

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

THIAGO ALEXSANDER DA COSTA PEREIRA

**FERRAMENTA PARA AUXÍLIO NO ENSINO E AVALIAÇÃO DE PRONÚNCIA
DE LÍNGUA INGLESA BASEADA EM SPEECH-TO-TEXT E SPACED
REPETITION**

CAMPO MOURÃO

2022

THIAGO ALEXSANDER DA COSTA PEREIRA

**FERRAMENTA PARA AUXÍLIO NO ENSINO E AVALIAÇÃO DE PRONÚNCIA
DE LÍNGUA INGLESA BASEADA EM SPEECH-TO-TEXT E SPACED
REPETITION**

**Computer-assisted tool for teaching and evaluation of English based on
speech to text and spaced repetition**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação do Curso de Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. André Luiz Satoshi Kawamoto

Coorientador: Prof^a. Me. Denize Ricardi

CAMPO MOURÃO

2022



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

THIAGO ALEXSANDER DA COSTA PEREIRA

**FERRAMENTA PARA AUXÍLIO NO ENSINO E AVALIAÇÃO DE PRONÚNCIA
DE LÍNGUA INGLESA BASEADA EM SPEECH-TO-TEXT E SPACED
REPETITION**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Bacharel em Ciência da Computação
do Curso de Bacharelado em Ciência da
Computação da Universidade Tecnológica
Federal do Paraná.

Data de aprovação: 13/junho/2022

André Luiz Satoshi Kawamoto
Doutor
Universidade Tecnológica Federal do Paraná

Denize Ricardi
Mestre
Universidade Tecnológica Federal do Paraná

Andre Luis Schwerz
Doutor
Universidade Tecnológica Federal do Paraná

Rafael Liberato Roberto
Doutor
Universidade Tecnológica Federal do Paraná

CAMPO MOURÃO
2022

RESUMO

A língua inglesa tem uma grande importância na atualidade. Essa importância se reflete inclusive na área de Computação, onde os termos em inglês são amplamente utilizados como jargões, nomeando dispositivos, componentes do computador e profissionais, entre outras coisas. Apesar disso, no Brasil boa parte da população, incluindo os profissionais da Computação, não tem conhecimento da língua inglesa. Para diminuir esse problema existem aplicativos voltados ao ensino de um segundo idioma, porém esses aplicativos concentram-se principalmente na gramática e *listening*, desconsiderando um aspecto muito importante: a pronúncia. O presente trabalho apresenta o processo de criação de uma ferramenta para auxiliar aprendizes de inglês na retenção de vocabulário e pronúncia correta de termos técnicos da Computação. Além disso, essa ferramenta pode ser útil para professores pois fornece subsídios para identificar as dificuldades dos alunos e ajudar o processo de ensino como um todo.

Palavras-chave: interface baseada em voz; computer-assisted language learning; speech-to-text; spaced repetition.

ABSTRACT

Currently the English language is the most predominant language in the world. This also reflects on the Computer Science field, where English terms are used for, terminology, device names, component and professionals. Although, in Brazil most of the population, including professionals in the Computer Science field, do not know how to speak English. Apps to reduce this issue exist, but the majority of those apps are focused on grammar and listening, disregarding the pronunciation aspect of learning a new language. This article presents the creation of a computer assisted language learning tool for pronunciation and retention of Computer related terms in English. This tool may also be useful for teachers to identify the difficulties of English pronunciation.

Keywords: voice based interface; computer-assisted language learning; speech-to-text; spaced repetition.

LISTA DE FIGURAS

Figura 1 – Curva de esquecimento	13
Figura 2 – Exemplo de <i>flashcard</i>	14
Figura 3 – <i>Leitner System</i>	15
Figura 4 – <i>ANKI</i> - Tela inicial	18
Figura 5 – Momento da revisão do <i>flashcard</i>	19
Figura 6 – Tela das categorias e avaliação do Duolingo	20
Figura 7 – Momento da revisão do <i>flashcard</i>	21
Figura 8 – Termos de uma lição	21
Figura 9 – Revisão de um termo	22
Figura 10 – Diagrama de casos de uso do sistema	30
Figura 11 – Arquitetura do sistema	31
Figura 12 – Diagrama Entidade Relacionamento	32
Figura 13 – Algoritmo adaptado	33
Figura 14 – Interface inicial Pronun-Master	35
Figura 15 – Navegação estrutural da aplicação	35
Figura 16 – Interface de criação de tópicos	36
Figura 17 – Interface de criação de <i>cards</i>	36
Figura 18 – Edição de um <i>card</i>	37
Figura 19 – Interface status de erros dos usuários	37
Figura 20 – Lista de Tópicos	38
Figura 21 – Tópico Computação 3 e seus <i>cards</i>	38
Figura 22 – Visão detalhada de um <i>cards</i>	39
Figura 23 – Revisão	39
Figura 24 – Revisão	39
Figura 25 – Pronúncia não aceita	40

LISTA DE ABREVIATURAS E SIGLAS

Siglas

API	Interface de programação de aplicações, do inglês <i>Application Programming Interface</i>
EF	Fator de Facilidade, do inglês <i>Ease Factor</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
SRS	Sistema de repetição espaçada, do inglês <i>Spaced Repetition System</i>
VUI	Interface de usuário de voz, do inglês <i>Voice User Interface</i>
W3C	<i>World Wide Web Consortium</i>

SUMÁRIO

1	INTRODUÇÃO	8
1.1	Considerações iniciais	8
1.2	Problema de Pesquisa	10
1.3	Objetivos e justificativa	10
1.4	Contribuições	10
1.5	Estrutura do trabalho	10
2	REFERENCIAL TEÓRICO	12
2.1	Técnicas de ensino para aprendizagem de segundo idioma	12
2.1.1	<i>Massed Practice</i>	12
2.1.2	<i>Spaced Repetition</i>	12
2.1.3	<i>Spaced repetititon system</i>	14
2.1.3.1	<i>Leitner System</i>	14
2.1.3.2	<i>Pimsleur</i>	15
2.1.3.3	<i>Algoritmo SM-2</i>	16
2.2	Aplicativos para Aprendizagem de um Segundo Idioma	17
2.2.1	ANKI	17
2.2.2	Duolingo	19
2.2.3	Memrise	20
2.3	Tecnologias de Interface de usuário de voz, do inglês <i>Voice User Interface (VUI)</i>	22
2.3.1	Google Cloud Speech-to-Text	22
2.3.2	Microsoft Azure	23
2.3.3	Dialogflow	24
2.3.4	Web Speech API	24
2.3.5	Síntese das características das ferramentas de <i>Speech-to-Text</i>	25
3	TRABALHOS RELACIONADOS	26
4	MÉTODO	28
4.1	Tecnologia de VUI	28
4.2	Análise de requisitos	28
4.2.1	Público-alvo	29

4.3	Arquitetura e modelagem do banco de dados	30
4.3.1	Modelagem do banco de dados	31
4.4	Algoritmo de <i>Spaced Repetition</i>	32
5	RESULTADOS	35
5.1	Discussões	40
6	CONCLUSÃO	41
	REFERÊNCIAS	42
	APÊNDICE A LISTA INICIAL DE PALAVRAS USADAS NA FERRAMENTA	45

1 INTRODUÇÃO

Após a Primeira Guerra Mundial a língua inglesa começou a se tornar uma das mais faladas no mundo e, a partir da globalização, veio a se converter num idioma compreendido pela maioria das pessoas. Por conta disso, o inglês é a principal língua para publicações de artigos visando divulgação científica e tecnológica (GRADDOL, 2000).

Essa predominância da língua inglesa pode ser vista também em outras áreas. Por exemplo, 62% dos sites da Internet têm seu conteúdo em inglês, seguido por apenas 5.7% dos sites em russo, de acordo com a W3Techs¹

1.1 Considerações iniciais

No Brasil uma apuração realizada em 2013 pelo instituto de pesquisa Data Popular em associação com o *British Council*² revelou que apenas 5% dos brasileiros dizem que tem algum conhecimento da língua inglesa³. Segundo esta pesquisa, empresas preferem que o candidato possua inglês instrumental e que tenha habilidade para realizar conversas em inglês, enquanto conhecimentos de gramática e escrita são menos valorizados. Os obstáculos que estudantes brasileiros encontram são a falta de tempo ou de recursos financeiros. A grande maioria (87%) prefere estudar em uma escola de idiomas e, ainda assim, existe uma grande taxa de desistência dos alunos. Muitos demonstram interesse em cursos 100% online visando reduzir custos ou maior flexibilidade.

