

# Aula IX - Desenvolvimento de Aplicativo I

Introdução à programação para Mobile

Java

WebServices

Aplicativos do Google Play

O SDK do Android

Android Studio

---

Fontes:

<https://developer.android.com/studio/intro/studio-config>

<https://developer.android.com/studio/run/managing-avds>

<https://developer.android.com/docs> ----> DOCUMENTAÇÃO

<https://distancia.qi.edu.br/mod/book/view.php?id=54357&chapterid=39091>

<https://blog.geekhunter.com.br/java-android/>

<https://www.androidpro.com.br/blog/java/programacao-android-java-basico-parte-01/>

---

## Java

Os aplicativos Android são desenvolvidos com Java – uma das linguagens de programação mais usadas e conhecidas do mundo. Essa linguagem foi uma escolha lógica para a plataforma Android, pois é poderosa, gratuita, de código-fonte aberto e além disso Java possui propriedades que só ela tem e já é conhecida por uma quantidade bastante elevada de desenvolvedores.

Os desenvolvedores Java com experiência podem tranquilamente migrar para o desenvolvimento com Android, usando as APIs (interfaces de programação de aplicativo) Android da Google e de outros fabricantes.

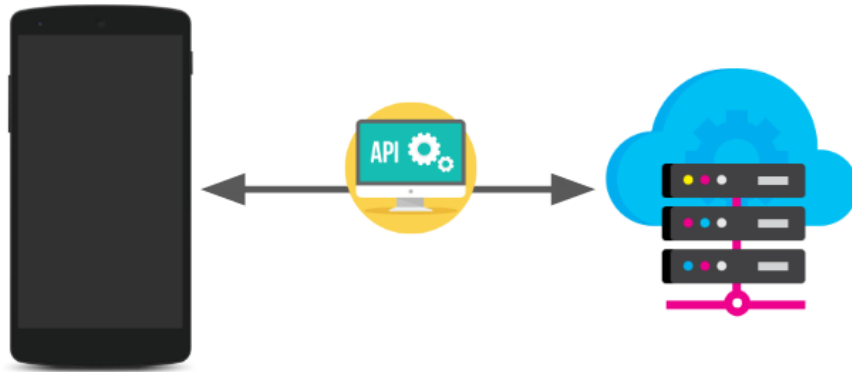
A linguagem Java é orientada a objetos e tem acesso às amplas bibliotecas de classe que podem auxiliar no desenvolvimento de aplicativos poderosos e robustos com extrema facilidade e rapidez.

## Web Services

Web services são componentes de software localizados em um computador, onde são acessados por um aplicativo (ou por outro componente de software) em outro computador por meio da Internet. Com eles, você pode criar *mashups*, permitindo o desenvolvimento de aplicativos rapidamente, combinando *web services* complementares, frequentemente de diferentes organizações e possivelmente com outras formas de *feeds* de informação.

Por exemplo, o 100 Destinations ([www.100destinations.co.uk](http://www.100destinations.co.uk)) combina as fotos e *tweets* do Twitter com os recursos de mapas do Google Maps, permitindo explorar países em todo o mundo por meio

de fotos tiradas por outras pessoas. O Programmableweb (<http://www.programmableweb.com/>) fornece um catálogo com mais de 9.400 APIs e 7.000 *mashups*, além de guias práticos e exemplos de código para você criar seus próprios *mashups*.



De acordo com o Programmableweb, as três APIs mais utilizadas para *mashups* são: Google Maps, Twitter e YouTube.



