

AULA 05

IMD0509 - DESENVOLVIMENTO PARA **DISPOSITIVOS MÓVEIS**

Prof. Emerson Alencar

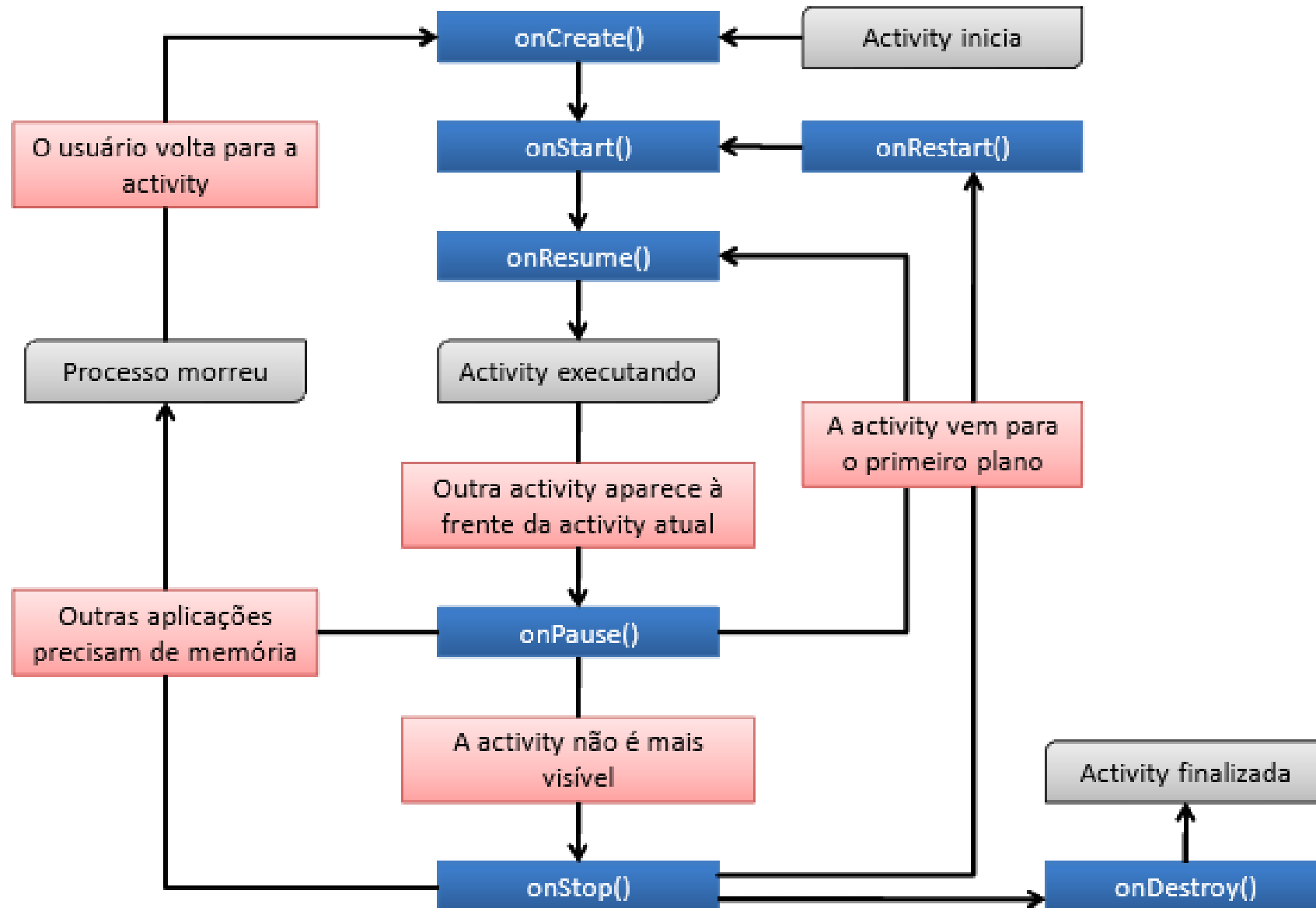
emerson@imd.ufrn.br



- Métodos de callback invocados pelo Android durante o ciclo de vida da activity
 - onCreate()
 - onStart()
 - onResume()
 - onPause()
 - onStop()
 - onRestart()
 - onDestroy()