No que diz respeito à área da Computação, há uma forte relação com a língua inglesa desde o início de sua popularização. Por volta do ano de 1944 foi criado nos Estados Unidos o ENIAC (*Electronic Numerical Integrator and Calculator*), considerado um dos precursores do computador digital. A partir daí, a evolução da tecnologia e dos computadores possibilitou a criação de linguagens de programação de alto nível e o desenvolvimento de programas para os computadores.

Com esse avanço, COBOL e Fortran foram as primeiras linguagens de alto de nível a serem desenvolvidas, sendo COBOL criada com o intuito de ser uma linguagem baseada no idioma inglês, para que fosse facilmente lida e autodocumentada, focada em sistemas administrativos além de ser manutenível. Isso resultou em mais de 300 palavras reservadas que constituíam originalmente a sintaxe do COBOL. Deste modo se tornando uma linguagem bastante popular na área de programas administrativos.

Com o passar do tempo outras linguagens foram criadas e, ainda que o número de palavras reservadas tenha sido reduzido, uma vez que foram substituídas por símbolos, a sin-

¹ <https://w3techs.com/> acesso: junho de 2022

² <https://www.britishcouncil.org.br/>

³ https://www.britishcouncil.org.br/sites/default/files/learning_english_in_brazil.pdf

taxe dessas linguagens obedece, de certa forma, o inglês (SELLINK; VERHOEF *et al.*, 1997; GOLDSTINE; GOLDSTINE, 1946; WEXELBLAT, 2014).

Atualmente, mesmo com a popularização do uso de computadores e da Internet ao redor do mundo, as linguagens de programação criadas em países não falantes da língua inglesa mantém a tendência de utilizar palavras reservadas em inglês como: Python⁴ (Holanda), Ruby⁵ (Japão) e Lua⁶ (Brasil).

Nos dias atuais, na tentativa de baratear seus custos operacionais, ou motivadas pela necessidade de atuar em diversos países, empresas contratam profissionais residentes em diversas localidades, inclusive o Brasil. Para esses profissionais, provavelmente, a comunicação se dá em inglês.

Entretanto, de forma geral, os falantes nativos de português têm dificuldade em adquirir inglês como segunda língua de forma a se comunicar de maneira aceitável ao conversar. Essa dificuldade em grande parte se percebe na pronúncia, devido à adição de novos fonemas, sílabas enfatizadas de maneira incorreta, adição de sílabas extras e utilização de fonemas do português para pronunciar palavras em inglês (ASSIS, 2007). Frequentemente, palavras em inglês que terminam com uma consoante são alvo de adição de sílabas como por exemplo: “submit” sendo pronunciado como “sub-i-mit-i”. O uso de fonemas do português em palavras em inglês é bastante comum em palavras que possuem duas vogais iguais consecutivas, exemplo: “look” é pronunciado como “luki”. Por conta desses fatores a comunicação em inglês pode não ser facilmente compreendida (LANTEIGNE, 2006).

Essa disparidade de pronúncia pode ocorrer, ainda, entre falantes do português que se encontram em diversas regiões do Brasil, podendo causar confusão de entendimento até mesmo entre dois nativos devido à forma como fonemas ou entonações são utilizados.

Na computação esse problema pode ser acentuado, visto que vários termos têm sido incorporados na comunicação do dia a dia. Alguns termos que têm correspondência em português não são utilizados, como por exemplo “*mouse*”, “*bug*” e “*touch*”.

O avanço da tecnologia proporcionou o crescimento de aplicativos e serviços Web voltados à aprendizagem de um segundo idioma. Melhores processadores e maior capacidade de armazenamento possibilitaram que aplicativos anteriormente disponíveis apenas para desktop fossem disponibilizados também para plataformas móveis (GODWIN-JONES, 2011).

Softwares como Duolingo⁷, Memrise⁸ e outros viabilizam estudar em qualquer lugar, em qualquer horário e com baixo custo. Uma possível desvantagem de muitos desses aplicativos é que o foco principal é gramática e *listening* (ação de ouvir), deixando a pronúncia correta da língua em segundo plano.

⁴ <https://docs.python.org/2.0/ref/node92.html>

⁵ <https://www.ruby-lang.org/pt/about/>

⁶ <https://www.lua.org/about.html>

⁷ <https://www.duolingo.com>

⁸ <https://www.memrise.com>

1.2 Problema de Pesquisa

Como citado, falantes de português nativo possuem dificuldades na pronúncia de palavras de língua inglesa. Essa dificuldade ocorre inclusive entre profissionais da área da Computação, uma área em que o conhecimento de alguns termos em inglês é essencial.

Além disso, existem poucas ferramentas com o objetivo de corrigir ou ensinar a pronúncia das palavras. Uma das técnicas mais populares consiste em fazer o aprendiz gravar sua própria voz, e ele próprio compara sua gravação com um áudio contendo a pronúncia correta. Isso pode ser feito via um aplicativo de gravação pelo celular ou computador.

1.3 Objetivos e justificativa

O objetivo deste trabalho é construir uma ferramenta para auxiliar a retenção de termos técnicos em inglês da área da Computação. Os usuários da ferramenta devem ser capazes de pronunciar palavras utilizando um microfone e, por meio de uma tecnologia de *Speech-To-Text* a ferramenta deve avaliar se essa pronúncia está correta ou não.

É previsto o uso de *Speech-To-Text* associado a *Spaced Repetition*, uma vez que alguns estudos indicam que eles contribuem na retenção dos termos recém aprendidos e aprimoram a pronúncia dos mesmos (KANG, 2016; NERI; CUCCHIARINI; STRIK, 2003; CHUKHAREV-HUDILAINEN; KLEPIKOVA, 2016).

Além disso, professores usuários da ferramenta terão subsídios para perceber como seus alunos estão pronunciando as palavras, bem como quais apresentam maior nível de dificuldade, podendo então atuar mais eficientemente para corrigi-los.

Uma vantagem para os alunos é que a ferramenta oferece feedback de forma automatizada, ajudando-os a aprimorar e memorizar as palavras. Além disso, o fato de ser disponibilizada online permite praticar em horários e em ambientes adequados ao seu cotidiano.

1.4 Contribuições

A ferramenta utiliza de recursos tecnológicos para o ensino da pronúncia de termos de língua inglesa, e fornece dados sobre o desempenho dos estudantes que a utilizam. Ela também possibilita que pesquisas acerca do *Spaced Repetition* e *Speech-to-Text* para a retenção de termos sejam realizadas.

1.5 Estrutura do trabalho

O restante desse trabalho está organizado da seguinte maneira: no Capítulo 2 são apresentados conceitos relacionados ao tema e que ajudam para a melhor compreensão do texto;

os trabalhos relacionados a aprendizagem com o uso de *Spaced Repetition* ou *Speech Recognition* estão relatados no Capítulo 3; no Capítulo 4 a metodologia para desenvolvimento da proposta e, finalmente, as considerações finais são apresentadas no Capítulo 6.

2 REFERENCIAL TEÓRICO

Neste capítulo são apresentadas técnicas para se aprender um segundo idioma, como também aplicativos que populares que se utilizam de *Spaced Repetition* e tecnologias de reconhecimento de voz.

2.1 Técnicas de ensino para aprendizagem de segundo idioma

Saber uma segunda língua é estar apto a utilizá-la a qualquer momento e assim garantir que, mesmo após um longo período, a pessoa seja capaz de se comunicar. Nesse sentido, um dos fatores determinantes para aprendizagem é utilizar um método eficiente. Uma das condições que indicam o nível de fluência em uma segunda língua é a quantidade de vocabulário. Duas técnicas bastante populares para aquisição de vocabulário por conta própria são *Massed Practice* e *Spaced Repetition*, apresentadas a seguir.

2.1.1 *Massed Practice*

Massed Practice ou Estudo Prolongado é um estudo de um tópico por um longo período de tempo sem interrupções ou com curtos intervalos (SCHMIDT; WRISBERG, 2008). Por exemplo, estudar todo o conteúdo de uma prova somente no dia anterior. Esta técnica é muito popular entre estudantes na tentativa de ir bem em avaliações. Quando a duração das sessões de estudo é maior que o intervalo entre elas se configura *Massed Practice*.

O que se pode observar com esta técnica é que, caso um teste seja realizado após um curto período de tempo de sua utilização, gera resultados satisfatórios. Porém para testes realizados após um longo período desde sua aplicação, não obtém bons resultados (FINI; ZAINALIPOUR; MOZAFARI, 2010; KRUG; DAVIS; GLOVER, 1990).

