

Introdução aos dispositivos móveis

Introdução

Marcelo Zamith

e-mail: mzamith@ufrj.br
<https://www.dcc.ufrj.br/~marcelo/>

Universidade Federal Rural do Rio de Janeiro - DCC



Introdução



Star Trek (1966)

- O que é ou são dispositivos móveis ?

- O que é ou são dispositivos móveis ?
 - Apenas celulares ?

- O que é ou são dispositivos móveis ?
 - ▶ Apenas celulares ?
 - ▶ “*Dispositivos móveis são tecnologias digitais que permitem a mobilidade e o acesso à internet*”

- O que é ou são dispositivos móveis ?

- ▶ Apenas celulares ?
- ▶ “*Dispositivos móveis são tecnologias digitais que permitem a mobilidade e o acesso à internet*”
- ▶ *smartphones*, consoles, câmeras digitais, *netbooks*, *notebooks*, *tablets*, *smartwatch*, sistemas de navegação de veículos, e o que mais for conectável e portátil.

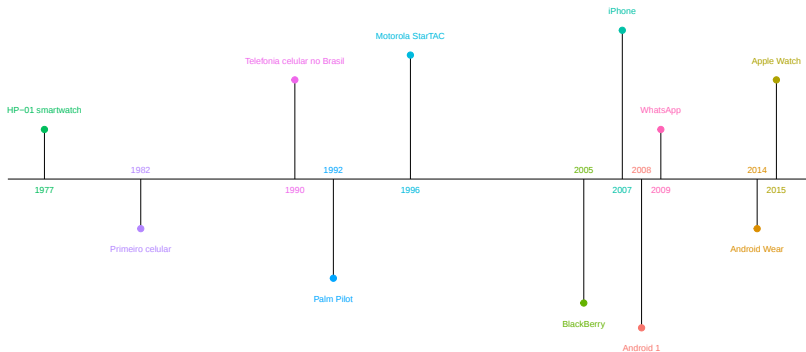
Introdução

- Motorola DynaTAC 8000x;
- produzido em 1972, comercializado em 1983;
- dimensões 33cm de altura, 4,5 cm de largura e 8,9 cm de espessura;
- peso de 794 gramas; e
- capacidade de memorizar até 30 números.
- Preço: US\$ **3.995,00**



- iPhone 14;
- dimensões 160,8mm de altura, 78,1mm de largura e 7,8mm de espessura;
- peso de 203 gramas; e
- capacidade de 512 GB.
- Preço: R\$ **11.599,00**

Introdução



Introdução

- Brasil em 2022:
 - ▶ 242 milhões de celulares “inteligentes” em uso no Brasil (CNN Brasil) ¹
 - ▶ 214 milhões de habitantes (IBGE)
- 14.02 bilhões de aparelhos em 2022
- Previsão para 2025: 18.22 bilhões

Introdução

- Sobre a Google Play em 2021²:
 - ▶ Apps na loja: 3.597.788
 - 2.184.456 na Apple Store
 - 480.174 na Amazon Appstore
 - ▶ 3.739 apps adicionados por dia
 - ▶ Apps baixados no ano 111.3 bilhões
- Março de 2022 - 97% dos apps são gratuitos

²<https://appinventiv.com/blog/google-play-store-statistics/>



Introdução

- Usamos o smartphone para que ?
 - ▶ comunicação (texto, voz e vídeo)
 - ▶ entretenimento
 - ▶ transações bancárias
 - ▶ obter informações (de forma geral)
 - ▶ acesso à serviços públicos

Introdução

Vale a pena estudar sobre mobile, smartphones e programação para dispositivos móveis ?



Introdução

- As principais plataformas de desenvolvimento:
 - ▶ Apple - iOS
 - ▶ Google - Android

Introdução

- As principais plataformas de desenvolvimento:
 - ▶ Apple - iOS
 - ▶ Google - Android - **+popular!** ←

Introdução

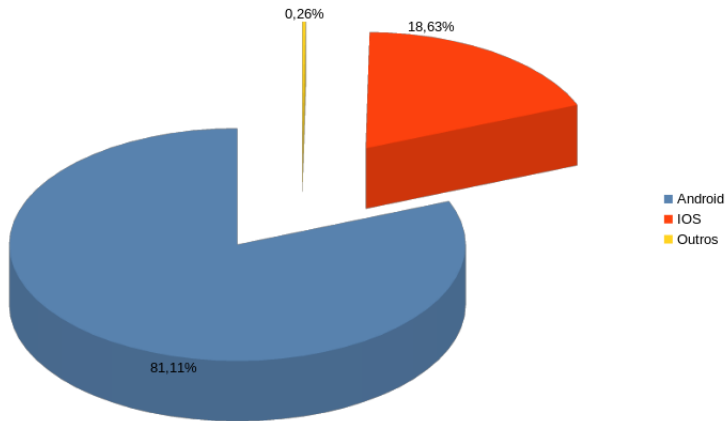
- As principais plataformas de desenvolvimento:
 - ▶ Apple - iOS
 - ▶ Google - Android

