

DESENVOLVIMENTO DE UM APLICATIVO PARA AUXÍLIO NO APRENDIZADO DOS VERBOS IRREGULARES EM INGLÊS

Valter Doná¹
Maria Helena Barriviera e Silva²

Resumo. Este artigo descreve a elaboração de um aplicativo para *Android* que busca ser uma ferramenta adicional no processo de memorização de verbos irregulares do idioma inglês. A ideia do aplicativo surgiu pelo fato dos *smartphones* estarem tão presentes no dia a dia das pessoas e no intuito das mesmas quererem servir-se do aplicativo por alguns minutos do seu dia, dentro de um ônibus a caminho do trabalho, por exemplo. Na elaboração do aplicativo foram utilizados o Ambiente de Desenvolvimento Integrado (IDE) *Android Studio* e a linguagem Java.

Palavras-chave: *Android*. Aplicativo. Jogo. Verbos irregulares em inglês.

Abstract. *This paper describes the development of an Android application that seeks to be an additional tool in the process of memorizing irregular English verbs. The purpose of the application came from the fact that smartphones are so present in people's daily lives and in order for them to want to use the application for a few minutes of their day, on a bus on the way to work, for example. In the development of the application were used the Integrated Development Environment (IDE) Android Studio and the Java language.*

Keywords: *Android. App. Game. Irregular verbs in English.*

1 INTRODUÇÃO

O *smartphone* faz parte do dia a dia das pessoas. É cada vez mais incomum encontrar alguém que não o possui; muitos possuem até mais de um aparelho. Na década de 1970 foi criado o primeiro telefone verdadeiramente móvel pesando cerca de

¹ Graduado em Tecnologia em Processamento de Dados na Faculdade de Tecnologia de Ourinhos – Fatec Ourinhos-SP – e atualmente cursando a pós-graduação *Lato Sensu* em Desenvolvimento de Sistemas para *Web* e Dispositivos Móveis na Universidade de Marília – Unimar. *E-mail:* valter.donah@gmail.com

² Bacharel em Ciência da Computação pela Universidade de Marília – Unimar –, especialista em Sistemas de Informação pela Universidade Federal de São Carlos – UFSCar –, e mestre em Ciência da Computação, subárea Inteligência Artificial pela Universidade de São Paulo, *campus* São Carlos – USP São Carlos. Atualmente exerce a função de professora junto à Universidade de Marília e na Fatec Graça, no curso de Tecnologia em Análise e Desenvolvimento de Sistemas. É coordenadora do curso de pós-graduação em Desenvolvimento de Sistemas para *Web* e Dispositivos Móveis na Universidade de Marília – Unimar. *E-mail:* mhbs-ftp@unimar.br

um quilo, já que em 1946 havia um que pesava 40 quilos. Atualmente, a função de “telefone” não é mais tão importante, uma vez que o aparelho é muito mais utilizado para outras atividades, como: acessar *e-mail*, navegar na *web*, enviar mensagens, jogar e acessar redes sociais (TANENBAUM, 2016).

O sistema operacional mais utilizado nos dias atuais em *smartphones* é o *Android*, da *Google*. Ele é baseado no núcleo *Linux* e está livremente disponível para fabricantes de dispositivos móveis usarem em seus produtos, por ser uma plataforma de código aberto é customizável para uma série de dispositivos (TANENBAUM, 2016).

Entre as tantas funcionalidades que os aplicativos para *smartphones* proporcionam atualmente, pode-se citar o auxílio no aprendizado de assuntos diversos. Assim, é possível utilizá-lo, por exemplo, para auxiliar no aprendizado de verbos irregulares da língua inglesa. Existem atualmente diversos aplicativos que auxiliam no aprendizado de idiomas. Em Positivo (2017) são listados 7 aplicativos utilizados no auxílio ao aprendizado da língua inglesa, a saber: *Duolingo*; *Hello English*; *How To Say*; *LinguaLeo*; *Busuu*; *Hi Native*; e *Unlock Your Brain*.

Provavelmente o mais famoso deles é o *Duolingo*, considerado um dos melhores aplicativos para quem quer aprender inglês de forma leve, divertida e de acordo com o nível de habilidade pessoal. Ele é gratuito e focado em vocabulário, escrita e compreensão oral, aliando lições relacionadas aos diversos momentos do dia a dia (POSITIVO, 2017).