2.1.2 *Spaced Repetition*

Diferente de *Massed Practice*, *Spaced Repetition* (Repetição espaçada) consiste em curtos períodos de estudo distribuídos ao longo do tempo. *Spaced Repetition* é baseada no efeito espaçado primeiramente observado por Ebbinghaus (2013, reimpressão).

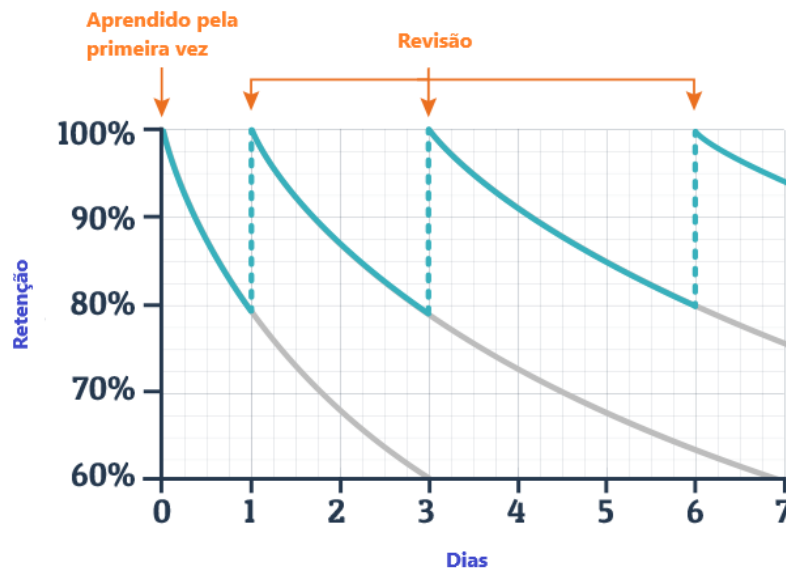
Ebbinghaus observou que ao estudar um conteúdo e revisá-lo ao longo do tempo, produz-se uma melhor retenção a longo prazo. Após Ebbinghaus ter documentado este efeito, o qual foi chamado de curva de esquecimento, surgiram vários estudos sobre o efeito espaçado para o aprendizado um segundo idioma (BLOOM; SHUELL, 1981; ATKINSON, 1972).

A curva de esquecimento representa o declínio de retenção dos termos ao longo de um período de tempo, ou seja uma informação recentemente aprendida tende a ser esquecida se

não for constantemente revisada (EBBINGHAUS, 2013, reimpressão). A Figura 1 apresenta a curva de esquecimento relacionada a uma taxa de retenção de 60% a 100% em um período de 7 dias.

Figura 1 – Curva de esquecimento

Curva de esquecimento típica para informações recém aprendidas



Fonte: Adaptado de wranx (2019).

No dia que o estudante aprende algo novo (dia 0) a sua taxa de retenção é de 100%. Durante os próximos dias, essa taxa vai decaindo, e pode alcançar 60% em 3 dias. Entretanto, a revisão desses termos eleva a taxa de retenção de volta a 100% e estende o tempo transcorrido até decair aos mesmos 60%, suavizando a curva de esquecimento. Assim, a retenção de termos pode ser maior ao longo do tempo, desde que revisões periódicas aconteçam (EBBINGHAUS, 2013, reimpressão).

É comum o uso de *flashcards* (cartões de estudo) em conjunto com *Spaced Repetition*. Esses cartões possuem informações em ambos os lados. Na frente, uma imagem, fórmula, texto, ou qualquer coisa associada ao item que se deseja aprender e no verso, as informações (respostas) sobre este item. A Figura 2 apresenta um *flashcard* (frente e verso) utilizado para aprender o verbo *paint* (pintar) em inglês.

Estudos comparando *Massed Practice* e *Spaced Repetition* indicam que, a longo prazo, a segunda técnica apresenta melhores resultados no que diz respeito às taxas de retenção (FINI; ZAINALIPOUR; MOZAFARI, 2010; KRUG; DAVIS; GLOVER, 1990)

Figura 2 – Exemplo de *flashcard*



Fonte: Maraglia e Peixoto (2017).

2.1.3 *Spaced repetition system*

Quando se automatiza, utilizando um computador, o cálculo do intervalo entre as repetições, seja por meio de uma heurística ou a partir de um algoritmo, temos um Sistema de repetição espaçada, do inglês *Spaced Repetition System* (SRS).

O uso de um sistema automatizado é vantajoso pois determinar a intermitência entre as revisões pode sobrecarregar o aluno devido ao número de *flashcards* que ele pode ter. Assim, pode-se priorizar os estudos em vez de usar o tempo no cálculo de intervalos.

Os algoritmos de SRS foram criados para obter um intervalo ideal entre as repetições de cada item. Assim, facilitando a experiência do usuário e maximizando o efeito de memorização dos termos. Estes algoritmos podem ser simples, como apenas adicionar ou remover um período de tempo específico entre a intermitência dos itens, ou podem ser mais complexos, como gerar um intervalo ótimo entre revisões e que se adaptam ao usuário, fazendo com que as revisões aconteçam antes do item ser esquecido de acordo com a curva de esquecimento (LINDSEY *et al.*, 2014; METZLER-BADDELEY; BADDELEY, 2009; TABIBIAN *et al.*, 2019).

2.1.3.1 *Leitner System*

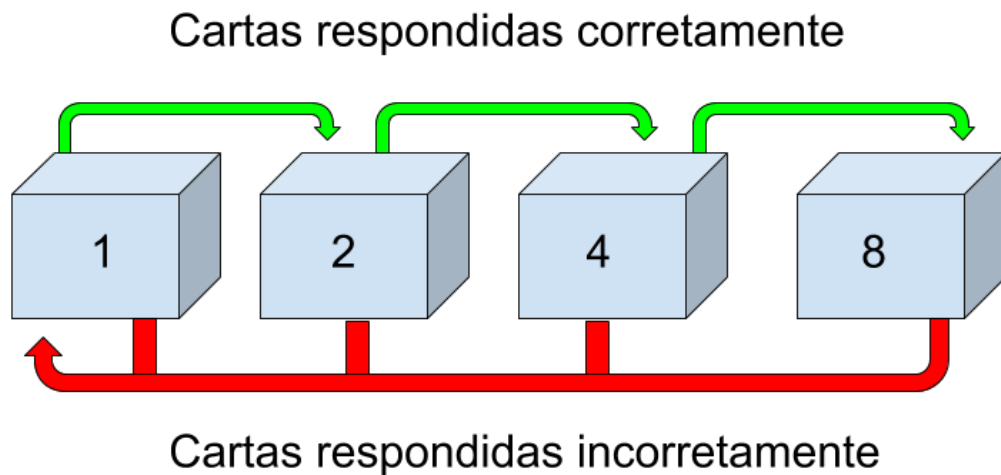
Este algoritmo foi proposto por Sebastian Leitner e consiste em separar *flashcards* em grupos, que são revisados em intervalos crescentes. Quanto maior o intervalo, maior é o fator de retenção do termo, ou seja, o item está melhor memorizado (LEITNER, 1972).

De acordo com esse algoritmo, os *flashcards* do primeiro grupo devem ser revisados com maior frequência, pois são aqueles que o aprendiz comete mais erros.

A Figura 3 corresponde ao sistema de Leitner com quatro grupos, com intervalos de 1, 2, 4, e 8 dias, respectivamente. Toda vez que o usuário revisar um item, duas possibilidades

podem acontecer: se ele acertar a resposta, esse item é movido para o próximo grupo; caso contrário, o item deve retornar para o primeiro grupo. As revisões dos itens deve obedecer os intervalos previstos.

Figura 3 – Leitner System



Fonte: Adaptado de Guru (2018).

A vantagem desse método é que o aprendiz pode focar nos cartões mais difíceis, que permanecem nos grupos iniciais. Como resultado, idealmente, há uma redução no tempo de estudo necessário devido ao efeito espaçado.

O Algoritmo 1 apresenta uma iteração para um grupo de *flashcards*.

Algoritmo 1 – Leitner system

Fonte: Adaptado de Guru (2018).

inserir *NGrupos, Intervalo*

- 1: **se** *Intervalo == IntervaloDoGrupo* **então**
 - 2: Fazer revisão de um Flashcard
 - 3: **se** *Resposta == Correta* **então**
 - 4: *ProximoGrupo := flashcard*
 - 5: **senão,**
 - 6: *PrimeiroGrupo := flashcard*
 - 7: **finaliza se**
 - 8: **finaliza se**
-

2.1.3.2 *Pimsleur*

Paul Pimsleur, em suas pesquisas sobre memorização para aprendizagem de idiomas, desenvolveu um *schedule* que foi chamado de *graduated interval recal* (intervalo gradual de lembrança). Um estudante, ao aprender um novo idioma, é apresentado a novos termos em intervalos incrementais. Estes intervalos são exponenciais, ou seja, se a primeira revisão após

a apresentação do termo ocorre após 10s (segundos), então a próxima revisão será após 10²s (100s), a terceira revisão após 10³s (aproximadamente 16 minutos), e assim sucessivamente. Deste modo permitindo que a memória de curto prazo se torne memória de longo prazo (PIMSLER, 1967).