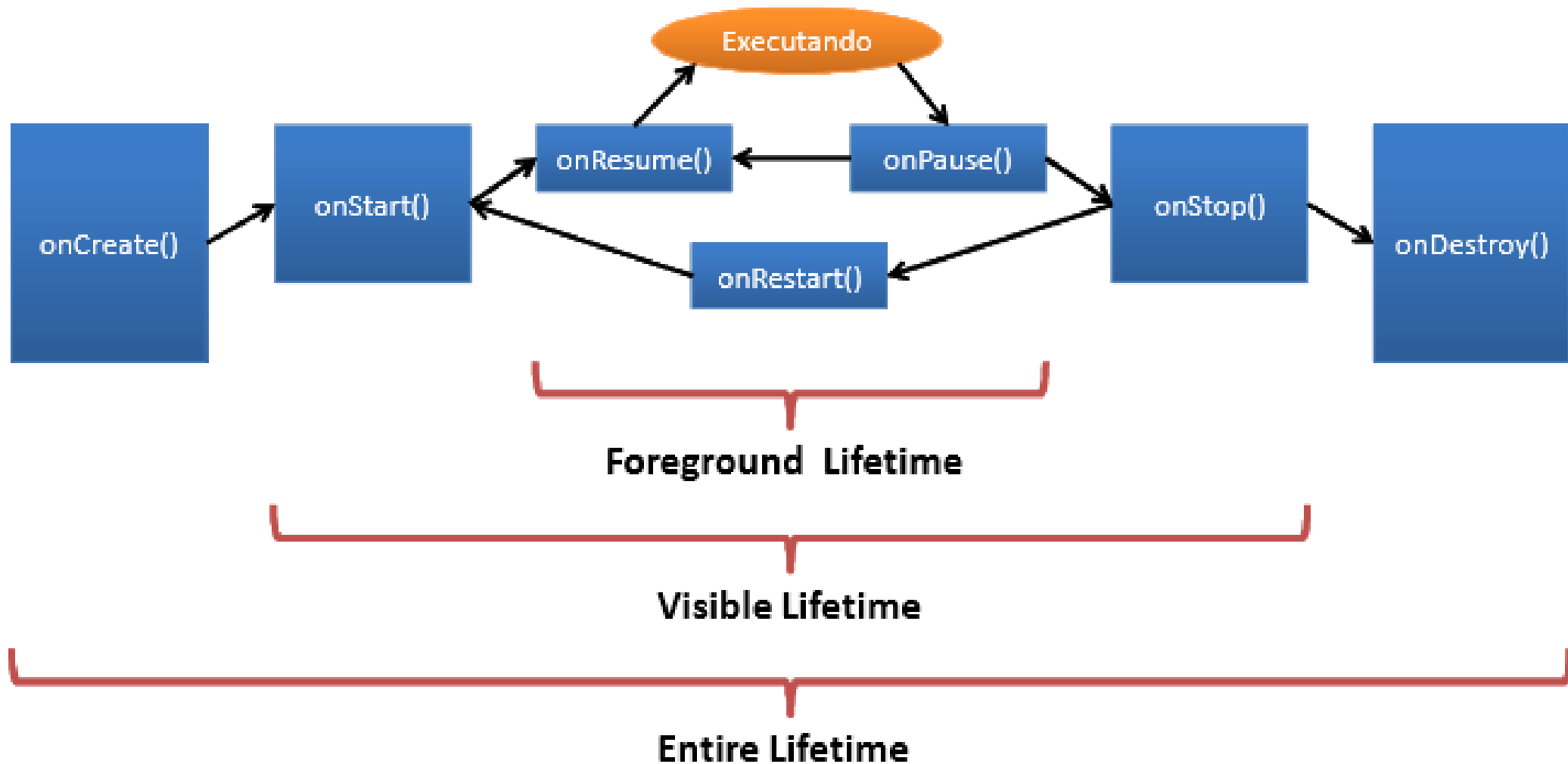
CICLO DE VIDA - ACTIVITY

_ X



CICLO DE VIDA - ACTIVITY

_ X



SALVANDO O ESTADO - ACTIVITY



- Uma activity pode ser destruída pelo Android para liberar recursos
- Quando ela for recriada, é possível que informações da activity sejam perdidas
- Para evitar que isto aconteça, é necessário salvar o estado da activity
- Não é preciso se preocupar com o estado das views, pois ele é gravado e depois recuperado
 - É preciso que a view tenha um ID especificado

O MÉTODO `onSaveInstanceState()` _ X

- O Android chama o método `onSaveInstanceState()` na activity quando ela não fica mais visível.

```
9  public class SegundaActivity extends Activity {
10
11     private String nome;
12     private int idade;
13
14     @Override
15     protected void onSaveInstanceState(Bundle outState) {
16         outState.putString("nome", nome);
17         outState.putInt("idade", idade);
18
19         super.onSaveInstanceState(outState);
20     }
21 }
```

Bundle para salvar os dados

Chama o método da superclasse

- Quando a activity é destruída, o estado dela fica guardado.