Seguindo o raciocínio dos aplicativos mencionados, este artigo objetiva descrever o desenvolvimento de um aplicativo simples, mais precisamente de um jogo, o qual provavelmente a maioria das pessoas já teve a oportunidade de brincar, que é o jogo da forca. No aplicativo o objetivo é “trabalhar” com verbos irregulares da língua inglesa.

Diferente dos verbos regulares que são mais fáceis de entender, pois seguem um padrão de conjugação de acordo com o tempo escolhido, os verbos irregulares não seguem regras, constituindo-se em dificuldade para quem está aprendendo o idioma inglês (OLIVEIRA, 2015).

Oliveira (2015) afirma que “o ideal é praticar sempre e colocar o inglês no seu dia a dia para se acostumar com os diferentes verbos e conjugações”. Assim, o intuito do aplicativo/jogo é ser uma ferramenta adicional para ajudar no aprendizado.

Para desenvolver o jogo foi utilizado o Ambiente de Desenvolvimento Integrado (IDE) *Android Studio* e a linguagem de programação *Java*. As listas de verbos irregulares utilizadas no aplicativo podem ser consultadas em Só Língua Inglesa (2017).

Segundo relatório da *Strategy Analytics* (apud DEITEL, 2015, p. 3), em outubro de 2013 o *Android* tinha 81,3% da fatia de mercado global de *smartphones*, comparados com 13,4% da *Apple*, 4,1% da *Microsoft* e 1% do *Blackberry*.

No Brasil, segundo dados divulgados pela consultoria IDC, 95,5% dos aparelhos comercializados entre julho e setembro de 2016 rodavam o sistema operacional *Android*. Os outros 4,5% eram *iPhones* (HIGA, 2016).

2 DESENVOLVIMENTO DO APLICATIVO

Como já mencionado anteriormente, no desenvolvimento do aplicativo foi utilizado o IDE *Android Studio 2.3.3* e a linguagem de programação *Java*.

Na Figura 1 é apresentado o diagrama de classes do aplicativo. O diagrama de classes fornece uma visão estática ou estrutural de um sistema, mas não mostra a natureza dinâmica das comunicações entre os objetos das classes no diagrama. Os elementos principais são caixas, e cada caixa é dividida em partes horizontais. A parte superior contém o nome da classe (*Forca* e *ForcaActivity*). A parte do meio lista os atributos da classe (como *palavraSorteada* e *categoria*). Um atributo refere-se a alguma coisa que um objeto daquela classe sabe ou pode fornecer o tempo todo. Atributos são usualmente implementados como campos da classe. A terceira parte do diagrama de classes contém as operações ou comportamentos da classe (como *montarResposta()* e *trocarImagem()*). Uma operação refere-se ao que os objetos da classe podem fazer e usualmente é implementada como um método da classe (PRESSMAN, 2011).