Geralmente, o *schedule* utilizado em aplicativos que seguem o algoritmo de Pimsleur obedece a seqüência: 5 segundos, 25 segundos, ..., 5 dias, ..., 2 anos.

O Algoritmo 2 apresenta o algoritmo pra calcular o intervalo de Pimsleur.

Algoritmo 2 – Pimsleur

Fonte: Pimsleur (1967, p.73-75).

inserir *Intervalo, Exponencial*

1: *Resultado*:= power(*Intervalo, Exponencial*)

2: *:Exponencial*+= 1

3: **retorna** *Resultado*

2.1.3.3 Algoritmo SM-2

SM-2¹ (Super Memo-2) é a segunda versão do algoritmo adotado pelo software *Super-Memo*, criado por Piotr Wozniak.

A ideia é classificar os termos presentes nos *flashcards* de acordo com o nível de dificuldade. Essa classificação é baseada em quatro parâmetros de entrada: qualidade, repetição, intervalo e Fator de Facilidade, do inglês *Ease Factor* (EF) anteriores. EF diz respeito à facilidade com que o usuário se recorda do termo. Na primeira chamada do algoritmo, os três últimos parâmetros assumem um valor *default* (padrão). Esses parâmetros são descritos a seguir.

A **qualidade** é um valor inteiro entre 0 e 5 que indica a facilidade em que a informação foi lembrada:

- 5 - Resposta perfeita, ou seja, o termo foi lembrado sem hesitação.
- 4 - Resposta correta após hesitar.
- 3 - Resposta correta após dificuldade de lembrar.
- 2 - Resposta incorreta, apesar de parecer fácil de lembrar.
- 1 - Resposta incorreta durante o teste, após o teste (e a apresentação da resposta), o termo foi lembrado.
- 0 - Não lembrou de nada.

A **repetição** é o número de vezes que o item foi repetido antes da revisão corrente. O valor para a primeira revisão do item é zero.

¹ <https://www.supermemo.com/en/archives1990-2015/english/ol/sm2>

O **intervalo anterior** é um valor gerado a partir da iteração anterior, utilizado para determinar quando será a próxima revisão. O valor *default* desse parâmetro é zero.

Finalmente, a **EF anterior** é um número maior que 1.3 gerado pela iteração anterior. Na primeira iteração o valor *default* é 2.5. Esse valor é ajustado de acordo com o parâmetro de qualidade (1 a 5), e é usado para determinar o intervalo até a próxima revisão do item.

O algoritmo é apresentado no Algoritmo 3.

Algoritmo 3 – SM-2

Fonte: Adaptado de SuperMemo (2015).

```

inserir qualidade, repetição, EF, IntervaloAnterior
1: se qualidade >= 3 então
2:   se repetição == 0 então
3:     Intervalo:= 1
4:   finaliza se
5:   se repetição == 1 então
6:     Intervalo:= 6
7:   finaliza se
8:   se repetição > 1 então
9:     Intervalo:= IntervaloAnterior * EF
10:  finaliza se
11:  Intervalo:= Ceil(Intervalo)
12:  repetição +=1
13:  EF:= EF + (0.1 - (5 - qualidade) * (0.08 + (5 - qualidade) * 0.02))
14: finaliza se
15: se qualidade < 3 então
16:   repetição = 0
17:   Intervalo:= 1
18:   EF:= EF
19: finaliza se
20: se EF < 1.3 então
21:   EF:= 1.3
22: finaliza se
23: retorna intervalo, repetição, EF

```

2.2 Aplicativos para Aprendizagem de um Segundo Idioma

Alguns aplicativos populares, que são utilizados para ensino de segundo idioma baseados na técnica de SRS são descritos a seguir.

2.2.1 ANKI

Anki² é um software *Open Source* de memorização e está disponível para as plataformas Windows, Mac OSX, Linux/FreeBSD e Web. É conhecido por ser “agnóstico de informação”, ou

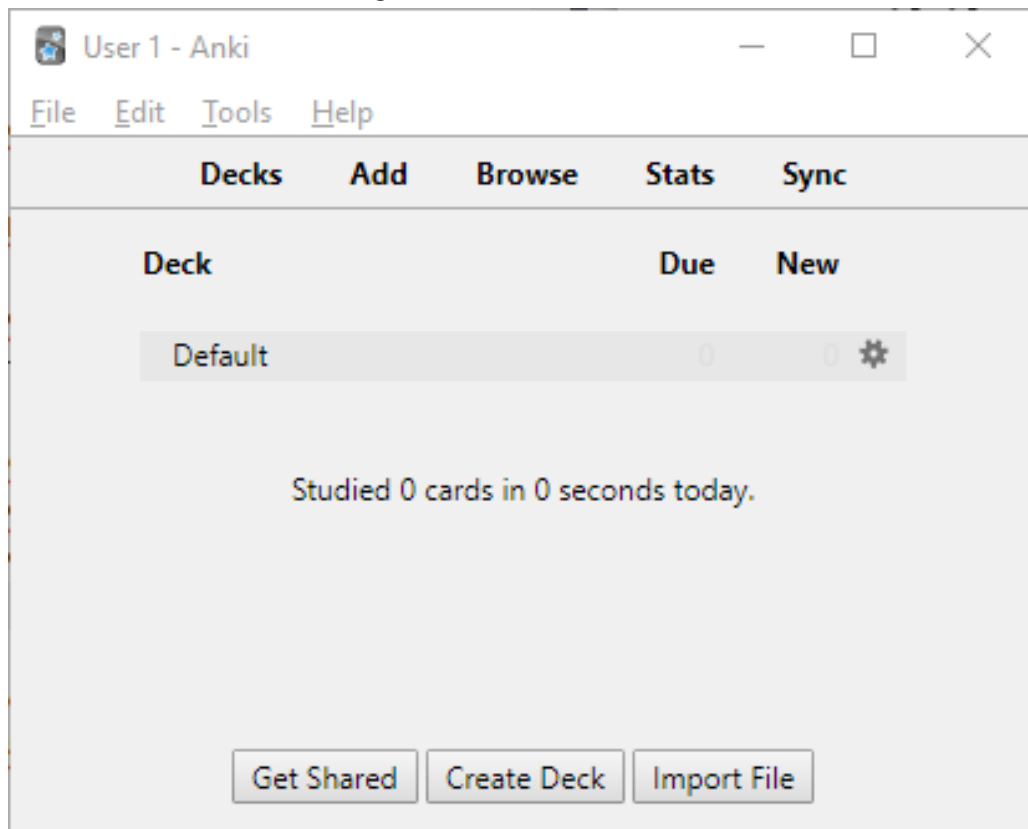
² <https://apps.ankiweb.net/>

seja, pode ser usado para estudar qualquer tipo de conteúdo, sendo que os mais populares são línguas estrangeiras e medicina.

Anki utiliza o algoritmo SM-2 para implementar *Spaced Repetition* e *Active Recall Testing*, que é uma forma de teste em que o usuário, ao ser questionado, precisa lembrar da resposta sem o auxílio de múltipla escolha.

Neste aplicativo os usuários são responsáveis por criarem seus próprios *flashcards*, para isso o software suporta imagens, áudio, vídeos e notações científicas (via LaTeX). A Figura 4 exibe a tela inicial do aplicativo. Nela, o usuário pode criar novos conjuntos de cartas (*decks*) ou importa-los. Nesta tela é possível escolher qual *deck* (conjunto de cartas) revisar como também verificar quantos itens estão disponíveis para revisão.

Figura 4 – ANKI- Tela inicial



Fonte: Autoria própria (2022).