Introdução

- As principais plataformas de desenvolvimento:
 - ▶ Apple - iOS
 - ▶ Google - Android - **+popular!** ←

Introdução

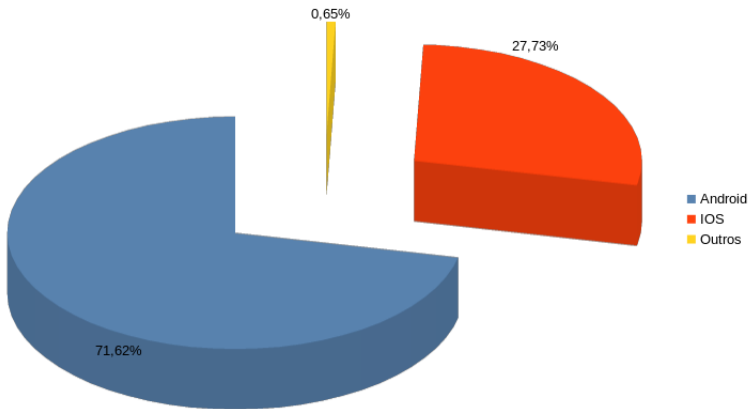
Brasil - setembro de 2022³



³ Fonte: <https://gs.statcounter.com/os-market-share/mobile/brazil>

Introdução

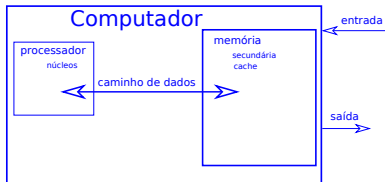
Mundo - setembro de 2022⁴



⁴ Fonte: <https://gs.statcounter.com/os-market-share/mobile/brazil>

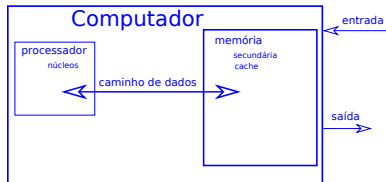
Hardware

- Arquitetura do computador

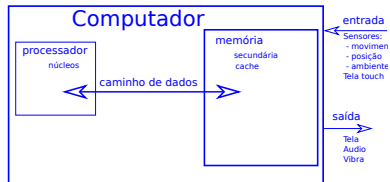


Hardware

● Arquitetura do computador



● Dispositivo móvel



Hardware

- Olhando para o *hardware*...

- ▶ Xiaomi Redmi 10C

- Chipset: Qualcomm SM6225 Snapdragon 680 4G
 - CPU: 8 cores - Qualcomm Kryo 265 CPU - 4x2.4 GHz e 4x1.9 GHz-64bits
 - GPU: Qualcomm Adreno 610 - suporte: OpenCL 2.0 / Vulkan 1.1 / OpenGL ES 3.2
 - Memória principal: 3/4 Gb
 - Memória secundária: 32/64/128 Gb

- ▶ Moto E-10

- Chipset: Qualcomm Snapdragon 632 SDM632
 - CPU: Octa-core, 1800 MHz, Kryo 250, 64-bit
 - GPU: Adreno 506
 - Memória principal: 2 Gb
 - Memória secundária: 32 Gb

Hardware

- Olhando para o *hardware*...

- ▶ Xiaomi Redmi 10C

- Chipset: Qualcomm SM6225 Snapdragon 680 4G
 - CPU: 8 cores - Qualcomm Kryo 265 CPU - 4x2.4 GHz e 4x1.9 GHz-64bits
 - GPU: Qualcomm Adreno 610 - suporte: OpenCL 2.0 / Vulkan 1.1 / OpenGL ES 3.2
 - Memória principal: 3/4 Gb
 - Memória secundária: 32/64/128 Gb

- ▶ Moto E-10

- Chipset: Qualcomm Snapdragon 632 SDM632
 - CPU: Octa-core, 1800 MHz, Kryo 250, 64-bit
 - GPU: Adreno 506
 - Memória principal: 2 Gb
 - Memória secundária: 32 Gb

É pouco ou é muito ?