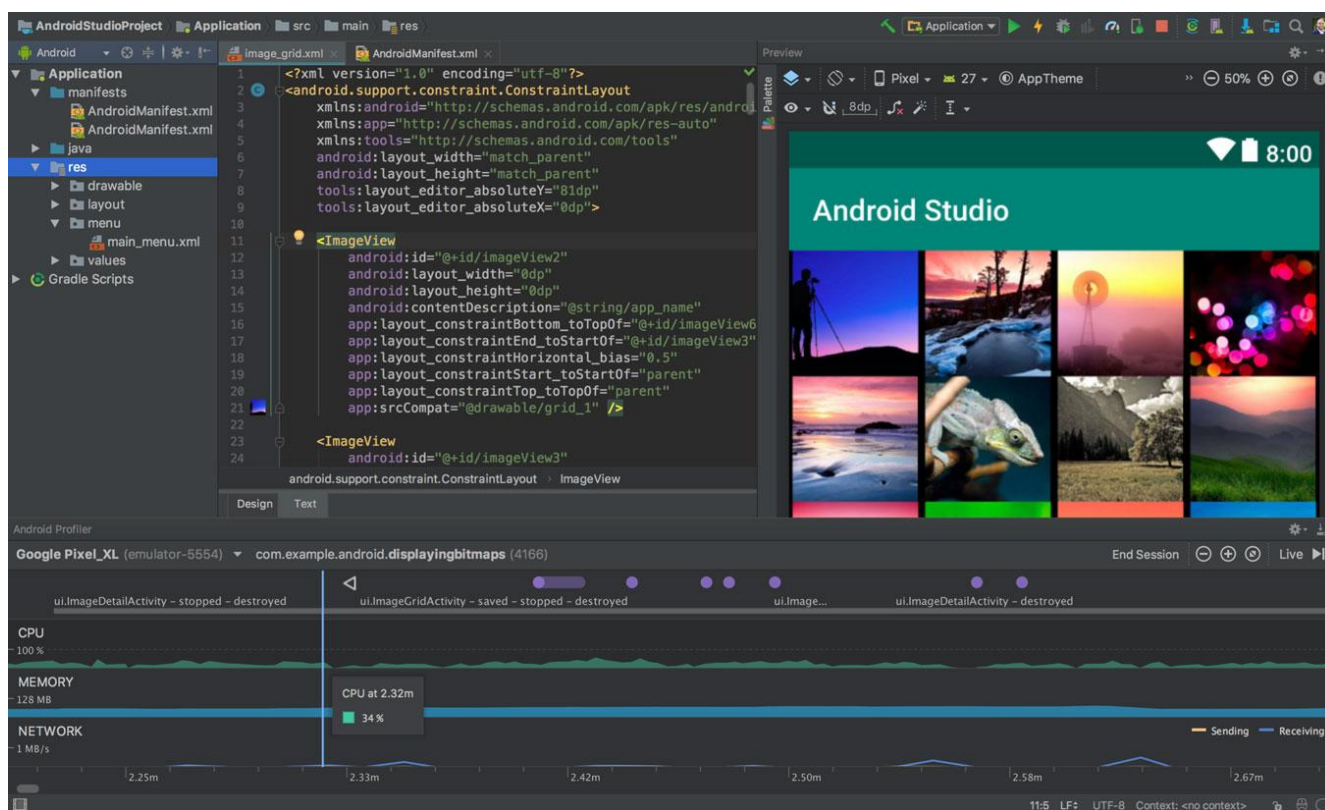
## Aplicativos do Google Play

É correto afirmar que existem mais de 1 milhão de aplicativos no Google Play e o número está crescendo rapidamente, talvez até possa se afirmar exponencialmente.

É possível baixar aplicativos por meio do aplicativo Play Store instalado no aparelho. É possível também conectar-se em sua conta no Google Play, no endereço <http://play.google.com>, com o navegador web e, então, especificar o dispositivo Android no qual deseja instalar o aplicativo. Ele será baixado por meio da conexão Wi-Fi ou 3G/ 4G do aparelho.

## O SDK do Android

O SDK (Software Development Kit) do Android fornece as ferramentas necessárias para construir aplicativos Android. Ele está disponível gratuitamente no site Android Developers (<https://developer.android.com/>).



## Android Studio

Android Studio é um ambiente de desenvolvimento integrado (IDE) para desenvolver para a plataforma Android. Teve seu lançamento em 16 de Maio de 2013 na conferência Google I/O.