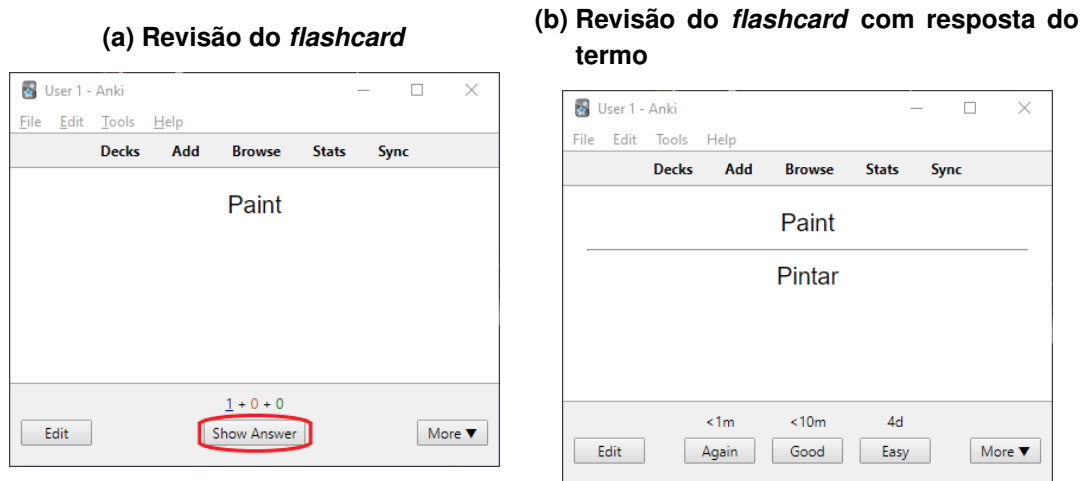
Os *decks* do Anki podem ser organizados por assunto e compartilhados com outros usuários pela Internet. O aplicativo oferece um amplo grau de personalização dos *flashcards* e *decks*.

A Figura 5 mostra as telas de revisão de um *flashcard* na interface original do aplicativo. A Figura 5a exibe a interface na qual o usuário é apresentado a um termo. O usuário deve tentar lembrar qual o significado do mesmo, e então apertar o botão “*Show Answer*” para revelar a resposta.

A partir desse momento, a interface exibida é a da Figura 5b. Nessa figura, observa-se que a resposta é apresentada logo abaixo do termo e três novos botões aparecem na barra

inferior para que o usuário selecione quando deverá ocorrer nova revisão, de acordo com o grau de entendimento do usuário.

Figura 5 – Momento da revisão do *flashcard*



Fonte: Autoria própria (2022).

2.2.2 Duolingo

Duolingo³ é uma plataforma gratuita que permite o aprendizado de cerca de 90 idiomas diferentes por meio de lições de gramática, disponível para Android, iOS, Windows Phone, Windows 10 Mobile e Web. Também oferece uma versão *Premium* chamada *Duolingo Plus* que custa US\$6.99 por mês, removendo propagandas entre as lições e adicionando a opção de realizar o *download* de cursos para que possam ser feitos sem o uso da Internet.

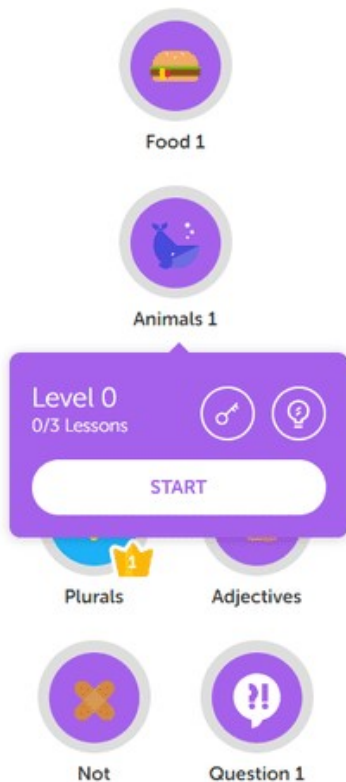
Esse aplicativo faz uso de gamificação para incentivar seus usuários a estudar os conteúdos através de tabelas de liderança que são feitas de acordo com pontuações recebidas a cada teste e lição feita pelo estudante. Estas lições são divididas em níveis incrementais de dificuldade e, em cada nível, categorias. Cada categoria possui várias lições nas quais o usuário é apresentado a fonemas, áudios e imagens para auxiliar o ensino.

A Figura 6 mostra as duas telas do Duolingo, a árvore de lições e a tela onde o usuário é testado. Na Figura 6a os usuários são apresentados a lições que possuem temas diversos para ensinar vocabulários e gramática relacionados, enquanto a Figura 6b apresenta a interface em que os usuários são testados sobre o itens abordados, avaliando o conteúdo estudado nas lições anteriores.

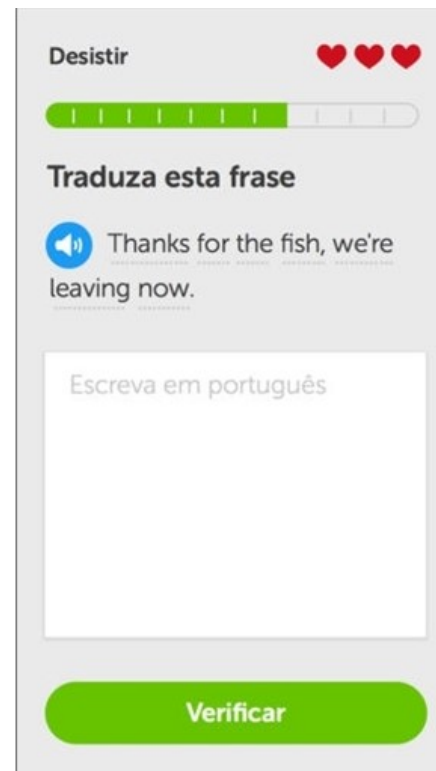
³ <https://www.duolingo.com>

Figura 6 – Tela das categorias e avaliação do Duolingo

(a) Árvore de categorias



(b) Avaliação



Fonte: Autoria própria (2022).

2.2.3 Memrise

Memrise⁴ é uma plataforma de ensino gratuita que utiliza *Spaced Repetition* para ensinar cursos diversos. Tais cursos podem ser criados tanto pela equipe responsável pelo aplicativo como também por usuários. Os tópicos mais populares são cursos de segundo idioma, a ferramenta permite *flashcards*, com imagens e vídeos e também faz o uso de gamificação com tabelas de lideranças global ou entre amigos do usuário e recompensando o usuário com pontos. Os pontos são adquiridos após realizar lições e testes que o aplicativo fornece. A Figura 7a mostra uma lista de cursos de idioma, com diferentes graus de dificuldade, que o usuário pode aprender. Nesta tela o usuário tem a opção de criar um curso ou procurar outras categorias. Após escolher um curso, o usuário é conduzido para a interface que exibe o seu conteúdo (Figura 7b). Nesta tela o usuário pode ver seu progresso, estudar uma das lições e revisar itens do curso. Dentro de cada lição se encontram o vocabulário, os termos que serão ensinados e o progresso dos itens vistos anteriormente. Também, é possível ver quais termos podem ser revisados imediatamente ou em um intervalo de tempo.

Na Figura 8, são exibidos os termos que compõem a primeira lição do curso de Coreano 1 e quais termos estão prontos para ser aprendidos.

⁴ <https://www.memrise.com>

Figura 7 – Momento da revisão do *flashcard*

(b) Lições do curso de Coreano

(a) Lista de cursos

Fonte: Autoria própria (2022).

Figura 8 – Termos de uma lição

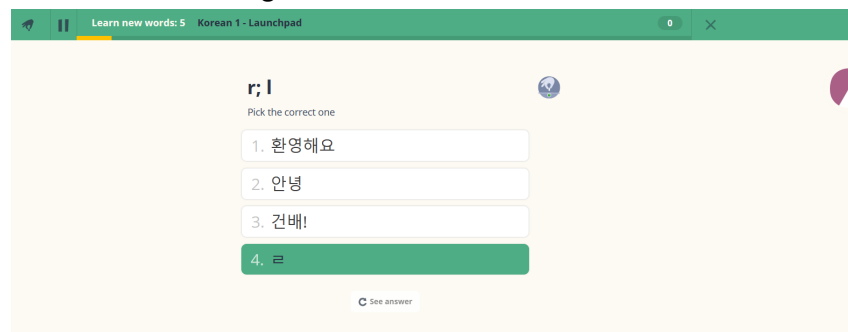
Palavra	Tradução	Status
ㅏ	a	Pronto para aprender
ㅑ	g	Pronto para aprender
가	gol	Pronto para aprender
ㅈ	j	Pronto para aprender
가자	let's go	Pronto para aprender
ㅏ	h	Pronto para aprender
하자	let's do it!	Pronto para aprender
ㄴ	n	Pronto para aprender
ㅣ / ㅍ	e	Pronto para aprender
네	yes	Pronto para aprender
ㅊ	o	Pronto para aprender
ㅌ	l; r	Pronto para aprender
노래	song	Pronto para aprender
...