Hardware

- Percepção do mundo:

Hardware

- Percepção do mundo:
 - ▶ Sensores:

Hardware

- Percepção do mundo:
 - ▶ Sensores:
 - **Movimento:** gravidade, aceleração linear, vetor de rotação, movimento significativo, contador de passos e detector de passos. E, acelerômetro e/ou do giroscópio são sempre baseados em hardware.

Hardware

- Percepção do mundo:
 - ▶ Sensores:
 - **Movimento**: gravidade, aceleração linear, vetor de rotação, movimento significativo, contador de passos e detector de passos. E, acelerômetro e/ou do giroscópio são sempre baseados em hardware.
 - **Posição**: sensor de campo geomagnético e o acelerômetro.

Hardware

- Percepção do mundo:

- ▶ Sensores:

- **Movimento:** gravidade, aceleração linear, vetor de rotação, movimento significativo, contador de passos e detector de passos. E, acelerômetro e/ou do giroscópio são sempre baseados em hardware.
 - **Posição:** sensor de campo geomagnético e o acelerômetro.
 - **Ambiente:** sensores que monitoraram algumas propriedades do ambiente, como: umidade relativa do ambiente, iluminância, pressão ambiente e temperatura ambiente.

Hardware

- Percepção do mundo:

- ▶ Sensores:

- **Movimento**: gravidade, aceleração linear, vetor de rotação, movimento significativo, contador de passos e detector de passos. E, acelerômetro e/ou do giroscópio são sempre baseados em hardware.
 - **Posição**: sensor de campo geomagnético e o acelerômetro.
 - **Ambiente**: sensores que monitoraram algumas propriedades do ambiente, como: umidade relativa do ambiente, iluminância, pressão ambiente e temperatura ambiente.

- ▶ Microfone

Hardware

- Percepção do mundo:

- ▶ Sensores:

- **Movimento:** gravidade, aceleração linear, vetor de rotação, movimento significativo, contador de passos e detector de passos. E, acelerômetro e/ou do giroscópio são sempre baseados em hardware.
 - **Posição:** sensor de campo geomagnético e o acelerômetro.
 - **Ambiente:** sensores que monitoraram algumas propriedades do ambiente, como: umidade relativa do ambiente, iluminação, pressão ambiente e temperatura ambiente.

- ▶ Microfone

- ▶ GPS

Hardware

- Percepção do mundo:

- ▶ Sensores:

- **Movimento:** gravidade, aceleração linear, vetor de rotação, movimento significativo, contador de passos e detector de passos. E, acelerômetro e/ou do giroscópio são sempre baseados em hardware.
 - **Posição:** sensor de campo geomagnético e o acelerômetro.
 - **Ambiente:** sensores que monitoraram algumas propriedades do ambiente, como: umidade relativa do ambiente, iluminância, pressão ambiente e temperatura ambiente.

- ▶ Microfone

- ▶ GPS

- ▶ Wi-Fi ←

- ▶ Bluetooth ←

Hardware

- Percepção do mundo:

- ▶ Sensores:

- **Movimento:** gravidade, aceleração linear, vetor de rotação, movimento significativo, contador de passos e detector de passos. E, acelerômetro e/ou do giroscópio são sempre baseados em hardware.
 - **Posição:** sensor de campo geomagnético e o acelerômetro.
 - **Ambiente:** sensores que monitoraram algumas propriedades do ambiente, como: umidade relativa do ambiente, iluminação, pressão ambiente e temperatura ambiente.

- ▶ Microfone

- ▶ GPS

- ▶ Wi-Fi ←

- ▶ Bluetooth ←

- ▶ Leitor de digital

Hardware

- Percepção do mundo:

- ▶ Sensores:

- **Movimento:** gravidade, aceleração linear, vetor de rotação, movimento significativo, contador de passos e detector de passos. E, acelerômetro e/ou do giroscópio são sempre baseados em hardware.
 - **Posição:** sensor de campo geomagnético e o acelerômetro.
 - **Ambiente:** sensores que monitoraram algumas propriedades do ambiente, como: umidade relativa do ambiente, iluminância, pressão ambiente e temperatura ambiente.

