

APLICATIVOS PARA ANDROID

UM PASSO A PASSO
DA CRIAÇÃO ATÉ A LOJA



Erik leger Dobrychtop

Aplicativos para Android

Um Passo a Passo da Criação Até a Loja



**editora
VIENA**

1^a Edição
Bauru/SP
Editora Viena
2022

Sumário

| | |
|--|------------|
| <i>Lista de Siglas e Abreviaturas</i> | 13 |
| 1. Introdução a Programação | 15 |
| 1.1. Linguagens de Programação..... | 18 |
| 1.2. Introdução a Sistemas Operacionais..... | 18 |
| 2. Introdução a Android | 23 |
| 2.1. Características do Android e Versões..... | 27 |
| 2.2. Arquitetura do Android | 27 |
| 2.3. SDK para Android..... | 28 |
| 3. Introdução a Kotlin..... | 31 |
| 3.1. Porque Kotlin? | 34 |
| 3.2. Sintaxe e classes, como funciona a linguagem..... | 35 |
| 3.3. Variáveis..... | 36 |
| 3.4. Tipos de Variáveis..... | 39 |
| 3.5. Arrays..... | 40 |
| 3.6. Funções..... | 42 |
| 4. Introdução a Kotlin..... | 47 |
| 4.1. Construtores | 53 |
| 4.2. Herança..... | 55 |
| 4.3. Modificadores de Acesso | 58 |
| 5. Instalação Java..... | 65 |
| 5.1. Instalação Android Studio | 69 |
| 5.2. Configuração Android Studio..... | 70 |
| 6. O Primeiro Aplicativo | 73 |
| 6.1. Interface Android Studio..... | 77 |
| 6.2. Utilizando Texto e Botões..... | 78 |
| 6.3. Gerador de Números | 84 |
| 7. Navegação..... | 89 |
| 7.1. Navegando com Dados | 97 |
| 7.2. Layout com XML..... | 99 |
| 8. Criar, Ler, Atualizar e Deletar | 103 |
| 9. Recursos Básicos..... | 115 |
| 9.1. Recurso Câmera..... | 117 |
| 9.2. Recurso Localização | 120 |
| 10. Publicação do Aplicativo | 131 |
| Referências..... | 139 |
| Glossário..... | 141 |

Lista de Siglas e Abreviaturas

| | |
|------|---|
| ↪ | Indica que a linha de comando faz parte da linha acima, ou seja, as duas linhas são uma única linha de comando. |
| API | <i>Application Programming Interface.</i> |
| APK | <i>Android Package Kit.</i> |
| APP | <i>Aplicativo.</i> |
| CLI | <i>Command Line Interface.</i> |
| CPU | <i>Central Processing Unit.</i> |
| CRUD | <i>Create, Read, Update, Delete.</i> |
| CSS | <i>Cascading Style Sheets.</i> |
| DOM | <i>Document Object Model.</i> |
| FM | <i>Frequência Modulada.</i> |
| GPS | <i>Global Positioning System.</i> |
| HSPA | <i>High Speed Packet Access.</i> |
| HTML | <i>HyperText Markup Language.</i> |
| IBM | <i>International Business Machines.</i> |
| IDE | <i>Integrated Development Environment.</i> |
| RAM | <i>Random Access Memory.</i> |
| SDK | <i>Software Development Kit.</i> |
| SSD | <i>Solid State Drive.</i> |
| UI | <i>User Interface.</i> |
| URL | <i>Uniform Resource Locator.</i> |
| UX | <i>User Experience.s</i> |

1

Introdução a Programação

- 1.1. Linguagens de Programação
- 1.2. Introdução a Sistemas Operacionais

1. Introdução a Programação

A programação é basicamente a implementação da lógica para facilitar procedimentos e funcionalidades específicas.

Uma lógica pode ser considerada um passo a passo, por exemplo, uma receita de bolo, em que se indicam ingredientes, qual proporção adicionar, como executar as tarefas para chegar ao resultado final. Porém a lógica de que iremos tratar relaciona-se a computadores, em que um desenvolvedor tem a função de instruir o processador do computador a executar determinadas tarefas; resumindo, são conjuntos de passos finitos e organizados que, quando executados, resolvem um determinado problema.



Atualmente é fácil observar a programação com aplicativos, sistemas complexos e multifártios. Porém no princípio da programação eram somente algoritmos, que, por si, tinham seu propósito, eram especializados para sua utilidade.

Por exemplo, a primeira programadora a ser documentada é Ada Lovelace, que relatou a atividade de uma máquina analítica de Charles Babbage, a qual poderia ser programada e executar diversos comandos, utilizando como base cartões perfurados; no entanto, a máquina não conseguiu ser concluída.