Fonte: Autoria própria (2022).

Já a Figura 9 mostra como testes e revisões são aplicados, os termos são apresentados em formato de texto, áudio ou vídeo, e através de múltipla escolha se determina qual a resposta certa, o usuário possui um tempo limite para responder.

O Memrise utiliza um algoritmo de *Spaced Repetition* similar ao de Pimsleur, com intervalos de: 4 horas, 12 horas, 1 dia, 6 dias, 12 dias, 24 dias, 48 dias, 96 dias, 180 dias. O aplicativo está disponível para: Android, iOS e Web.

Figura 9 – Revisão de um termo



Fonte: Autoria própria (2022).

2.3 Tecnologias de VUI

Speech-to-Text (voz para texto) é o termo utilizado para softwares que utilizam modelos de *Machine learning* (aprendizado de máquina) para processamento de linguagem natural, ou seja, a partir de uma amostra de áudio é gerado um texto correspondente ao conteúdo identificado nessa amostra. *Speech synthesis* (síntese de fala) ou *Text-to-Speech* (texto para voz), por sua vez, refere-se ao uso de um sintetizador que transforma o conteúdo do texto em áudio com o uso de voz sintética.

Essa tecnologia permite vários tipos de aplicações como, por exemplo, criar comandos de voz para celulares ou transcrever ligações telefônicas.

Para que o uso desses recursos seja facilitado, empresas fornecem Interface de programação de aplicações, do inglês *Application Programming Interface* (API) para que usuários possam construir ferramentas que utilizam essa tecnologia. Algumas APIs que oferecem tais serviços são: Google Cloud Speech-to-Text, Microsoft Azure, Web Speech API e são descritas a seguir.

2.3.1 Google Cloud Speech-to-Text

Google Cloud Speech-to-Text⁵ é uma API da Google que, por meio de modelos neurais de reconhecimento de voz, pode ser usada para reconhecer mais de 120 idiomas e dialetos.

Tendo a capacidade de analisar áudio e identificar automaticamente o idioma falado, a função de identificar o idioma é recomendado para comandos curtos, e funciona listando 3 idiomas alternativos e um idioma principal, deste modo ao realizar o comando em qualquer um dos 4 idiomas ele será reconhecido e realizado.

O processo de transformar áudio em texto pode ocorrer em tempo real ou por meio de um arquivo. Segundo a própria Google, a tecnologia de *machine learning* utilizada é similar à usada na criação automática de legendas do YouTube.

⁵ <https://cloud.google.com/speech-to-text>

O usuário pode escolher entre quatro modelos de reconhecimento de voz, cada um apropriado para um objetivo:

- *command_and_search*, para comandos curtos ou pesquisa por voz;
- *phone_call*, para chamadas telefônicas, o áudio neste model são gravados com uma taxa de amostragem de 8.000 Hz e está disponível apenas para o idioma inglês;
- *video*, indicado para vídeo e situações em que existe mais de um interlocutor, sugere-se que o áudio fornecido tenha uma taxa de amostragem de 16.000 Hz, está disponível apenas para o idioma inglês;
- *default*, o modelo padrão para áudios de longas durações e que não corresponde a nenhum modelo anterior.

Para as transcrições existe a possibilidade de se adicionar um contexto, que auxiliará no processamento e reconhecimento do áudio, podendo ser uma lista de frases que serve para aumentar a reconhecimento das frases ou palavras desejadas.

Em sua opção gratuita os modelos *phone_call* e *video* não estão disponíveis. A opção permite processar até 60 minutos de áudio, e a partir de 60 minutos o custo é de US\$0,006 a cada 15 segundos.

A API oferece suporte para as seguintes linguagens de programação: C#, Java, Python, Ruby, Go, Node.JS e PHP.

2.3.2 Microsoft Azure

Microsoft Azure⁶ é uma API fornecida pela Microsoft que oferece a função de *Speech-to-Text* e *Text-to-Speech*, com suporte para Android SDK e Java 8.

Nela os usuários podem criar modelos de reconhecimento de voz personalizados ou usar modelos próprios da Microsoft. Por padrão, é utilizado um modelo universal de reconhecimento, recomendando para cenários de conversação. O serviço utilizado nesses modelos é provido pela mesma tecnologia usada na Cortana. A API também é capaz de transcrever e, opcionalmente, traduzir áudio em tempo real.

A API possui uma versão grátis a qual permite até 5 horas de áudio por mês para a função *Speech-to-Text*, já para *Text-to-Speech* permite 5 milhões de caracteres por mês. Neste plano é permitido somente apenas uma requisição de cada função por vez. Os valores do plano pago para a função *Speech-to-Text* é US\$1 por hora de áudio, para *Text-to-Speech* é US\$4 para cada 1 milhão de caracteres utilizados, além de permitir mais de uma requisição por vez.

⁶ <https://azure.microsoft.com>

2.3.3 Dialogflow

Dialogflow⁷ é um projeto de *Speech-to-Text* que foi adquirido pela Google, especializado em reconhecer 14 idiomas e realizar a análise de sentimento dos mesmos, recomendado para serviços de assistente de voz.

Através de modelos de aprendizagem de máquina a API utiliza intenções, entidades e contexto para compreender o contexto da frase. O usuário precisa configurar um agente que é um módulo para processar linguagem natural. Quando existe uma relação entre o áudio ou texto e uma intenção, uma resposta pode ser enviada de volta para o usuário. As intenções são criadas no agente, cada intenção poderá ser mapeada a uma resposta e geralmente representa um diálogo como por exemplo “meu animal favorito é a zebra”, permitindo assim processar a motivação.

Uma entidade serve para identificar e extrair dados da linguagem natural. Usada para captar informações como nome de produtos ou nome de ruas, todos esses dados precisam ter uma entidade correspondente. Um contexto é o estado atual permitindo a troca de intenções durante o diálogo.

Como utiliza modelos da Google Cloud os planos gratuitos e pagos são similares aos oferecidos pelo Google Cloud, como também as linguagens na qual suporta, sendo elas: C#, Java, Python, Ruby, Go, Node.JS e PHP.

Em sua versão gratuita para todas suas funções, excluindo as funções *Toll-free phone call* e análise de sentimento, porém existe limitações, para requisições de áudio pode realizar 100 requisições por minuto com um máximo de 15.000 por mês, cada requisição pode possuir no máximo de 60 segundos. Para requisições de texto ou assistente da Google o limite é de 180 requisições por minuto. O serviço de chamada de telefone possui um limite de 3 minutos para cada chamada.

A versão *plus* oferece 300 requisições por minuto para áudio com duração máxima de 60 segundos, e 600 requisições por minuto para texto ou assistente do Google. Para o serviço de chamada de telefone não possui limites. O preço da versão *plus* depende de qual funcionalidade vai ser utilizada, sendo *Speech-to-Text* US\$0.0085 a cada 15 segundos e *Speech Synthesis* US\$4 para cada 1 milhão de caracteres.

2.3.4 Web Speech API

Web Speech API⁸, é um iniciativa conjunta da comunidade *World Wide Web Consortium* (W3C) para fornecer *Speech Recognition* e *Speech Synthesis* para *Web Apps* ou páginas da web, através de JavaScript. A API suporta comandos curtos de voz e frases contínuas que o

⁷ <https://dialogflow.com>

⁸ <https://wicg.github.io/speech-api/>

Quadro 1 – Comparativo entre APIs

API	Idiomas suportados	Plataforma alvo	Linguagem de programação	Plano gratuito - Speech to Text	Licença
Google cloud speech to text	120+	PC/Web	C#, Java, Python, Ruby, Go, Node.js, PHP	Áudios menores de 60 minutos	US\$0.006 para cada 15 segundos
MicrosoftAzure	30	PC/Web	Android SDK, Java 8	5 horas por mês	US\$1 por hora
Dialogflow	14	PC/Web	C#, Java, Python, Ruby, Go, Node.js, PHP	Áudios menores de 60 minutos	US\$0.0085 para cada 15 segundos acima de 60 minutos
Web Speech API	31	Android, Web (Apenas Google Chrome)	JavaScript	Grátis	Grátis

Fonte: Autoria própria (2022).

usuário pode configurar utilizando uma lista de gramática para definir quais palavras o sistema deve reconhecer.

A funcionalidade de *Text-to-Speech* está disponível para os navegadores Google Chrome, Microsoft Edge, Firefox, Opera, Safari, Android Webview, Chrome for Android, Firefox for Android, Safari on iOS. Entretanto, *Speech-to-Text* pode ser usada apenas no Google Chrome.

