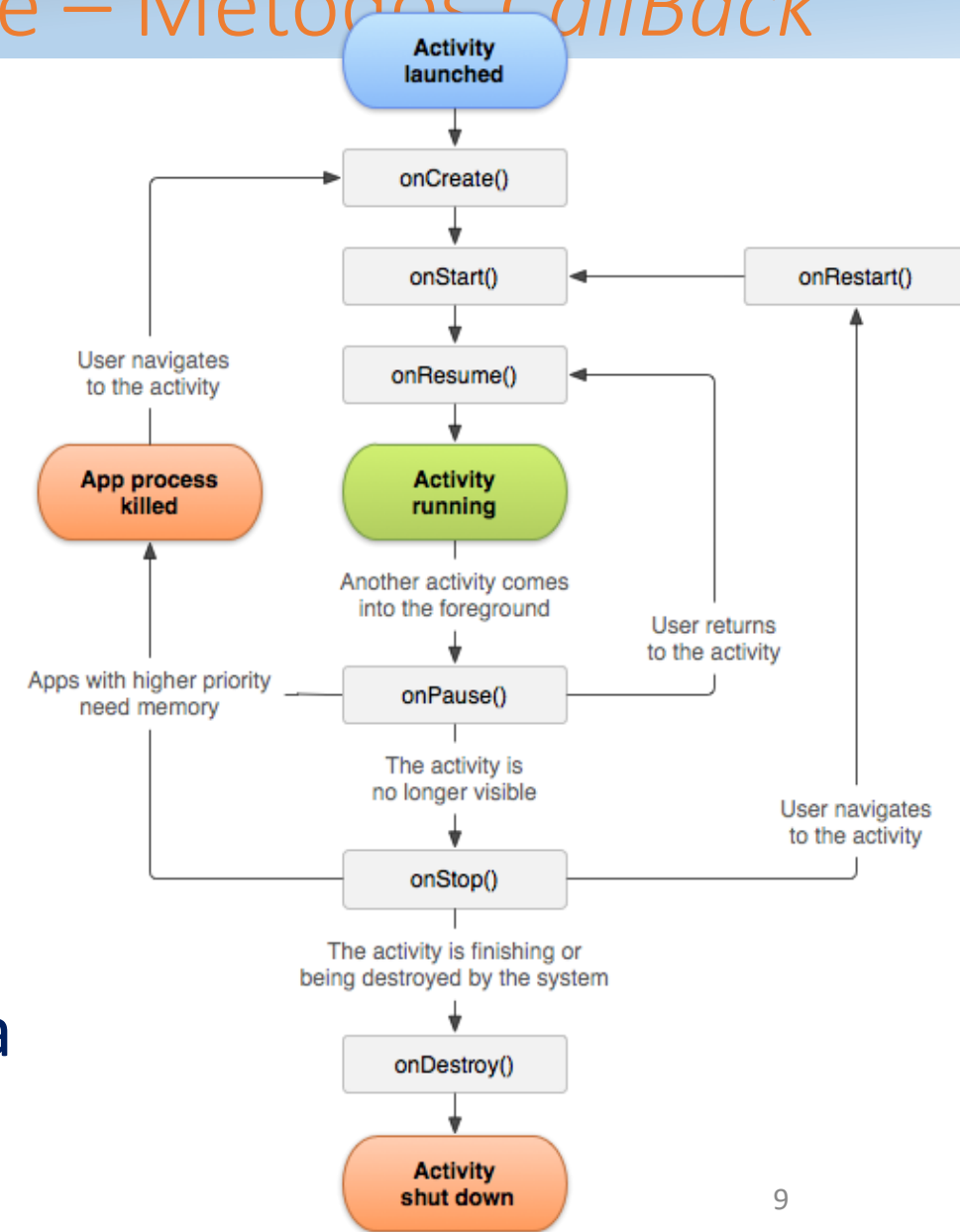
- **Stopped** (interrompido)
  - A atividade está completamente obscurecida por outra atividade
    - Está agora em “background”;
- A atividade continua ativa
  - É mantida em memória mas *não* está anexada ao gestor de janelas;
- Já não está visível para o utilizador
  - E pode ser eliminada pelo sistema se for necessário memória para outro processo.