- ▶ Microfone

- ▶ GPS

- ▶ Wi-Fi ←

- ▶ Bluetooth ←

- ▶ Leitor de digital

Hardware

- Percepção do mundo:
 - ▶ Sensores:
 - **Movimento**: gravidade, aceleração linear, vetor de rotação, movimento significativo, contador de passos e detector de passos. E, acelerômetro e/ou do giroscópio são sempre baseados em hardware.
 - **Posição**: sensor de campo geomagnético e o acelerômetro.
 - **Ambiente**: sensores que monitoraram algumas propriedades do ambiente, como: umidade relativa do ambiente, iluminação, pressão ambiente e temperatura ambiente.
 - ▶ Microfone
 - ▶ GPS
 - ▶ Wi-Fi ←
 - ▶ Bluetooth ←
 - ▶ Leitor de digital
- E o *software* ? Sistema operacional ? Android (Linux)

O Sistema Operacional

- *Open source*
- É baseado no Linux



O Sistema Operacional

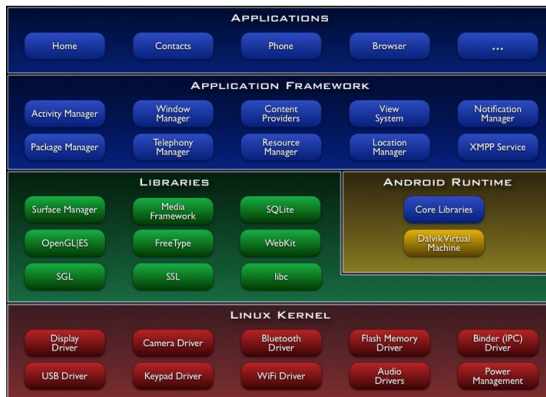
- *Open source*
- É baseado no Linux ← o que isso quer dizer ?



O Sistema Operacional

- *Open source*
- É baseado no Linux ← o que isso quer dizer ?
- É o kernel
- <https://source.android.com/docs/setup/about/android-13-release>

Arquitetura



Aplicação



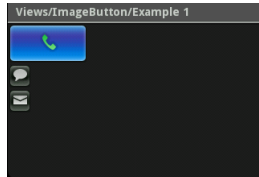
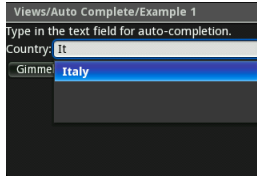
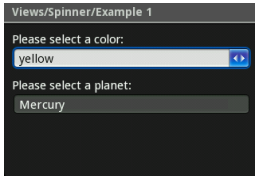
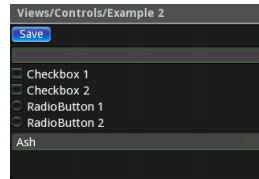
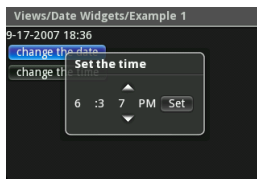
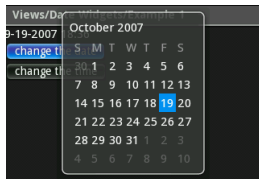
- cliente de email;
- calendário;
- mapas;
- navegador;
- contatos;
- outros apps...

Framework



- view;
- context provider;
- resource manager;
- notification manager;
- activity manager;

Framework



Biblioteca

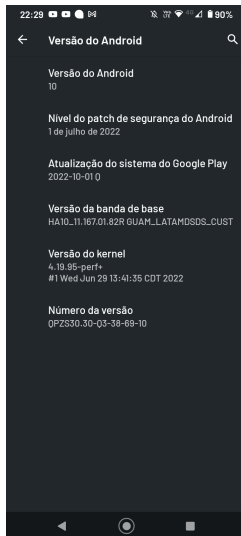


- System C Library;
- Media Libraries;
- Surface Manager;
- LibWebCore;
- SGL;
- 3D Libraries;
- FreeType;
- SQLite

- Versão do kernel
- Camada de abstração entre o software e hardware
- Serviços:
 - ▶ Segurança
 - ▶ Memória
 - ▶ Processos
 - ▶ Drivers
 - ▶ Protocolos de comunicação



O Sistema Operacional



- Configuração → Sobre o telefone → Versão do Android
 - kernel: 4.19.95-perf+

Aplicação

- Aplicação é formada por componentes fracamente acoplados
- Arquivo **manifest** define como os componentes interagem
 - ▶ Activities
 - ▶ Services
 - ▶ Content Providers
 - ▶ Intents
 - ▶ Broadcast Receivers
 - ▶ Widgets
 - ▶ Notifications

Aplicação

Activities

É o componente responsável para apresentação do aplicativo, é a interface do usuário. Normalmente, construída em 1 ou mais extensões da classe `Activity`. O layout da interface é construída com Fragments e Views e tem o objetivo de responder às ações do usuário. Equivalem ao form nas aplicações desktop.