Os primeiros computadores eletrônicos foram construídos a partir de 1940, e, mesmo quando começou a programação computacional, o desenvolvedor havia de utilizar linguagem de máquina, diretamente com linguagem binária ou hexadecimal.

Em seguida, foi criada uma linguagem de programação, denominada Assembly. A linguagem Assembly é reconhecida como uma linguagem de baixo nível, que segue as características da arquitetura de um computador, assim todo código possui um significado específico para o processador, facilitando a comunicação entre o desenvolvimento em Assembly e o processador.

Um exemplo de desenvolvimento com a linguagem de programação de baixo nível Assembly pode ser um simples hello world, ou até mesmo um complexo código para um controlador comandar algum processo industrial.

Dentro de linguagem de baixo nível, há a primeira geração e a segunda geração. Na primeira geração, o desenvolvedor havia de utilizar diretamente o código da máquina, utilizando o código hexadecimal.

A segunda geração é a própria linguagem Assembly citada anteriormente; nesse caso, o desenvolvedor não utilizará linguagem provinda do processador, projetadas para minimizar as dificuldades da programação em programação binária, facilitando, assim, o desenvolvimento.

Desde Assembly houve uma constante evolução até linguagens de alto nível, que possuem um nível de abstração relativamente elevado. O desenvolvimento desse tipo de linguagens ficou de modo mais distante à arquitetura do computador e mais próximo à linguagem humana.

Atualmente diversas linguagens de programação são baseadas em inglês, consistindo em declarações inglesas; por exemplo, na linguagem Java, caso o desenvolvedor desejar verificar a condição de alguma informação, pode usar o comando **If**, que significa “se” em inglês. Desse modo, o desenvolvedor não tem necessidade de conhecer propriedades do processador, ausentando de conhecimentos de baixo nível.

Um exemplo simples de um aparelho que utiliza programação de alto nível são nossos aparelhos celulares, ou até mesmo os aplicativos que são instalados nos aparelhos.

Como foi explicado anteriormente, um algoritmo pode ser simplesmente um passo a passo, uma sequência de operações. Observe a seguir um exemplo simplificado de um ato que realizamos diariamente, como lavar as mãos:

1. Ligar a água;
2. Molhar as mãos;
3. Pegar o sabonete;
4. Esfregar o sabonete nas mãos;
5. Guardar o sabonete;
6. Esfregar as mãos uma na outra;
7. Enxaguar as mãos;
8. Desligar a água;
9. Secar as mãos.

O desenvolvedor irá realizar procedimentos semelhantes ao exemplo anterior, porém utilizando a linguagem de programação, dando instruções como para rodar algum áudio, vídeo, até mesmo realizar integrações com redes sociais.

1.1. Linguagens de Programação

Uma linguagem de programação é um conjunto de regras que fornecem um percurso de quais operações o computador tem de executar. Basicamente são enviados comandos para um computador, por meio da linguagem de programação. Existem diversas linguagens de programação com fins específicos, no caso, Kotlin é utilizado para criar aplicativos, assim, serão escritos comandos para que o celular execute.

Cada linguagem de programação possui um conjunto diferente de regras de sintaxe, por exemplo, existem diversas linguagens de alto nível, como C, C++, Go, PHP e até a linguagem Kotlin, que será utilizada no livro.

1.2. Introdução a Sistemas Operacionais

Sistema Operacional (Operating System) é a parte central do aparelho que se encarrega do bom funcionamento dos recursos, podendo ser considerado um programa ou um conjunto de programas que permitem que o usuário tenha a melhor experiência ao utilizar as funcionalidades do aparelho.

Mendonça e Martini (2009) citam que é função do Sistema Operacional (SO) detectar e controlar o hardware, mais especificamente é ele que possibilita uma interface amigável entre os programas e o hardware. “Um sistema operacional fornece o ambiente dentro do qual os programas são executados” (SILBERSCHATZ, GAVIN, GAGNE, 2008, p. 29).

Entre os principais sistemas operacionais estão o Linux, o Windows e o Mac. Já em relação a dispositivos móveis os principais são Android e iOS, sendo que o primeiro apresenta a singularidade de ser considerado código aberto, o que permite que os aplicativos desenvolvidos possam ser mais facilmente executados.

Já o iOS, criado pela empresa Apple, apenas pode ser executado em aparelhos desenvolvidos pela própria empresa, o que restringe um pouco mais a atuação dos desenvolvedores; contudo, isso não é um fator limitante para que os aplicativos continuem sendo criados, pois cresce cada vez mais o número de usuários que utilizam aparelhos com esse sistema.

Como o intuito do presente livro é focar na criação de aplicativos para sistema Android, no próximo capítulo será detalhado um pouco mais sobre esse sistema, quais são suas características, suas versões e demais informações que possam expandir os conhecimentos dos desenvolvedores, para que estes aprimorem suas criações.