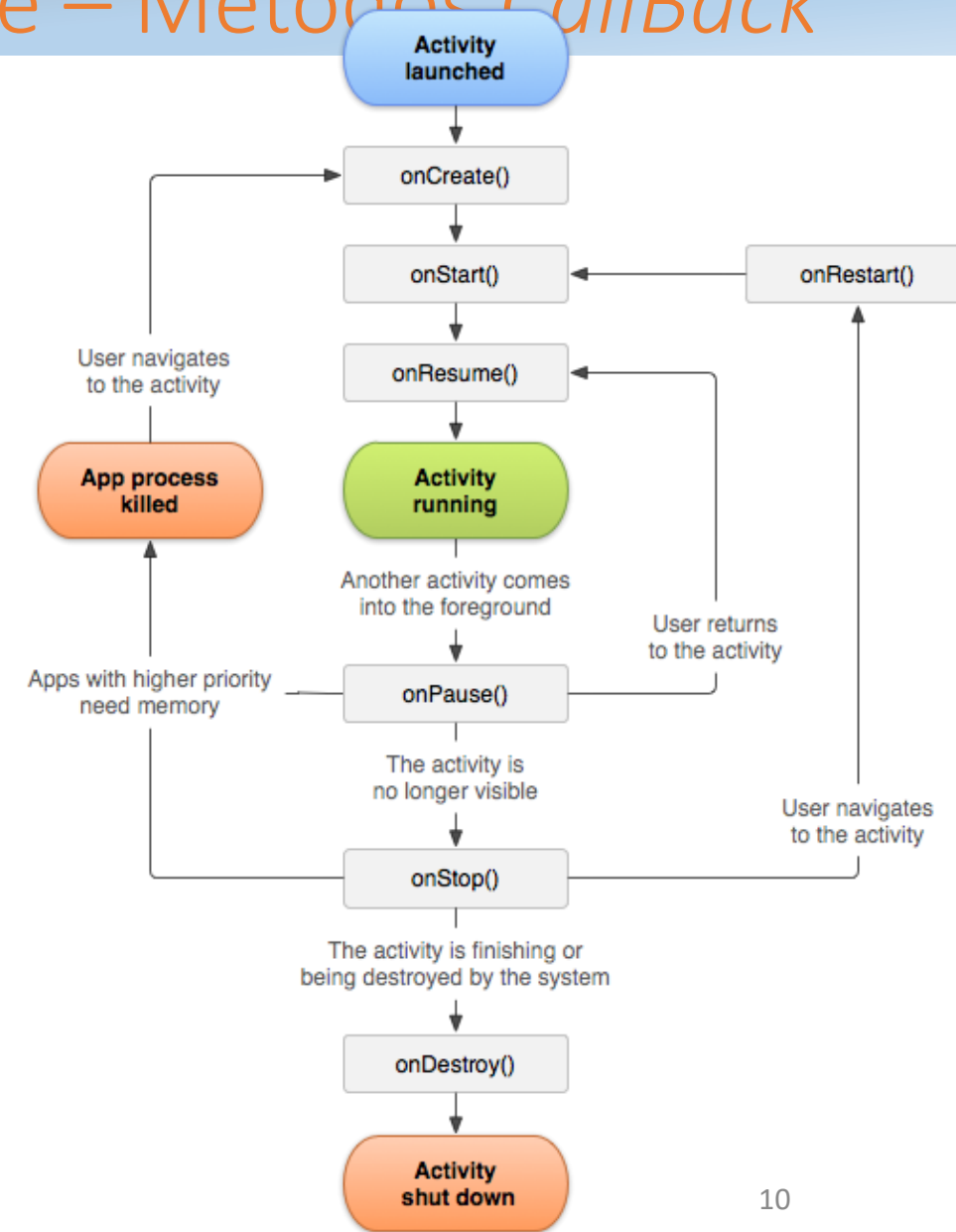
# Ciclo de Vida de uma Atividade – Métodos *CallBack*

- Quando uma atividade transita entre os diferentes estados descritos anteriormente;
- É notificada através de diversos métodos de ***CallBack***
  - São ganchos que podem ser modificados para executar o trabalho apropriado quando o estado da atividade muda;
- O esqueleto de atividade apresentado a seguir contém cada um dos métodos fundamentais do ciclo de vida:



