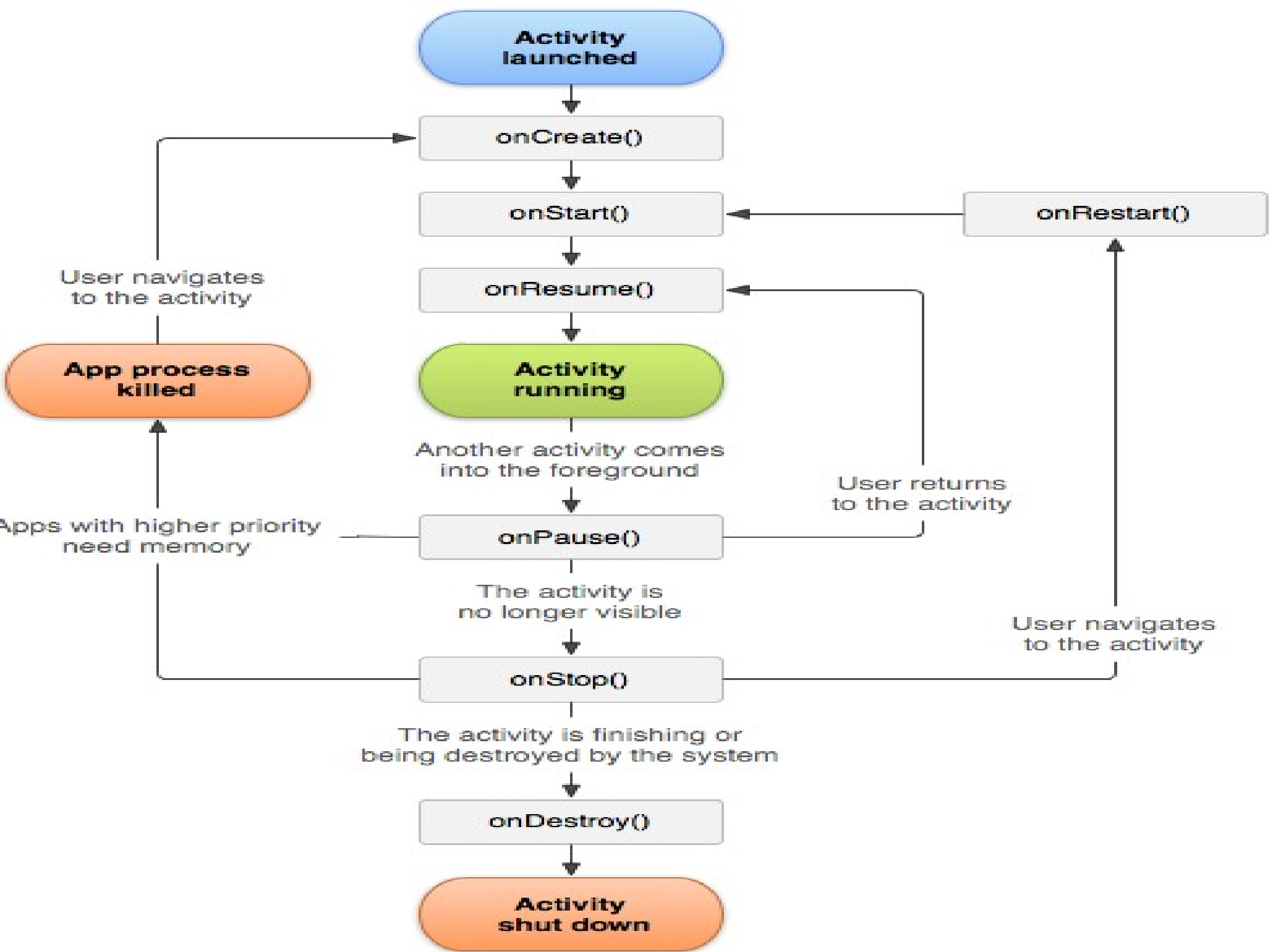


Android: Ciclo de vida da Activity

Se sua aplicação estiver em execução e o celular receber uma ligação? O que acontecerá com seu aplicativo? Quais métodos serão executados e em qual ordem? Para respondermos essas perguntas e entendermos um pouco mais sobre activities é essencial conhecer o “ciclo de vida” das mesmas.



Android: Ciclo de vida da Activity

- **onCreate()** É o primeiro método a ser executado quando uma Activity é lançada. Geralmente é responsável por carregar os layouts XML e outras operações de inicialização. É executada somente uma vez durante a “vida útil” da Activity.
- **onStart()** É chamado imediatamente após a onCreate() – e também quando uma Activity que estava em background volta a ter foco.

Android: Ciclo de vida da Activity

- **onResume()** Assim como a `onStart()`, é chamado na inicialização da Activity (logo após `onStart()`) e também quando uma Activity volta a ter foco. Qual a diferença entre as duas? A `onStart()` só é chamada quando a Activity não estava mais visível na tela e volta a ter o foco, enquanto a `onResume()` sempre é chamada nas “retomadas de foco”.