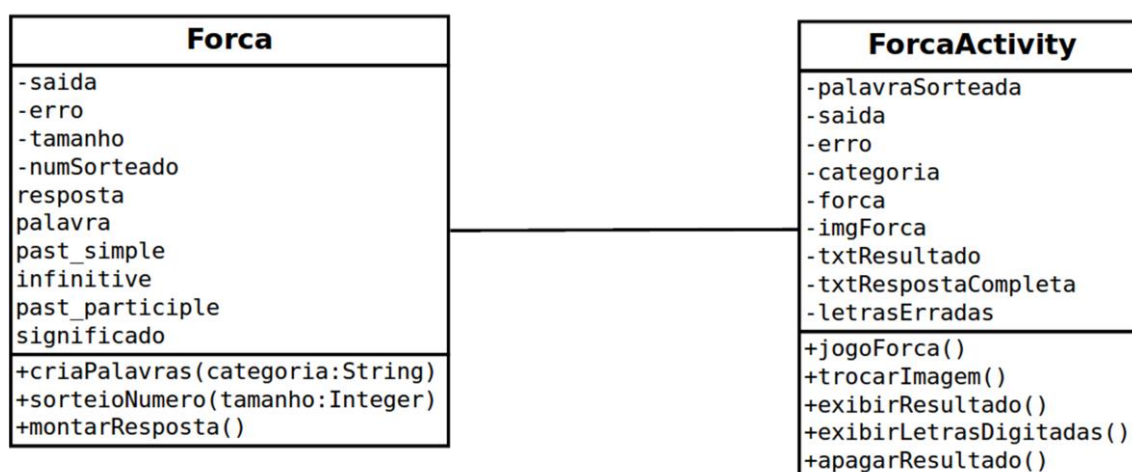


Figura 1. Diagrama de classes

Fonte: do próprio autor

Na tela ilustrada na Figura 2 o usuário tem a opção de escolher entre três tempos verbais: *Infinitive*, *Past Simple* e *Past Participle*. Após a escolha é apresentada a tela com o desenho da forca (Figura 3), o espaço para a palavra, os botões com as letras de A-Z, botão para sortear palavra e botão sair (volta à tela anterior). Logo abaixo aos botões há um espaço onde será apresentado o verbo sorteado conjugado nos seus respectivos tempos e o significado em português. Como já mencionado, a lista dos verbos utilizados pode ser consultada em Só Língua Inglesa (2017).

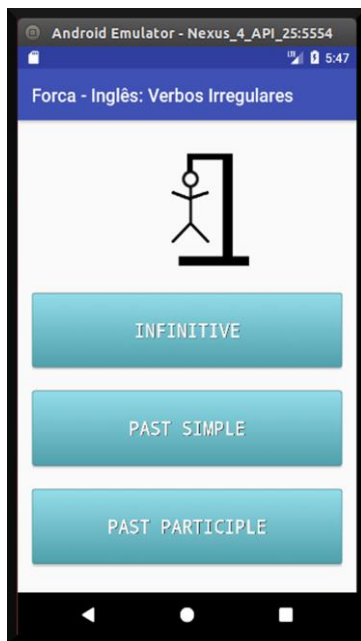


Figura 2. Tela inicial
Fonte: do próprio autor

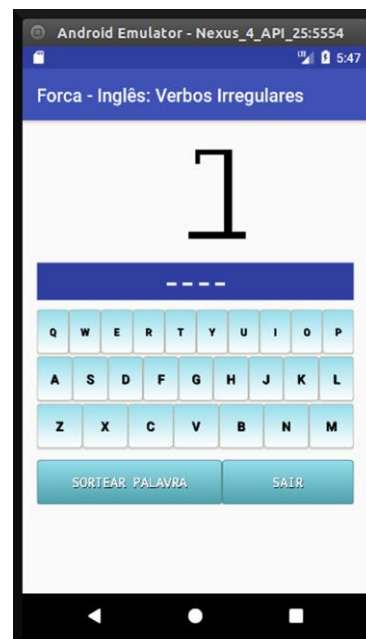


Figura 3. Tela do jogo
Fonte: do próprio autor

O *LinearLayout* foi utilizado em ambas as telas, pois ele “organiza os componentes um ao lado do outro, ou um abaixo do outro, dependendo da propriedade *android:orientation*” (GLAUBER, 2016). Os estilos dos botões foram adaptados de Querino Filho (2014).

Na construção da tela ilustrada na Figura 2 foram utilizados os seguintes componentes do *Android*: *ImageView* e *Button*. Na segunda tela (Figura 3) além dos *ImageView* e *Button* tem-se também o *TextView*.

No momento em que o usuário escolhe uma letra, a mesma vai para o respectivo espaço na palavra, caso esteja correta, ou para o espaço destinado às letras que não correspondam à palavra sorteada (Figura 4). A cada letra errada é acrescentado uma parte do corpo na forca. Quando o usuário acerta a palavra é exibida uma mensagem parabenizando-o pelo acerto e logo abaixo dos botões *SORTEAR PALAVRA* e *SAIR*, além do verbo nos tempos *Infinitive*, *Past Simple* e *Past Participle*, e o significado em português (Figura 5 e Figura 6).