# Ciclo de Vida de uma Atividade – Métodos *CallBack*

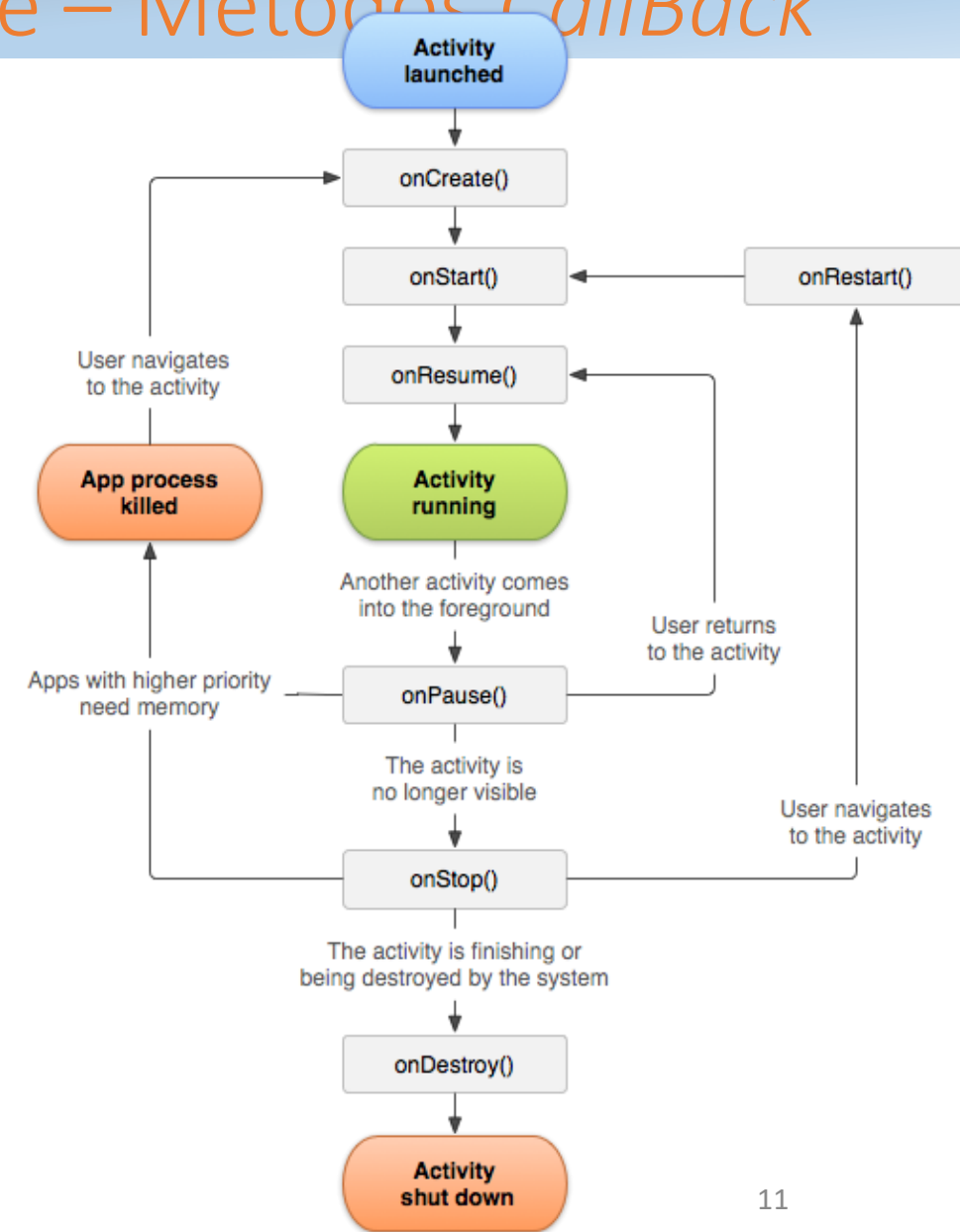
```
public class ExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // The activity is being created.
    }
    @Override
    protected void onStart() {
        super.onStart();
        // The activity is about to become visible.
    }
    @Override
    protected void onResume() {
        super.onResume();
        // The activity has become visible (it is now "resumed").
    }
    @Override
    protected void onPause() {
        super.onPause();
        // Another activity is taking focus (this activity is about to be "paused").
    }
    @Override
    protected void onStop() {
        super.onStop();
        // The activity is no longer visible (it is now "stopped")
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        // The activity is about to be destroyed.
    }
}
```



# Ciclo de Vida de uma Atividade – Métodos *CallBack*

Genericamente, pode considerar-se:

- “tempo de vida” de uma atividade a tudo o que acontece entre a chamada do método `onCreate()` e a chamada ao `onDestroy()`;
- “tempo de visibilidade” de uma atividade é o que acontece entre a chamada ao `onStart()` e ao `onStop()`;
- e o “tempo de foreground” acontece entre as chamadas dos métodos `onResume()` e `onPause()`.