Exercícios

- ## 1. O que é programação?

2. Quando começou a programação computacional, o desenvolvedor deveria utilizar linguagem de máquina diretamente com qual linguagem?

3. Cite uma linguagem de baixo nível da segunda geração.

4. O que é uma linguagem de programação?

5. O que é um Sistema Operacional?

Anotações



Anotações



2

Introdução ao Android

- 2.1. Características do Android e Versões
- 2.2. Arquitetura do Android
- 2.3. SDK para Android

2. Introdução ao Android

Android é um sistema operacional e uma plataforma de programação mantida pela empresa Google, atualmente utilizada em diversos dispositivos móveis, como telefones celulares, tablets e uma gama crescente de dispositivos.

Android foi criado em 2003 pela Android Inc. com o objetivo de ser um sistema operacional para aplicativos móveis; é um projeto de código aberto, possibilitando que qualquer um consiga consultar e examinar o código.

O sistema operacional Android é baseado no Kernel do Linux e a estrutura é embasada no sistema operacional GNU/Linux.

Após a evolução do mercado, a Google aderiu ao projeto Android em 2005 e, em meados de 2008, saiu o primeiro dispositivo celular com o sistema operacional Android.

Como sistema operacional, o Android é um intermediador entre o usuário e o dispositivo. Quando o usuário tira uma foto, é fornecido um botão que, ao ser selecionado, informa ao dispositivo o que fazer. Ou até mesmo quando algum jogo é acionado, o Android informa quais botões e movimentos estão sendo realizados.



A interface fornecida pelo Android é realmente rica em detalhes, fornece para o usuário uma tela sensível ao toque para que seja realizada a interação com aplicativos, e também existe a possibilidade de manipulação direta, assim o usuário pode utilizar movimentos para manipular objetos na tela, como passar, tocar, entre outros. Também possui um teclado virtual para que haja entrada de texto; do mesmo modo há a possibilidade de teclados físicos ou controles conectados por Bluetooth.

Sua página inicial conta com várias páginas, entre as quais o usuário pode navegar com apenas o gesto de deslizar para a frente ou para trás. Utilizando os recursos de movimentação entre as telas, é possível o usuário também observar a fluidez do Android, que foi projetado para proporcionar resposta instantânea.

Com diversos aplicativos, também é possível visualizar componentes de uma interface gráfica, mais conhecidos como widgets, que têm a funcionalidade de exibir conteúdos automaticamente, como temperatura, preço de ações ou até o preço do dólar.

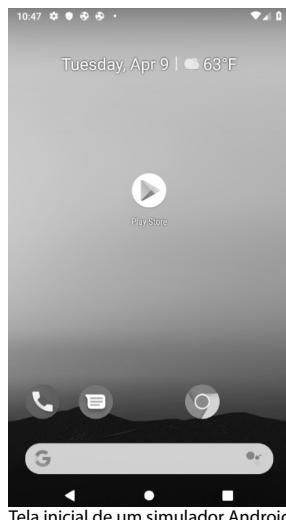
O sistema operacional também é capaz de reproduzir conteúdo multimídia, como animações, músicas e vídeo.

Quando é retratado sobre hardware e sensores, o Android consegue detectar a rotação da tela, pois muitos aparelhos móveis contam com acelerômetros, giroscópios e sensores de proximidade.

Se o usuário girar o celular e posicionar o aparelho de lado, a rotação irá de retrato para paisagem, assim o usuário terá uma visão mais extensa; por exemplo, caso o usuário jogar um jogo de corrida, também é possível direcionar o automóvel com o simples movimento de girar o celular.

Caso o usuário observar na parte superior da tela, existem status sendo informados, como horário, conectividade do wi-fi ou redes móveis, bateria e até sinal da telefonia móvel.

Observe a seguir uma imagem com a tela inicial do Android:



Tela inicial de um simulador Android.

A linguagem de programação Java foi criada por uma empresa chamada Sun Microsystems e, após um tempo de corporação, foi adquirida pela empresa Oracle. O Java é a base da plataforma Android.

Caso o desenvolvedor já tenha criado alguns aplicativos para Android ou até mesmo já tenha questionado como são programados, eles foram desenvolvidos utilizando a linguagem de programação Java. O WhatsApp, por exemplo, foi desenvolvido utilizando Java, assim, caso o desenvolvedor tenha o objetivo de trabalhar nessa empresa, é fundamental conhecer a tecnologia.

O interessante do desenvolvimento Android é a sua comunidade, que possui muitas bibliotecas e recursos criados pela própria.

A linguagem de programação Kotlin, criada em meados de 2010 pela empresa Jetbrains, famosa por criar ferramentas como IntelliJ, PhpStorm, o Android Studio que é utilizado para desenvolver para Android, foi baseada nas ferramentas da Jetbrains.