Ao utilizá-la no Google Chrome, o áudio fornecido é enviado para um servidor de processamento, deste modo requer acesso à Internet para que funcione. Usando a API em uma aplicação de JavaScript ela fará o uso da ferramenta de *Speech Recognition* padrão do sistema no qual estiver sendo utilizada, por exemplo Cortana (Windows 10) e Siri (iOS).

2.3.5 Síntese das características das ferramentas de *Speech-to-Text*

O Quadro 1 apresenta uma comparação entre as APIs citadas nesse capítulo, no que diz respeito a *Speech-to-Text*, considerando a quantidade de idiomas suportados, plataformas-alvo, linguagem de programação e licença de uso.

3 TRABALHOS RELACIONADOS

Neste capítulo são mostrados alguns trabalhos que utilizam software de reconhecimento de voz e que se baseiam ou não em *Spaced Repetition* para ensinar, melhorar ou avaliar a pronúncia dos usuários.

Hincks (2003) realizou uma pesquisa para investigar se o uso de uma ferramenta de *Speech Recognition* ajudaria na evolução da pronúncia em um grupo de estudantes de Inglês de meia idade. Os onze estudantes tiveram acesso a uma cópia do programa *Talk to Me* (Auralog) e, como suplemento, um curso de 200 horas em inglês técnico. O progresso dos estudantes foi comparado com um grupo de controle que não utilizou o programa. Os resultados indicaram que o uso do programa melhorou a pronúncia dos estudantes que tinham fortes sotaques (HINCKS, 2003).

Neri, Cucchiarini e Strik (2003) examinaram diversas pesquisas e estudos para avaliar a usabilidade de uma ferramenta de *Speech Recognition* como forma de treinar pronúncia, além de analisar e discutir o estado da arte desse tipo de tecnologia. Em suas conclusões, eles descrevem também que uma ferramenta ideal de treinamento de pronúncia deve possuir cinco fases (*Speech Recognition*, Pontuação, Detecção de erro, Apresentação de *feedback*).

No trabalho de Baturay, Yıldırım e Daloğlu (2009) foi proposto um estudo para verificar os efeitos que uma aplicação web de *Spaced Repetition* tem sobre a retenção de vocabulário para estudantes de línguas estrangeiras. Usando uma aplicação chamada *WEBVOCLE*, os participantes foram apresentados uma lista de itens no qual usavam na aplicação. Por meio de testes se determinava o desenvolvimento dos vocabulários adquiridos. Os resultados obtidos por esse estudo mostram que o uso da aplicação é efetiva para o crescimento de retenção dos vocabulários dos participantes.

Em seu trabalho, Keder (2009) conceitua *Spacing Effect* e descreve softwares que utilizam *Spaced Repetition* com o intuito de aprender alguma informação. As aplicações citadas são SuperMemo, Anki, Mnemosyne e FullRecall. Uma aplicação baseada em *Spaced Repetition* foi criada para a Universidade de Masarykova, que permite, entre outras coisas, que usuários criem livros didáticos, que um *flashcard* seja revisado antes da data prevista, que *flashcards* os quais o usuário não tem interesse em aprender sejam excluídos, além de notificar o momento em que revisões devem ocorrer. Keder concluiu que a aplicação da ferramenta obteve sucesso, com a utilização por parte de muitos alunos da universidade, e que um estudo analisando os dados obtidos pode ser realizado no futuro.

Wallace (2016) apresenta em sua pesquisa a utilização do *Google Web Speech API* para transcrever falas dos estudantes. O estudante lê um *script* ao mesmo tempo que grava sua voz, e também faz o uso do *Google Web Speech API* para transcrever a fala. Em seguida compara o *script* original com o resultado da transcrição obtido pela *API*. Essa revisão geralmente resulta em discrepância entre os dois *scripts* que podem ser atribuídos a erros de pronúncia. Wallace pontua também que o uso efetivo da *API* só é alcançado se o estudante usar um bom microfone

para clareza das palavras. E o sotaque dos estudantes não deve diferenciar muito do modelo de reconhecimento da *API*. Os resultados indicam o benefício de usar *Google Web Speech API*. Ele oferece aos estudantes um ambiente seguro, onde somente eles julgam as suas pronúncias. Também ilustra em na transcrição gerada possíveis erros de pronúncia. Nossa proposta se diferencia dos trabalhos acima, porque tem como objetivo relacionar *Speech-to-Text* e *Spaced Repetition*, de forma a preencher a lacuna existente na área de ensino de pronúncia e retenção de vocabulário.

Outro objetivo deste trabalho é a possibilidade de um professor avaliar como seus alunos estão pronunciando os termos da computação. Assim, como a proposta de Keder (2009), ao armazenar os dados dos alunos a medida que usam a ferramenta, viabiliza uma pesquisa futura para que se analise a eficiência de uma ferramenta combina *Speech-to-Text* e *Spaced Repetition*.

4 MÉTODO

Este capítulo apresenta as etapas utilizadas para a construção de uma ferramenta que auxilie o ensino de pronúncia de língua inglesa, que faz o uso de reconhecimento voz e *spaced repetition*.

A etapa inicial constituiu-se de uma revisão na literatura sobre os métodos de aprendizagem de outra língua com o auxílio de computadores. Esta pesquisa teve como escopo apenas o aprendizado de língua inglesa para falantes de língua portuguesa.

4.1 Tecnologia de VUI

Nesta etapa foi efetuada uma análise das diferentes tecnologias de *Speech-to-text* apresentadas na seção 2.3. A *Web Speech API* foi escolhida, pois tem suporte de *Speech to Text* para a língua inglesa e possui uma licença de uso gratuita.

Após a escolha da tecnologia, uma prova de conceito do projeto, utilizando a *Web Speech API*, foi construída para verificar a usabilidade da mesma no contexto deste trabalho. Em paralelo foi decidido um nome para a ferramenta, *Pronun-Master*, que é a combinação das palavras pronúncia e do inglês *master*.

Em seguida, foi definido os termos que poderiam ser utilizados inicialmente na ferramenta, sendo estas palavras de língua inglesa em razão de que o curso de Bacharelado em Ciência da Computação utiliza inúmeras palavras em inglês. E delimitar o escopo inicial da ferramenta, desta forma decidiu-se que apenas termos que são diretamente relacionados e utilizados na computação seriam utilizados. Tais termos são normalmente usados como palavras reservadas em linguagens de programação como por exemplo: *While*, *Function* e *Null*. Termos gerais da área de computação que são frequentemente utilizados em Inglês também foram adicionados, como, por exemplo: *Mobile*, *Algorithm* e *Backup*. Os termos foram agrupados em categorias dentro da computação de forma que fossem similares ou relacionados, a lista de palavras está descrita na Tabela 1.

4.2 Análise de requisitos

De forma sucinta, a ferramenta se propõe em verificar a pronúncia de palavras em inglês do usuário e espaçar essas verificações usando *spaced repetition system*. Para atingir este objetivo é preciso que cada palavra esteja separada em um *card*, possuindo o termo em inglês, tradução, imagem e um vídeo contendo a forma correta de como pronunciar o termo. Para que um usuário não precise adicionar individualmente cada *card* em sua coleção, estes são agrupados em tópicos. A verificação da pronúncia de palavras é chamada de *review* ou revisão pela ferramenta, uma revisão acontece quando o usuário possui um ou mais *cards* disponíveis,

a disponibilidade de cada *card* é prevista pelo algoritmo de SRS. No momento da revisão de um *card*, a interface apresenta ao usuário o termo a ser pronunciado e sua imagem. Por meio de um microfone o usuário então poderá testar a sua pronúncia, uma vez que o teste terminar, a ferramenta informa se a pronúncia foi aceita ou não. Este processo se repete até que não tenha mais *cards* para serem revisados.

Esta etapa descreve as funcionalidades e responsabilidades levantadas para a construção da ferramenta. As funcionalidades arquitetadas estão listadas e descritas abaixo:

Acesso autenticado: Todos os usuários devem fazer algum tipo de acesso autenticado para utilizar dos recursos da ferramenta.

Visualização de tópicos: Deve existir uma interface dedicada onde estão listados todos os tópicos disponíveis.

Adicionar tópico: Na visualização de um tópico o usuário deve poder adicioná-lo a sua coleção. Esta ação adiciona todos os *cards* pertencentes ao tópico na lista de *cards* para revisão do usuário.

Revisão: O usuário é apresentado com o termo de um *card* e deve por meio de um microfone pronunciar-lo. Este processo ocorre até que o usuário não tenha mais *cards* disponíveis para revisão.