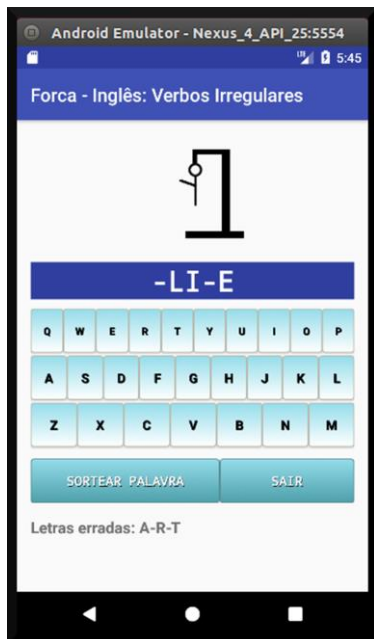


Figura 4. Letras erradas
Fonte: do próprio autor

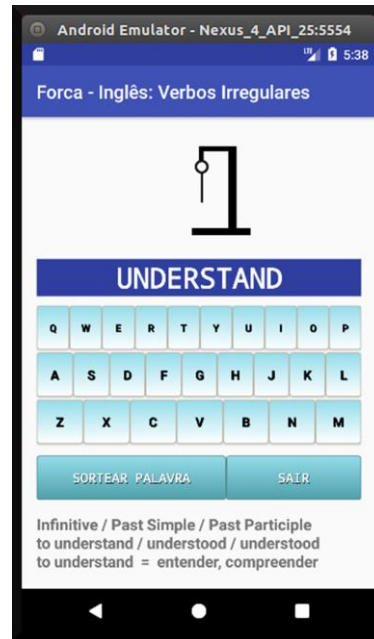


Figura 5. Palavra completada corretamente
Fonte: do próprio autor

Na sexta letra errada sugerida pelo usuário, o corpo desenhado fica completo na forca, aparecendo a mensagem ao usuário de que o mesmo errou a palavra sorteada. Nesse momento o aplicativo também mostra qual é o verbo correto (*feedback*), bem como as demais informações abaixo dos botões (Figura 7).

Um bom *feedback* é essencial para uma boa experiência de jogo. Um *feedback* constitui-se de um aviso ou elemento, visual ou sonoro, que traz informações e ajuda a validar as ações do usuário. Também auxilia no aprendizado do usuário e objetiva fazer com que o mesmo tenha a experiência de “imersão” no jogo.



Figura 6. Mensagem que acertou

Fonte: do próprio autor



Figura 7. Mensagem que errou

Fonte: do próprio autor

Para exibir as mensagens foi utilizado o *Toast* com o método *makeText* (Figura 8). Um *Toast* é uma mensagem que aparece durante um tempo e depois desaparece automaticamente. A duração pode ser *LENGTH_SHORT* e *LENGTH_LONG*, com duração de cerca 2 e 3,5 segundos respectivamente (GLAUBER, 2016). Neste caso foi utilizado o *LENGTH_SHORT*.

```
113 Toast.makeText(this, "Favor, sortear nova palavra ou sair!", Toast.LENGTH_SHORT).show();
```

Figura 8. Uso do *Toast*

Fonte: do próprio autor

O método de sorteio é apresentado na Figura 9. Ele retorna um número inteiro utilizando a classe *Random*, do pacote *java.util*. Essa classe é utilizada para gerar números aleatórios. Um objeto *Random* é criado e a variável *num* recebe um número sorteado dentro do tamanho recebido como parâmetro do *HashMap*. Esse número será utilizado para buscar uma chave equivalente e o valor contido no *HashMap*.

HashMap é uma das classes que implementam a *interface Map*. *HashMaps* armazenam elementos em tabelas de *hash*. Já *Map* é uma coleção que associa chaves a valores e que não pode conter chaves duplicadas. Coleção, por sua vez, é uma estrutura de dados – na realidade, um objeto – que pode armazenar referências a outros objetos (DEITEL, 2017).

```

562 // Método para sortear número
563 public int sorteioNumero(int tamanho) {
564     int num;
565     Random random = new Random(); // cria um objeto Random
566     num = random.nextInt(tamanho);
567     return num;
568 }

```

Figura 9. Método para sortear número

Fonte: do próprio autor

Na Figura 10 é criado um mapa *infinite*, definindo a chave do tipo *Integer* e o valor do tipo *String* (CAELUM,2017; DEITEL,2017; LANHELLAS,2017).

```

20 Map<Integer, String> infinite;

```

Figura 10. *Map infinite*

Fonte: do próprio autor

Para a lista dos verbos foi criado um *HashMap* e adicionados cada verbo com sua respectiva chave utilizando o método *put*. Como pode ser observado na Figura 11, na linha 35 foi adicionado ao *HashMap* o valor “*ARISE*” (*String*) com sua chave 0 (*Integer*). Foram criados *HashMaps* para todas as listas utilizadas (CAELUM,2017; DEITEL,2017; VINÍCIUS,2017).

```

32 // cria um HashMap para categoria verbos infinitive
33 infinite = new HashMap<>();
34 // associa um valor a uma chave específica
35 infinite.put(0, "ARISE");
36 infinite.put(1, "AWAKE");
37 infinite.put(2, "BE");
38 infinite.put(3, "BEAR");