No Google IO de 2017, foi anunciada a linguagem de programação Kotlin como mais uma linguagem suportada para desenvolvimento Android. A linguagem é interoperável com Java, é possível escrever Java dentro de código Kotlin e vice-versa. Por exemplo, é possível ter código Kotlin que utilizem recursos da linguagem Java. Assim é facilitado realizar a migração de uma aplicação Java para código Kotlin,

e adicionar suporte Kotlin. Outro problema que Kotlin também tem como solução, é o fato de todos os recursos e bibliotecas que haviam sido escritos em Java poderem ser reutilizados.

Para desenvolver para a plataforma Android, é disponibilizado um kit de desenvolvimento para programar e criar aplicativos desejados; além disso, a Google disponibiliza uma loja para distribuição de aplicações.

2.1. Características do Android e Versões

O sistema operacional Android possui atualizações categorizadas com versões. Observe uma tabela com as últimas versões do Android:

| Nome | Ano de lançamento |
|-------------|-------------------|
| Lollipop | 2014 |
| Marshmallow | 2015 |
| Nougat | 2016 |
| Oreo | 2017 |
| Pie | 2018 |
| Android 10 | 2019 |
| Android 11 | 2020 |
| Android 12 | 2021 |

Cada versão vem com uma atualização diferente, sendo realizada melhoria em segurança, menu, entre outros. Por exemplo, observe algumas das atualizações do Android Pie:

- Nova interface de usuário para o menu de configurações rápidas.
- O relógio foi movido para a esquerda da barra de notificação.
- O “dock” agora tem um fundo semitransparente.

2.2. Arquitetura do Android

É necessário que o desenvolvedor conheça a arquitetura do sistema operacional Android, pois em certos desenvolvimentos pode ocorrer de a arquitetura do sistema ser diferente do planejado.

No Android existem coleções de bibliotecas, que auxiliam o desenvolvedor a realizar a criação, e esses componentes fazem parte da arquitetura do Android.

Por exemplo, um aplicativo possui um ciclo, quando o aplicativo é minimizado, é possível gerenciar as ações nesse momento, definindo quais ações devem ser tomadas, gerenciando o ciclo de vida de um aplicativo.

Quando o desenvolvedor desejar manipular as atividades de uma aplicação Kotlin, é necessário manipular as atividades. Nesse caso, existem métodos implementados, caso a atividade seja criada pela primeira vez, é chamado o método OnCreate; quando a atividade está se tornando visível para o usuário, é realizado o método OnStart, ou quando a atividade estiver na condição OnPause, essas condições não são visíveis para o usuário. Quando uma atividade estiver na condição OnPause e voltar a iniciar a atividade, o método chamado é o OnStart.

No momento em que a atividade começar a interagir com o usuário, é utilizado o método `OnResume`. É notável que o método não indica que a aplicação realize uma análise sobre o estado da aplicação, e nessa etapa o desenvolvedor pode encontrar problemas. O evento é apenas uma comunicação pelo sistema operacional que retomou as atividades.

No caso com os métodos `OnPause` e `OnStop` há alguns problemas, como também é chamado para casos como alternar entre aplicativos, tela de bloqueio, entre outros. Quando a atividade não é visível para o usuário, é pelo método `OnStop`. Já o método `OnRestart` é chamado após a atividade parar, antes do início.

Quando a atividade está prestes a ser destruída, é convocado o método `OnDestroy`, que normalmente é um dos menos utilizados.

2.3. SDK para Android

Uma ferramenta necessária para desenvolver aplicativos é um kit de desenvolvimento de software, conhecido como `SDK`. O kit possui um conjunto de ferramentas de desenvolvimento, dentre as quais existem bibliotecas de software com códigos, depuradores, emuladores de plataforma Android, documentação, tutoriais e código de amostra.



Para desenvolver, é necessário o desenvolvedor conhecer `Kotlin` e `XML` (Extensible Markup Language), pois com a linguagem `Kotlin` o desenvolvedor cria um arquivo binário que pode ser executado em celulares, tablets, relógios, carros, entre outros dispositivos; já para realizar a separação de interfaces do usuário, é utilizado o `XML`, em que são desenvolvidos conjuntos corretos de acordo com o aparelho.

A Google oferece aos programadores um ambiente de desenvolvimento integrado, conhecido como `IDE`, e denominado `Android Studio`, que possui recursos avançados para complementar a criação de aplicativos; a ferramenta possui teste, emulador, debug, entre outros recursos. O Sistema Operacional `Android` disponibiliza para os programadores uma arquitetura de desenvolvimento extraordinária.

Exercícios

1. O que é o Android?

2. O Android é baseado no Kernel de qual Sistema Operacional?

3. Qual é a linguagem oficial do Android?

4. Cite três nomes de versões de Android.

5. O que é SDK?

Anotações