Aplicação

Content Providers

Armazenamento de dados persistente compartilhável. É um recurso que permite que diferentes aplicativos compartilhem os mesmo dados. Basicamente, compartilhamento de dados/conteúdo entre APPs. Fotos, SMS e lista de contatos são alguns exemplos de armazenamento de dados persistente compartilhável nativos.

Aplicação

Services

São computações executadas sem interface gráfica, executados em segundo plano. Atualizando suas fontes de dados, *activities*, disparando notificações e realizando **comunicação entre processos**. Em geral, são usados para executar tarefas de longa duração ou que não exigem interação do usuário, tais como comunicação de dados e tarefas que precisam continuar mesmo quando as *activities* do aplicativo não estão ativos ou visíveis.

Aplicação

Intents

É uma poderosa estrutura de troca de mensagens entre aplicativos e/ou *activities*. **Uma das estruturas mais elegantes de comunicação entre processos.** Utilizada para iniciar *activities* e serviços. Em relação à comunicação, o processo que recebe a comunicação pode responder de forma explícita ou implícita.

Aplicação

Broadcast Receivers

Permite a resposta a um `Intents` de forma específica. Viabilizando aplicações orientadas a eventos.

Aplicação

Widgets

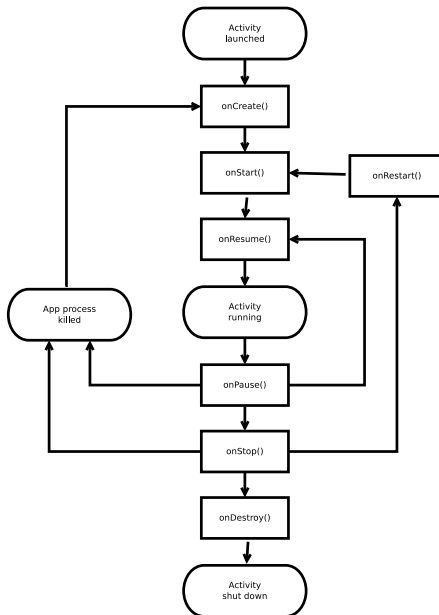
Conjunto de componentes de interface, responsáveis pela interação do usuário. Importante ressaltar que são componentes orientados a eventos e, além disso, cada componente responde a um ou mais eventos.

Aplicação

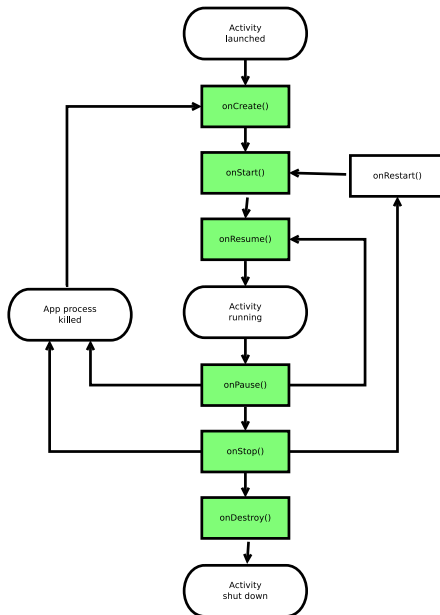
Notifications

As notificações permitem alertar aos usuários sobre eventos do aplicativo, focando ou interrompendo sua atividade atual. É um componente utilizado para obter a **atenção** do usuário quando o aplicativo não está ativo ou visível, como um serviço ou o receptor de uma transmissão. Exemplo: email que ao alerta o usuário reproduz sons, exibe ícones, rola o texto, vibra o aparelho.

Ciclo de vida da interface



Ciclo de vida da interface



Estados da aplicação - ciclo de vida

- Métodos:

- ▶ **onCreate ()** : executa a inicializado do aplicativo. O seu parâmetro define se está sendo criado para primeira vez;
- ▶ **onStart ()** : executado logo na sequência do método **onCreate ()** e ao final chama o método **onResume ()** ;
- ▶ **onResume ()** : após sua execução, chama novamente o método ;
- ▶ **onPause ()** : é executado sempre quando a *activity* não está mais em primeiro plano. Pode ou não indicar que a *activity* será destruída; no caso da *activity* se tornar completamente invisível, então o método **onStop ()** é chamado;
- ▶ **onStop ()** : quando a *activity* não estiver mais visível ao usuário. Neste caso, a *activity* é colocada no estado de **interrompido**;
- ▶ **onDestroy ()** : é um método chamado antes da *activity* ser destruída. O método é invocado nas situações:
 - a *activity* está sendo descartada completamente; ou
 - o sistema está destruindo temporariamente a atividade devido a uma mudança na configuração (como a rotação do dispositivo ou o modo de várias janelas)

Primeiros passos

- Alguns conhecimentos....