# Intents

- Um *Intent* é um objeto que permite estabelecer a ligação entre dois componentes distintos
  - Entre duas atividades, ou entre atividade e serviço, etc.;
- É uma “*intenção em fazer algo*”;
- Os *Intents* são utilizados normalmente para iniciar outra atividade;
- Podem ter filtros que são especificados no **AndroidManifest.xml**
  - Filtros por ação, categoria, ou tipo de dados;

# Intents – Iniciar uma Atividade

1. Configurar o `AndroidManifest.xml` (opcional)
  - `intent-filter`
2. Criar um *Intent*, indicando a atividade a iniciar
3. Configurar os dados a passar à atividade (opcional)
4. Invocar o método ***startActivity()*** com o *intent* criado em 2.

# Tipos de Intents

- *Intents* explícitos:
  - Especificam o componente a iniciar pelo nome (o nome de classe totalmente qualificado);
- Usados para iniciar um componente na própria aplicação
  - porque se sabe o nome de classe da atividade ou serviço que se deseja iniciar.
    - Por exemplo, iniciar uma nova atividade em resposta a uma ação do utilizador ou iniciar um serviço para fazer o download de um ficheiro em segundo plano.

# Intents Explícitos

- Exemplo para iniciar uma atividade com nome ***SignInActivity***:

```
Intent intent = new Intent(this, SignInActivity.Class) ;  
startActivity(intent) ;
```

**Context:**

"Pai" da Actividade

# Tipos de Intents

- *Intents* implícitos:
  - Não nomeiam nenhum componente específico;
  - Mas declaram uma ação geral a realizar, o que permite que um componente de outra aplicação a processe;
- Plataforma oferece vários *Intents* para uso comum:
  - Manipulação de alarmes, Calendários
  - E-mail, Contactos, SMS
  - Mapas, Câmara
  - ... <https://developer.android.com/guide/components/intents-common.html>



# Intents Implícitos

- Exemplo para abrir um determinado URL:

```
public class MainActivity extends AppCompatActivity {  
  
    private EditText editTextURL;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        editTextURL = findViewById(R.id.editTextURL);  
    }  
  
    public void onClickAbrir(View view) {  
        String url = editTextURL.getText().toString();  
        if( url.length() > 0 ){  
            Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(url));  
            startActivity(intent);  
        }  
    }  
}
```

# Intents – Bundle Extra

- Um *Intent* também permite a passagem de dados de uma atividade para outra
  - Utilizando o *bundle* “Extra”;
  - Um *bundle* assume a forma de um dicionário
    - isto é, um conjunto de pares chave+valor;
    - e pode ser utilizado da seguinte forma:

```
Intent intent = new Intent(Intent.ACTION_SENDTO);  
intent.setData(Uri.parse("mailto:" + email));  
intent.putExtra(Intent.EXTRA_SUBJECT, "PSI - AMSI 2021/2022");  
startActivity(intent);
```

- onde o **email** corresponde a uma string com o endereço de email;

# Intents – Iniciar Aplicação

- Quando usamos um *Intent* Implícito
  - Para executar uma ação através de uma aplicação no dispositivo
    - Devemos verificar se o dispositivo tem alguma aplicação
      - Através dos métodos **resolveActivity()** e **getPackageManager()**

```
Intent intent = new Intent(Intent.ACTION_SENDTO);  
intent.setData(Uri.parse("mailto:" + email));  
intent.putExtra(Intent.EXTRA_SUBJECT, "PSI - AMSI 2021/2022");  
//startActivity(intent);  
  
if (intent.resolveActivity(getPackageManager()) != null) {  
    startActivity(intent);  
}
```

# Iniciar um atividade para um resultado

- Por vezes pode ser necessário receber um resultado da atividade que foi iniciada;
- Nesse caso deve iniciar a atividade através do método ***startActivityForResult()*** (em vez do *startActivity()*);
- Para receber o resultado da atividade iniciada, é necessário reimplementar o método *callback* ***onActivityResult()***;
- Quando a atividade subsequente é feita, ela devolve um resultado através de um *Intent* para o método ***onActivityResult()***.