Android: Ciclo de vida da Activity

- **onPause()** É o primeiro método a ser invocado quando a Activity perde o foco (ou seja, outra Activity vem à frente).
- **onStop()** - Análoga à onPause(), só é chamada quando a Activity fica completamente encoberta por outra Activity (não é mais visível).

Android: Ciclo de vida da Activity

- **onDestroy()** O último método a ser executado. Depois dela, a Activity é considerada “morta” - ou seja, não pode mais ser relançada. Se o usuário voltar a requisitar essa Activity, outro objeto será construído.
- **onRestart()** Chamado imediatamente antes de onStart(), quando uma Activity volta a ter o foco depois de estar em background.

Android: Ciclo de vida da Activity

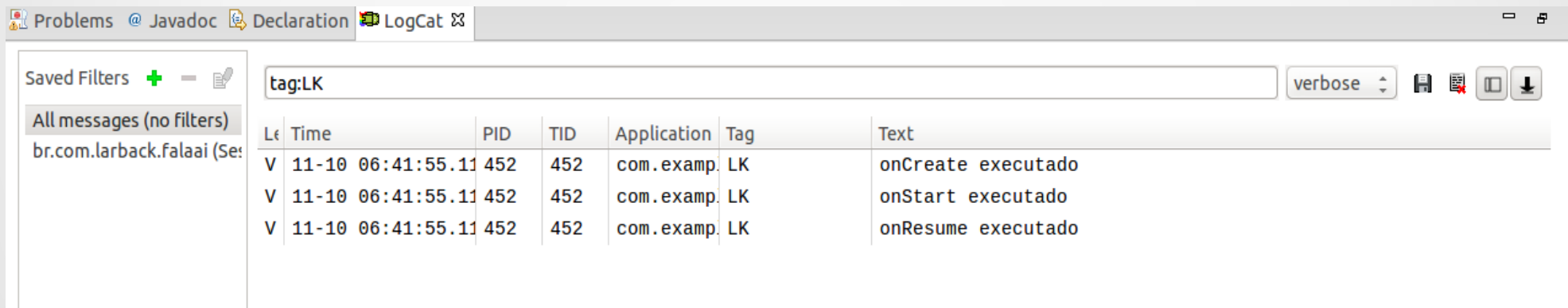
Para conseguirmos um entendimento total deste assunto, implementaremos um exemplo que imprimirá uma mensagem no log toda vez que uma fase do ciclo de vida for executada. Executaremos o aplicativo e monitoraremos o log.

Android: Ciclo de vida da Activity

```
public class MainActivity extends Activity {
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState); setContentView(R.layout.activity_main);
    Log.v("LK","onCreate executado");
}
protected void onDestroy() {    super.onDestroy();    Log.v("LK", "onDestroy executado");    }
protected void onPause() {    super.onPause();    Log.v("TAG", "onPause executado");    }
protected void onResume() {    super.onResume();    Log.v("LK", "onResume executado");    }
protected void onStart() {    super.onStart();    Log.v("LK", "onStart executado");    }
protected void onStop() {    super.onStop();    Log.v("LK", "onStop executado");
}
protected void onRestart() { super.onRestart(); Log.v("LK", "onRestart executado"); }
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
}
```


Android: Ciclo de vida da Activity

Após a implementação, execute a aplicação e abra o LogCat (window->show view ->Other...-> Android->LogCat). Na barra de pesquisa filtre as mensagens pela tag criada (no exemplo, “LK”)



The screenshot shows the LogCat window in Android Studio. The search filter is set to 'tag:LK'. The log messages are filtered to show only those with the tag 'LK'. The messages are as follows:

Level	Time	PID	TID	Application	Tag	Text
V	11-10 06:41:55.11	452	452	com.examp	LK	onCreate executado
V	11-10 06:41:55.11	452	452	com.examp	LK	onStart executado
V	11-10 06:41:55.11	452	452	com.examp	LK	onResume executado

Android: Ciclo de vida da Activity

Para finalizar, com louvor, seus testes, deixe sua aplicação aberta e simule uma ligação para o emulador.

Para isso, abra o terminal do seu sistema operacional, conecte-se ao emulador via telnet:

telnet localhost 5554 (onde 5554 é a porta do emulador)

E ligue para o mesmo com o comando `gsm call <numero origem>`:

gsm call 12345

Android: Ciclo de vida da Activity

Para saber mais, leia a documentação oficial da classe Activity disponível (em inglês) em:

<http://developer.android.com/reference/android/app/Activity.html>

Qualquer dúvida, entre em contato pelo meu site pessoal ou através do nosso grupo de estudos.

<http://www.larback.com.br>

<https://www.facebook.com/groups/novoid/>