RESTAURANDO O ESTADO-ACTIVITY _ X

- Quando a activity for recriada, o Android chama o método onCreate() passando como parâmetro o bundle que contém os dados salvos

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    if(savedInstanceState != null){

        this.nome = savedInstanceState.getString( key: "nome");
        this.idade = savedInstanceState.getInt( key: "idade");
    }
}
```

Se o Bundle não é **null**, a activity é restaurada

- O método ***onRestoreInstanceState(Bundle)*** também é chamado (depois de onStart())

- Utilizar logs é uma forma de mostrar informações úteis durante a execução do programa
- A forma mais simples de fazer isso no Java é através de `System.out.println()`
 - Não é possível categorizar o erro

- O LogCat é um mecanismo versátil de logging do Android
- Acessado através da classe **android.util.Log**, que possui alguns métodos estáticos

Método	Tipo de Mensagem
Log.v()	Verbose (depuração detalhada)
Log.d()	Debug (depuração)
Log.i()	Info (informação)
Log.w()	Warning (aviso)
Log.e()	Error (erro)

- Invocar os métodos da classe Log de acordo com o tipo de mensagem

Log.i (“Tela Inicial”, “Iniciando processo”)

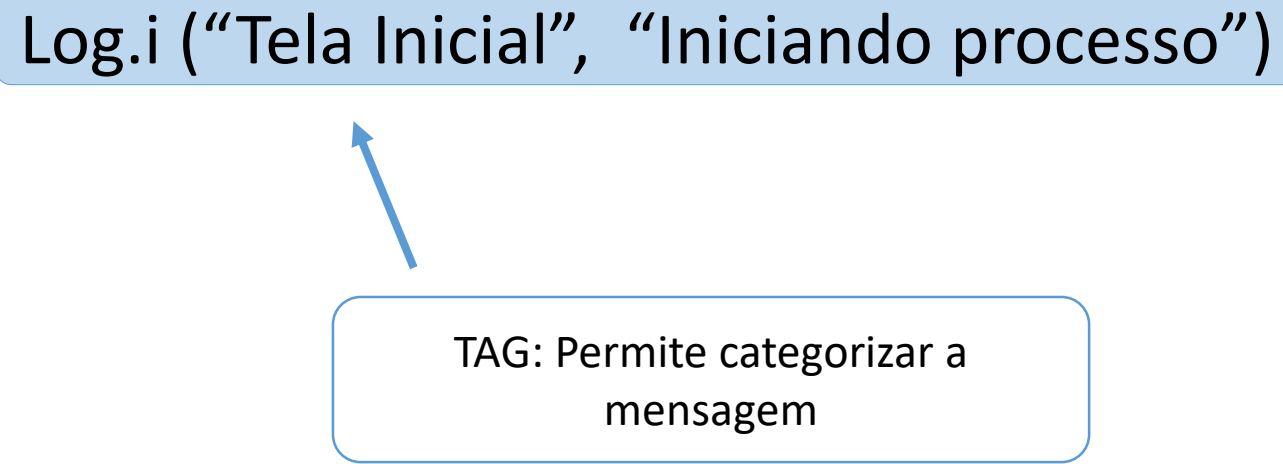


Diagram illustrating the use of the Log.i method. A light blue box contains the code `Log.i (“Tela Inicial”, “Iniciando processo”)`. A blue arrow points from a white box below to the first parameter of the Log.i method call. The white box contains the text: TAG: Permite categorizar a mensagem.

TAG: Permite categorizar a mensagem

Acessando o Android Studio



The background is a solid blue color. It is decorated with a complex, abstract geometric pattern. This pattern consists of numerous white dots of varying sizes, some of which are connected by thin white lines to form a network of triangles and other polygons. Scattered throughout the design are several yellow elements: small solid dots, thin circles, plus signs, and a single diamond shape. The overall aesthetic is modern and mathematical.

Perguntas?