# Exercício

- No projeto **GuessNumber**:
  1. Iniciar, a partir de um botão na **MainActivity**, uma nova atividade “**SobreActivity**”, que deve incluir o nome e o email de quem a realizou;
  2. Adicionar à **SobreActivity** a informação relativa a quantas vezes acertou no número gerado;
  3. Perguntar ao utilizador, através de uma **ResultadoActivity**, se deseja reiniciar o jogo, após ter acertado ou alcançado o número de tentativas disponível;

# Exercício 1

- Iniciar, a partir de um botão na **MainActivity**, uma nova atividade “**SobreActivity**”, que deve incluir o nome e o email de quem a realizou;
  - Criar um layout com 2 *TextView* para Nome e Email;
  - Iniciar a atividade a partir da criação do *intent* e do respetivo `startActivity()`;

```
public class MainActivity extends AppCompatActivity {  
    ...  
    public void onClickSobre(View view) {  
        Intent intent = new Intent(this, SobreActivity.class);  
        startActivity(intent);  
    }  
}
```

## Exercício 2

- Adicionar à **SobreActivity** a informação relativa a quantas vezes acertou no número gerado;
- Criar um atributo para guardar o número de vezes que acertou;
- Adicionar ao *Intent* um bundle “Extra” para passar o valor e guardar na constante definida para o efeito;

```
public class SobreActivity extends AppCompatActivity {  
    public static final String NUMERO_GUESS = "pt.ipleiria.estg.dei.amsi.NUMERO_GUESS";  
    private TextView textViewResultado;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_sobre);  
        textViewResultado = findViewById(R.id.textViewResultado);  
        int numGuess = getIntent().getIntExtra(NUMERO_GUESS, -1);  
        textViewResultado.setText("Acertou no valor: " + numGuess + "vezes.");  
    }  
}
```

```
public class MainActivity extends AppCompatActivity {  
    //...  
    public void onClickSobre(View view) {  
        Intent intent = new Intent(this, SobreActivity.class);  
        intent.putExtra(SobreActivity.NUMERO_GUESS, numGuess);  
        startActivity(intent);  
    }  
}
```

## Exercício 3

- Perguntar ao utilizador, através de uma **ResultadoActivity**, se deseja reiniciar o jogo, após ter acertado ou alcançado o número de tentativas disponível, fechando a atividade;
  - Criar um layout com dois botões:
    - *Ok e Cancel;*

```
public class ResultadoActivity extends AppCompatActivity {  
    public static final String RESULTADO =  
        "pt.ipleiria.estg.dei.amsi.RESULTADO";  
    private TextView textViewResultado;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        //...  
        String resultado = getIntent().getStringExtra(RESULTADO);  
        textViewResultado.setText(resultado);  
    }  
    public void onClickOk(View view) {  
        setResult(RESULT_OK);  
        finish();  
    }  
    public void onClickCancel(View view) {  
        setResult(RESULT_CANCELED);  
        finish();  
    }  
}
```



## Exercício 3

- Criar o *intent* adicionando o texto de resultado como um *Extra*;
- Iniciar a atividade a partir do método **startActivityResult()** e avaliar o resultado devolvido a partir do método **onActiviyResult()**;

```
public class MainActivity extends AppCompatActivity {
    private static final int JOGAR_NOVAMENTE = 1;
    //...

    private void apresentarResultado(String resultado) {
        Intent intent = new Intent(this, ResultadoActivity.class);
        intent.putExtra(ResultadoActivity.RESULTADO, resultado);
        startActivityForResult(intent, JOGAR_NOVAMENTE);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        if( requestCode == JOGAR_NOVAMENTE)
            if( resultCode == RESULT_OK) {
                resetGame();
            } else {
                finish();
            }
        }
    }
}
```



# Fontes e Mais Informação

- Atividades

- <http://developer.android.com/guide/components/activities.html>

- Intents e filtros de Intents

- <http://developer.android.com/guide/components/intents-filters.html>

- Gerir o ciclo de vida de uma atividade

- <https://developer.android.com/guide/components/activities/activity-lifecycle>

- Interagir com outras Apps

- <http://developer.android.com/training/basics/intents/index.html>

- Partilhar dados simples

- <http://developer.android.com/training/sharing/index.html>

# Próximo Tema:

## ***Desenvolvimento Móvel em Android: Fragmentos e NavigationView***

- Fragmentos
  - <https://developer.android.com/guide/components/fragments>
- *DialogFragment*
  - <https://developer.android.com/guide/topics/ui/dialogs>
  - <https://guides.codepath.com/android/using-dialogfragment>
- *NavigationView*
  - <https://developer.android.com/guide/navigation>