Primeiros passos

- Alguns conhecimentos....
 - ▶ SO
 - ▶ Lógica de programação
 - ▶ POO
 - ▶ Arquitetura de hardware
 - ▶ Redes
 - ▶ Engenharia de software
 - ▶ Estrutura de dados e armazenamento em disco

Primeiros passos

- Alguns conhecimentos....
 - ▶ SO
 - ▶ Lógica de programação
 - ▶ POO
 - ▶ Arquitetura de hardware
 - ▶ Redes
 - ▶ Engenharia de software
 - ▶ Estrutura de dados e armazenamento em disco
- baixar e instalar o SDK ?
<https://developer.android.com/studio>

Primeiros passos

- Alguns conhecimentos....
 - ▶ SO
 - ▶ Lógica de programação
 - ▶ POO
 - ▶ Arquitetura de hardware
 - ▶ Redes
 - ▶ Engenharia de software
 - ▶ Estrutura de dados e armazenamento em disco
- baixar e instalar o SDK ?
<https://developer.android.com/studio>
- o celular precisar ser configurado ? **modo desenvolvedor**

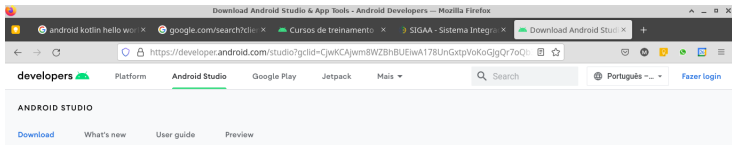
Primeiros passos

- Android SDK:
 - ▶ <https://developer.android.com/studio>

Primeiros passos

- Android SDK:

▶ <https://developer.android.com/studio>



Android Studio provides the fastest tools for building apps on every type of Android device.

[Download Android Studio](#)

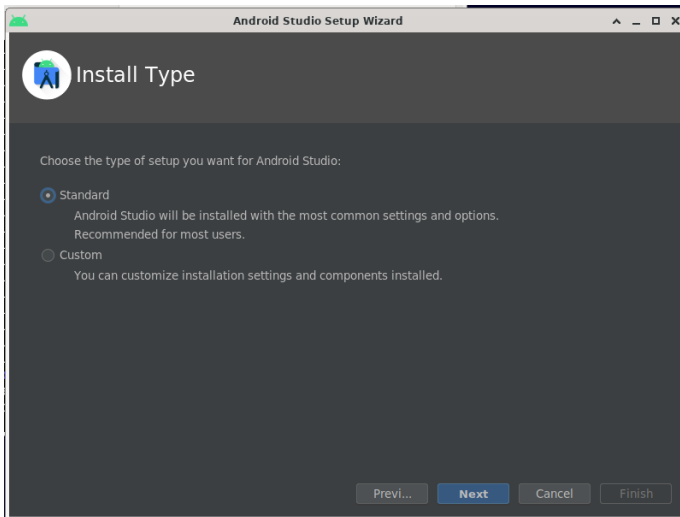
Android Studio Dolphin | 2021.3.1 for Linux 64-bit (930 MiB)

[Download options](#)

[Release notes](#)

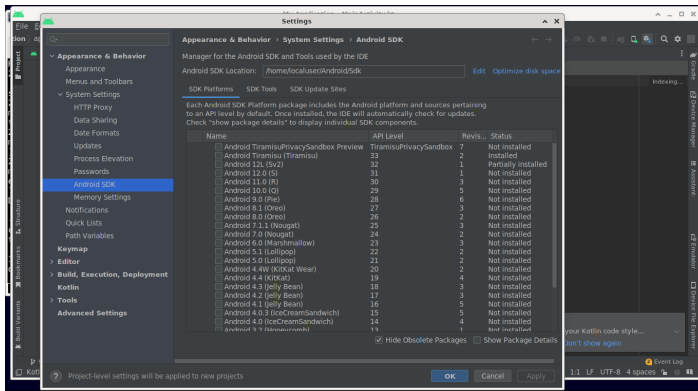
Primeiros passos

- A instalação:



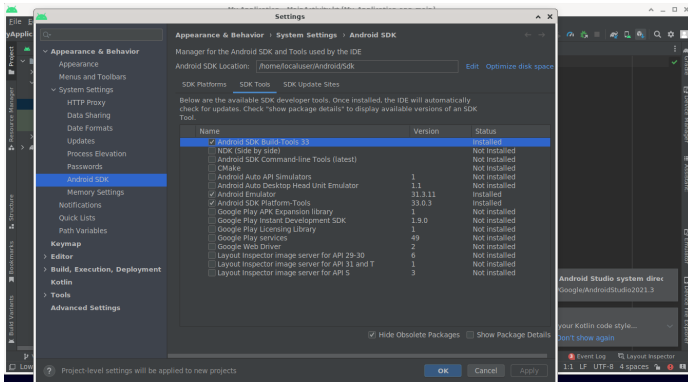
Primeiros passos

- A instalação:
 - ▶ SDK manager



Primeiros passos

- A instalação:
 - NDK manager



Primeiros passos

- *“Hello World!”*

Primeiros passos

- “*Hello World!*”
- Criar um projeto
 - ▶ Java - melhor documentado e mais conhecido
 - ▶ Kotlin - evolução em linguagem de desenvolvimento android
 - ▶ Flutter - portabilidade para o iOS

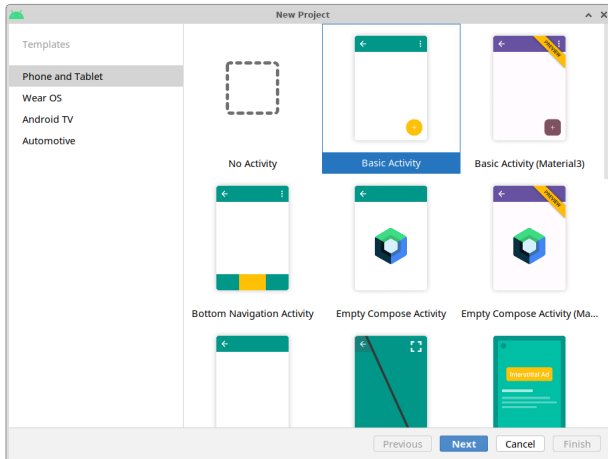


Primeiros passos

- “*Hello World!*”

Primeiros passos

- “Hello World!”

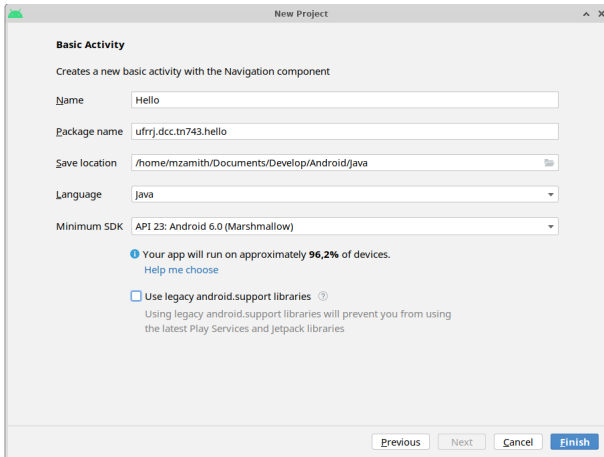


Primeiros passos

- “*Hello World!*”

Primeiros passos

- “Hello World!”



New Project

Basic Activity

Creates a new basic activity with the Navigation component

Name: Hello

Package name: ufrj.dcc.tn743.hello

Save location: /home/mzamith/Documents/Develop/Android/java

Language: Java

Minimum SDK: API 23: Android 6.0 (Marshmallow)

i Your app will run on approximately **96.2%** of devices.
[Help me choose](#)

☐ Use legacy android.support libraries ⓘ
Using legacy android.support libraries will prevent you from using the latest Play Services and Jetpack libraries

Previous Next Cancel Finish

Primeiros passos

- Estrutura do projeto
- Arquivos:
 - `AndroidManifest.xml`
 - `build.gradle`

AndroidManifest.xml

- Localizado no diretório raiz do projeto;
- descreve informações para compilação e execução do programa;
- quais informações são providas pelo arquivo:
 - ▶ componentes do aplicativo (como telas de interfaces);
 - ▶ recursos utilizados;
 - ▶ permissões do aplicativo;
 - ▶ recursos de hardware e software;
 - ▶ tema da interface;
 - ▶ diversas outras informações