```

Figura 11. *HashMap*

Fonte: do próprio autor

Para manipulação das palavras foi utilizada a classe *StringBuilder* do pacote *java.lang*. Ela foi utilizada, pois serve para criar e manipular informações de *string* dinâmica, isto é, *strings* modificáveis. Cada *StringBuilder* é capaz de armazenar um número de caracteres especificado pela sua capacidade. Se a capacidade de uma *StringBuilder* for excedida, ela é expandida para acomodar os caracteres adicionais (DEITEL, 2017).

3 CONSIDERAÇÕES FINAIS

Este artigo acompanhou a elaboração de um aplicativo móvel, cujo intuito é tornar-se uma ferramenta adicional a fim de ajudar as pessoas no processo de aprendizagem, neste caso, na memorização ou treino dos verbos irregulares do idioma inglês. O aplicativo foi desenvolvido simulando o jogo da forca, que é um jogo simples e que muitos conhecem e já tiveram a oportunidade de brincar. O mesmo foi desenvolvido para a plataforma *Android* por este ser o sistema operacional mais utilizado na atualidade.

O *software* foi desenvolvido com tecnologias de programação amplamente conhecidas no meio de desenvolvimento atual, e espera-se que tenha atingindo bom grau de funcionalidade, usabilidade e relevância como resultado final. Detalhes técnicos a respeito da programação são citados no corpo do artigo.

Paralelamente, também objetivou-se neste artigo ressaltar a importância de adequar conteúdo a dispositivos móveis, estendendo seu uso às mais diversas áreas de atuação humana, principalmente na educação.

Referências

CAELUM ensino e inovação. **Java e orientação a objetos**. Disponível em: <<https://www.caelum.com.br/apostila-java-orientacao-objetos/collections-framework/#16-11-mapas-java-util-map>>. Acesso em: 03 nov. 2017.

DEITEL, Paul; DEITEL, Harvey; DEITEL, Abbey. **Android: como programar**. 2. ed. Porto Alegre: Bookman, 2015.

DEITEL, Paul; DEITEL, Harvey. **Java: como programar**. 10. ed. São Paulo: Pearson, 2017.

GLAUBER, Nelson. **Dominando o Android do básico ao avançado**. 2. ed. São Paulo: Novatec, 2016.

HIGA, Paulo. **95,5% dos smartphones vendidos no Brasil são Androids**. 2016. Disponível em: <<https://tecnoblog.net/203749/android-ios-market-share-brasil-3t-2016/>>. Acesso em: 31 out. 2017.

LANHELLAS, Ronaldo. **HashMap Java: Trabalhando com listas key-value**. Disponível em: <<https://www.devmedia.com.br/hashmap-java-trabalhando-com-listas-key-value/29811>>. Acesso em: 03 nov. 2017.

OLIVEIRA, Sarah. **Verbos irregulares em inglês**. 2015. Disponível em: <<https://englishlive.ef.com/pt-br/blog/verbos-irregulares-em-ingles/>>. Acesso em: 16 set. 2017.

POSITIVO. **7 apps ótimos para quem quer aprender inglês**. Disponível em: <<http://www.positivoinformatica.com.br/doseujeito/estilo-de-vida/7-apps-para-quem-quer-aprender-ingles/>>. Acesso em: 16 set. 2017.

PRESSMAN, Roger S. **Engenharia de software: uma abordagem profissional**. 7. ed. Porto Alegre: AMGH, 2011.

QUERINO FILHO, Luiz Carlos. **Desenvolvendo seu primeiro aplicativo Android**. 1. ed. São Paulo: Novatec, 2014.

SÓ LÍNGUA INGLESA. Disponível em: <<http://www.solinguainglesa.com.br/conteudo/irregulares1.php>>. Acesso em: 11 set. 2017.

TANENBAUM, Andrew S.; BOS, Herbert. **Sistemas Operacionais Modernos**. 4. ed. São Paulo: Pearson Education do Brasil, 2016.

VINÍCIUS, Thiago. **Conhecendo a interface Map do Java**. Disponível em: <<https://www.devmedia.com.br/conhecendo-a-interface-map-do-java/37463>>. Acesso em: 03 nov. 2017.