

# Desenvolvimento de uma aplicação para celular (resumão)

PROF. KALIL DE OLIVEIRA

## Depois do Android, celulares fazem de tudo, inclusive chamadas telefônicas!

- >> Um dos produtos mais consumidos do mundo, há cerca de três *smartphones* para cada computador.
- >> O Android, que vem na maioria deles, surgiu de uma aliança da Google com HTC, Intel e outras empresas.
- >> Características: adaptável (código aberto), seguro (aplicativos isolados e que requerem autorização) etc.
- >> Cinco Camadas: Applications, Applications Framework, Libraries, Android Runtime e Linux Kernel.
- >> *Applications*: Tipos de componentes nesta camada incluem calendário, mapas, câmeras, jogos etc.
- >> *Applications Framework*: fornece APIs de alto nível que os desenvolvedores usam para criar seus aplicativos.
- >> *Libraries*: Tipos de componentes nesta camada: *Media Framework* (gravação e reprodução de áudio e vídeo), *WebKit* (layout de navegador de código aberto) e *SQLite* (um banco de dados leve), entre outros.
- >> *Android Runtime*: Camada responsável pela execução de aplicativos Android.
- >> *Linux Kernel*: é a base do sistema operacional Android e fornece drives para hardware, entre outros.
- >> Ferramentas para desenvolvimento: Software Development Kit (SDK) da linguagem Java e do Android.
- >> Site indispensável para desenvolvedores (tutoriais, ferramentas, notícias etc.): <https://developer.android.com>.
- >> Desenvolver um app requer várias fases como: os requisitos (as regras do negócio), a interface gráfica (design), o ambiente de desenvolvimento, o código, testes, documentação, compilação, empacotamento, validação final, publicação e atualizações.

## Seu primeiro programa: "Olá mundo"

- >> Estrutura de pastas: no console do windows, vá até a raiz `cd \` e digite o comando que cria as pastas com uma única linha `mkdir olamundo\src\main olamundo\res olamundo\bin`.
- >> Arquivo 1: **OlaMundo.java**: Considerando que você já tem o SDK Java instalado e o SDK do Android na sua máquina, acione o bloco de notas: `notepad c:/olamundo/bin/OlaMundo.java`.
  - >> Parte 1: Bibliotecas necessárias: `import android.app.Activity; import android.os.Bundle; import android.widget.TextView;`
  - >> Parte 2: Assinatura da classe estende Activity: `public class OlaMundo extends Activity {`
  - >> Parte 3: Sobrescrevemos o método: `@Override protected void onCreate(Bundle retornoDeInstanciaSalva) {`
  - >> Parte 4: Bundle é passado para onCreate da Activity: `super.onCreate(retornoDeInstanciaSalva);`
  - >> Parte 5: Componente txtOla é criado na Activity: `TextView txtOla = new TextView(this);`
  - >> Parte 6: Conteúdo do componente de texto é atualizado: `txtOla.setText("Olá Mundo!");`
  - >> Parte 7: Activity recebe o componente por parâmetro e fechamos os blocos: `setContentView(txtOla); }`
- >> Arquivo 2: **AndroidManifest.xml**: `notepad c:/olamundo/AndroidManifest.xml`.
  - >> Parte 1: namespace: `<manifest xmlns:android="http://schemas.android.com/apk/res/android">`
  - >> Parte 2: nome da aplicação : `<application> <activity android:name="OlaMundo">`
  - >> Parte 3: defini as intenções (como ser iniciado): `<intent-filter>`
  - >> Parte 4: ação principal (main) da aplicação: `<action android:name="android.intent.action.MAIN" />`
  - >> Parte 5: inicia (launcher) quando clicado: `<category android:name="android.intent.category.LAUNCHER" />`
  - >> Parte 6: fechando as tags: `</intent-filter> </activity> </application> </manifest>`
- >> Arquivo 3: **OlaMundo.class**: Compilar `javac -cp "C:\Android\platforms\android-30\android.jar" c:/olamundo/src/main/OlaMundo.java -d c:/olamundo/bin`

>> Os próximos passos são o empacotamento e instalação no celular. Quanto a isso, há ferramentas como o Gradle (um gerenciador de dependências, entre outros) e recursos do próprio SDK do Android que precisam ser analisados. Por fim, no celular, deve-se autorizar "Fontes desconhecidas" ou "Instalação de aplicativos de fontes desconhecidas".

### A facilidade do No-code : explorando o site Kodular (<https://www.kodular.io>)

>> Para praticar, faça uma tela com um lblOla, txtOla e btnOla. O conteúdo de lblOla diz "Olá. Qual é o seu nome?". A txtOla (um EditText) recebe a entrada de dados e btnOla recebe o clique. Na programação em bloco, a lógica é a seguinte:

```
Se (txtOla.Texto ≠ "") então:  
    set lblOla.Texto para "Bem-vindo " + get txtOla.Texto  
    set txtOla.Visible para false  
    set btnOla.Visible para false  
senão:  
    set lblOla.Texto para "Por favor, insira seu nome")
```

>> Outros exercícios. Divirta-se!

- (1) Calculadora de Somar. Componentes: txtNum1, txtNum2, btnOk, lblResultado;
- (2) Conversor de real em dólar, com: txtReal, btnConverter, lblResultado;
- (3) Lista de tarefas. Vai precisar de txtTarefa, btnAddTarefa, lstTarefa;

### Componentes são objetos como textViews, buttons ...

- >> *TextView*: Exibe texto na tela. Pode ser usado para mostrar informações estáticas.
- >> *EditText*: Permite ao usuário inserir texto. Usado em formulários e inputs.
- >> *Button*: Representa um botão clicável. Pode ser configurado para executar uma ação quando clicado.
- >> *ImageView*: Exibe uma imagem na tela. Usado para mostrar gráficos e ícones.
- >> *ListView*: Exibe uma lista de itens roláveis. Cada item pode ser um texto, imagem etc.
- >> *RecyclerView*: Versão mais flexível e eficiente do ListView. Usado para listas grandes e dinâmicas.
- >> *ScrollView*: Permite que o conteúdo dentro dele seja rolado, útil para layouts grandes.

### ... e objetos podem ser manipulados com eventos quando clicados, pressionados etc

- >> *onClick*: Evento de clique. Usado para capturar e responder a cliques de botão.
- >> *onLongClick*: Evento para quando um botão é pressionado por um longo período.
- >> *onTextChanged*: Evento para quando o texto em um EditText é alterado e validar entrada em tempo real.
- >> *onItemClick*: Evento para uma ListView ou RecyclerView. Usado para capturar e responder à seleção de itens.
- >> *onScroll*: Evento para quando o conteúdo em um ScrollView é rolado. Usado para carregar mais dados ou atualizar a interface.

### Um passo avançado : escolha uma IDE.

>> No site oficial do Desenvolvedor Android a sugestão é pelo Android Studio, mas considerando que esteja instalado junto com o SDK do Java e o SDK do Android, o Eclipse ou até mesmo o VSCode podem ajudá-lo a criar um projeto a partir de seus assistentes e plugins.

>> O caminho envolve geralmente um rotina de telas que pedem: nome, package e ícones, que resultam em uma estrutura de pastas. Entre elas SRC, com arquivos JAVA e LAYOUT (dentro de RES), o XML que possibilita ajustes importantes para os componentes da tela.

>> Com alguma configuração, o desenvolvedor pode clicar e arrastar os componentes e editar estes componentes com os assistentes. Salvando tudo e executando o run/Android Application será possível ver seu projeto emulado.