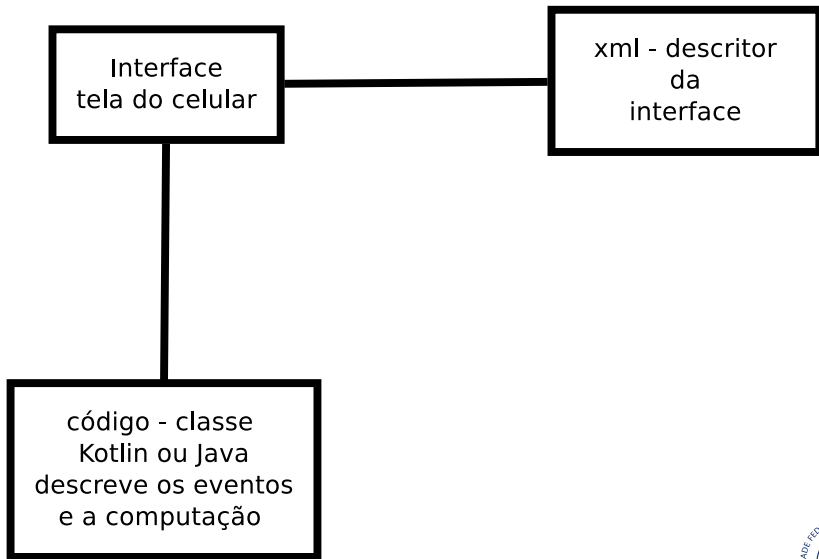
AndroidManifest.xml

● Exemplo:

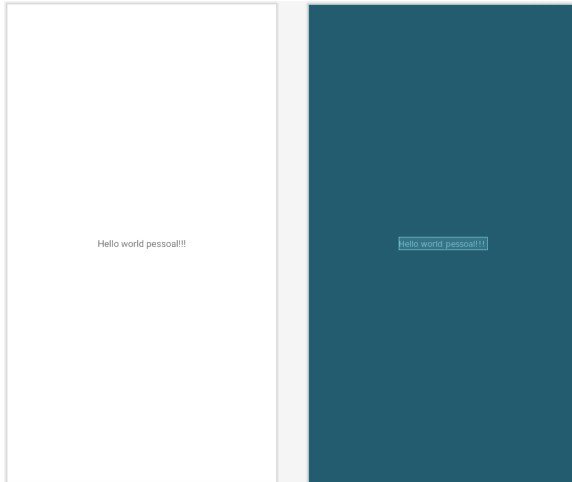
```
1  <?xml version="1.0" encoding="utf-8"?>
2      <manifest xmlns:android="http://schemas.android.com/apk
3      xmlns:tools="http://schemas.android.com/tools"
4      package="ufrj.dcc.tn743.hello">
5      <application
6          android:allowBackup="true"
7          android:dataExtractionRules="@xml/data_extraction_ru
8          android:fullBackupContent="@xml/backup_rules"
9          android:icon="@mipmap/ic_launcher"
10         android:label="@string/app_name"
11         android:roundIcon="@mipmap/ic_launcher_round"
12         android:supportsRtl="true"
13         android:theme="@style/Theme.Hello"
14         tools:targetApi="31">
15         <activity
16             android:name=".MainActivity"
17             android:exported="true">
18             <intent-filter>
19                 <action android:name="android.intent.action.MAIN"
20                 <category android:name="android.intent.category
21             </intent-filter>
22         </activity>
23     </application>
24
25     </manifest>
```



Estrutura de desenvolvimento



Interface - tela do celular



xml - descritor da interface

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      tools:context=".MainActivity">
9
10     <TextView
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:text="Hello world pessoal!!!"
14         app:layout_constraintBottom_toBottomOf="parent"
15         app:layout_constraintEnd_toEndOf="parent"
16         app:layout_constraintStart_toStartOf="parent"
17         app:layout_constraintTop_toTopOf="parent" />
18
19 </androidx.constraintlayout.widget.ConstraintLayout>
```

Código - descreve eventos e a computação

```
1 package ufrrj.dcc.tn743.hello
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5
6 class MainActivity : AppCompatActivity() {
7     override fun onCreate(savedInstanceState: Bundle?) {
8         super.onCreate(savedInstanceState)
9         setContentView(R.layout.activity_main)
10    }
11 }
```