Android Studio é disponibilizado para download gratuitamente sob a licença Apache 2.0. O Android Studio estava em estágio de acesso antecipado, alpha, desde a versão 0.1 em Maio de 2013, mudando então em estágio beta a partir da versão 0.8 que foi lançada em Junho de 2014. A primeira compilação estável sem os devidos "bugs" foi lançada em Dezembro de 2014, dando início à versão 1.0.

Tendo como base o software IntelliJ IDEA da JetBrains, Android Studio foi feito especificamente para o desenvolvimento para Android. Está disponível para download em Windows, Mac OS X e Linux,

substituindo o Eclipse Android Development Tools (ADT) como a IDE primária do Google de desenvolvimento para Android.

- Android Studio é um ambiente de desenvolvimento integrado (IDE) com o foco no desenvolvimento para a plataforma Android.
- Ele nos oferece uma interface amigável, de fácil utilização e com recursos que aceleram o processo de desenvolvimento.
- Entre esses recursos estão um editor gráfico para criar as telas da aplicação e a integração com os emuladores do Android.

## Características:

- Suporte para compilações baseadas em Gradle.
- Refatoração específica para Android e reparações rápidas.
- Ferramentas de Lint para capturar performance, usabilidade, compatibilidade de versão e outros problemas.
- Integração com ProGuard e capacidade de assinatura de aplicativo.
- Um assistente baseado em predefinições com designs e componentes comuns de Android.
- Um rico editor de layout que permite que usuários arrastem componentes de interface de usuário, opção de pré-visualizar layouts em várias configurações de tela.
- Suporte para a criação de apps para Android Wear.
- Suporte nativo para a Google Cloud Platform, permitindo a integração com Google CloudMessaging e AppEngine.

## Arquitetura de trabalho do Android Studio

É importante conhecer um pouco mais da arquitetura do Sistema Operacional Android e onde, nessa arquitetura, os aplicativos desenvolvidos se encaixam, para então entender como fazer o uso dos recursos do sistema durante o desenvolvimento.



## Componentes básicos de um aplicativo

Existem 4 componentes básicos de um aplicativo Android:  
**Activity, Service, Content Provider e Broadcast.**

**As funções desses componentes são:**

**Activity:** é um componente que disponibiliza uma tela para que o usuário possa interagir. Sempre que o usuário visualiza qualquer coisa na tela do dispositivo, esse funcionamento é controlado por uma Activity.

**Services:** é um componente que permite que o aplicativo tenha acesso ao processamento sem que esteja diretamente vinculado à uma tela. Por exemplo, quando um aplicativo de música não está em primeiro plano e é necessário que a música continue sendo reproduzida, todo o processamento deve ser realizado dentro de um serviço.

**Content Provider:** devida a arquitetura do Android, que impede que um aplicativo tenha acesso às informações de outro aplicativo, a troca de informações, quando necessária, ocorre através deste componente.

**Broadcast:** Através deste componente, um aplicativo pode se inscrever para eventos do Android ou de outro aplicativo ou também publicar eventos para outros aplicativos. Assim é possível reagir aos eventos que ocorrem. Por exemplo, se um aplicativo precisa ser acionado sempre que uma chamada telefônica for recebida, é através de um BroadcastReceiver que ele será acionado.

## Versões do Android

O Android, assim como a maioria dos softwares, evolui com o passar do tempo. Nas evoluções vários recursos são adicionados, alguns retirados e também há reestruturações arquiteturais para deixar o sistema mais fluído, com menos consumo de recursos, entre outras mudanças.

Para os desenvolvedores é muito importante estar atento a cada nova versão do Android, pois os recursos disponíveis para utilização nos aplicativos também mudam conforme a versão que se deseja atender.

Para a configuração dos projetos internamente, utilizamos o número da versão da API correspondente à versão do sistema operacional, por isso também é importante ter conhecimento das versões da API.



Para você conhecer a história do Android, no link abaixo poderá acessar as principais características do sistema operacional.

### [História do Android](#)