Criação de tópicos e *cards*: A criação de tópicos e *cards* deve ser feita por um usuário do tipo administrador, a criação deve ser realizada em uma interface distinta e sem acesso para os demais usuários.

Visualização de status: Por meio de uma interface os administradores da ferramenta podem visualizar os *cards* em que o usuário possui maior dificuldade.

Revisão espaçada dos *cards*: A ferramenta organiza o intervalo de revisão dos *cards* de forma transparente ao usuário. Este intervalo é calculado por meio de um algoritmo de SRS.

Reconhecimento de palavras: As palavras pronunciadas pelos usuários devem ser reconhecidas por um *speech-to-text*, e informar o usuário se a pronúncia foi aceita ou não.

4.2.1 Público-alvo

A ferramenta tem como público-alvo professores de inglês, que buscam algum tipo de auxílio para ensinar a pronúncia das palavras para seus alunos. Também fazem parte do público-alvo pessoas que querem aprender ou melhorar como pronunciam certas palavras inglesas.

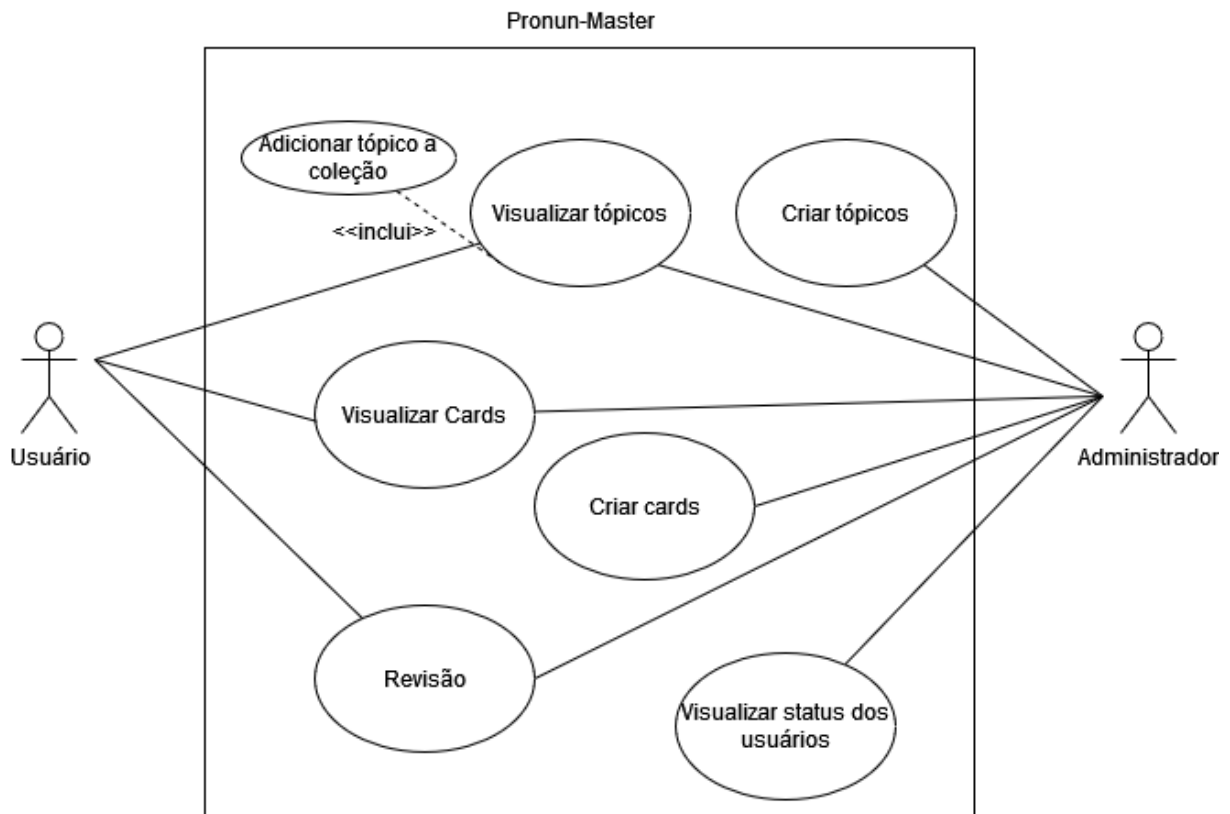
A partir da definição do público-alvo foram criadas duas personas, um recurso da área de Interação Humano-computador que descreve personagens fictícios representando usuários típicos da ferramenta:

Persona 1: Arnaldo é professor em uma escola de idiomas, ele dá aulas de inglês. Sempre está pensando em inovar e melhorar a dinâmica de sua aula para que os alunos aprendam de forma mais prática e eficiente.

Persona 2: Amanda é uma estudante universitária, está no último ano do curso de Análise e desenvolvimento de sistemas. Ao concluir o curso pretende trabalhar remotamente em uma empresa de outros país.

Para ilustrar o comportamento da ferramenta foi elaborado um diagrama de casos, que apresenta de forma resumida os papéis dos usuários e as funcionalidades da ferramenta na Figura 10.

Figura 10 – Diagrama de casos de uso do sistema



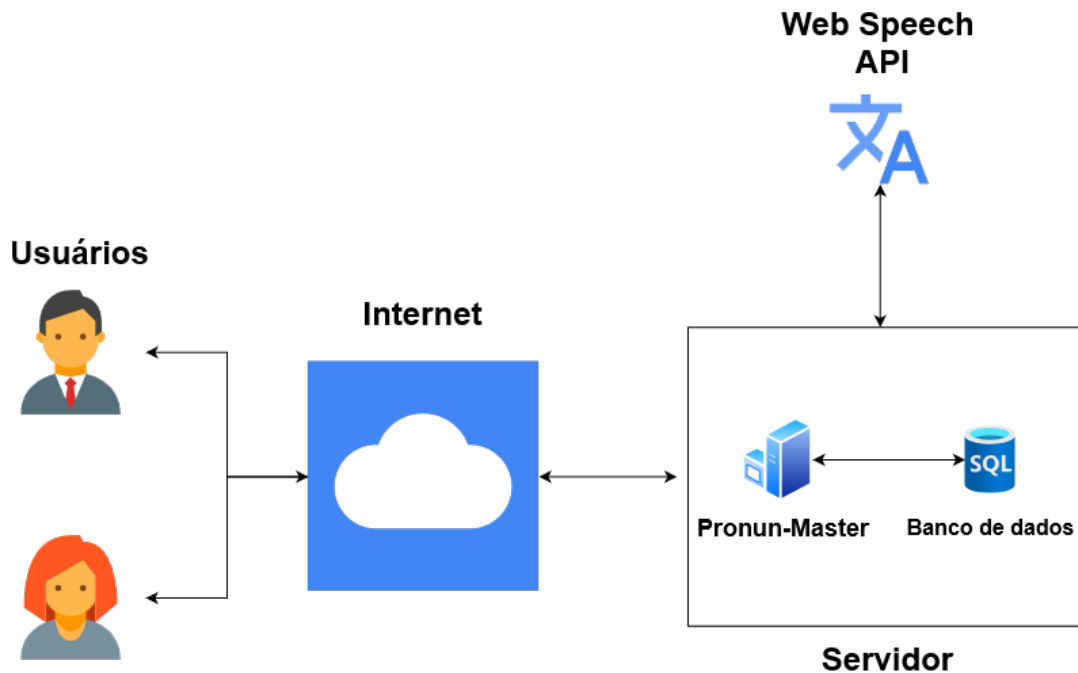
Fonte: Autoria própria (2022).

4.3 Arquitetura e modelagem do banco de dados

A ferramenta *Pronun-Master* foi desenvolvida para ter uma interface web, sendo acessível sem fazer qualquer tipo de instalação de *software*. A Figura 11 representa a arquitetura

projetada para o sistema, composta por usuários que por meio da internet interagem com a ferramenta através de uma interface web. Esta interface está hospedada em um servidor remoto, a mesma também possui integração com um banco de dados, para persistir os dados dos usuários, e se comunica com a *Web Speech API* para que seja feito o reconhecimento da pronúncia dos termos em inglês.

Figura 11 – Arquitetura do sistema



Fonte: Autoria própria (2022).

4.3.1 Modelagem do banco de dados

Devido a utilização de *Spaced Repetition* é necessário que a ferramenta persista os dados de cada *review* e, conseqüentemente, de cada usuário em um banco de dados. O Sistema de Gerenciamento de Banco de Dados (SGBD) escolhido foi o PostgreSQL (POSTGRESQL, 2022), pela sua robustez e por ser código aberto.

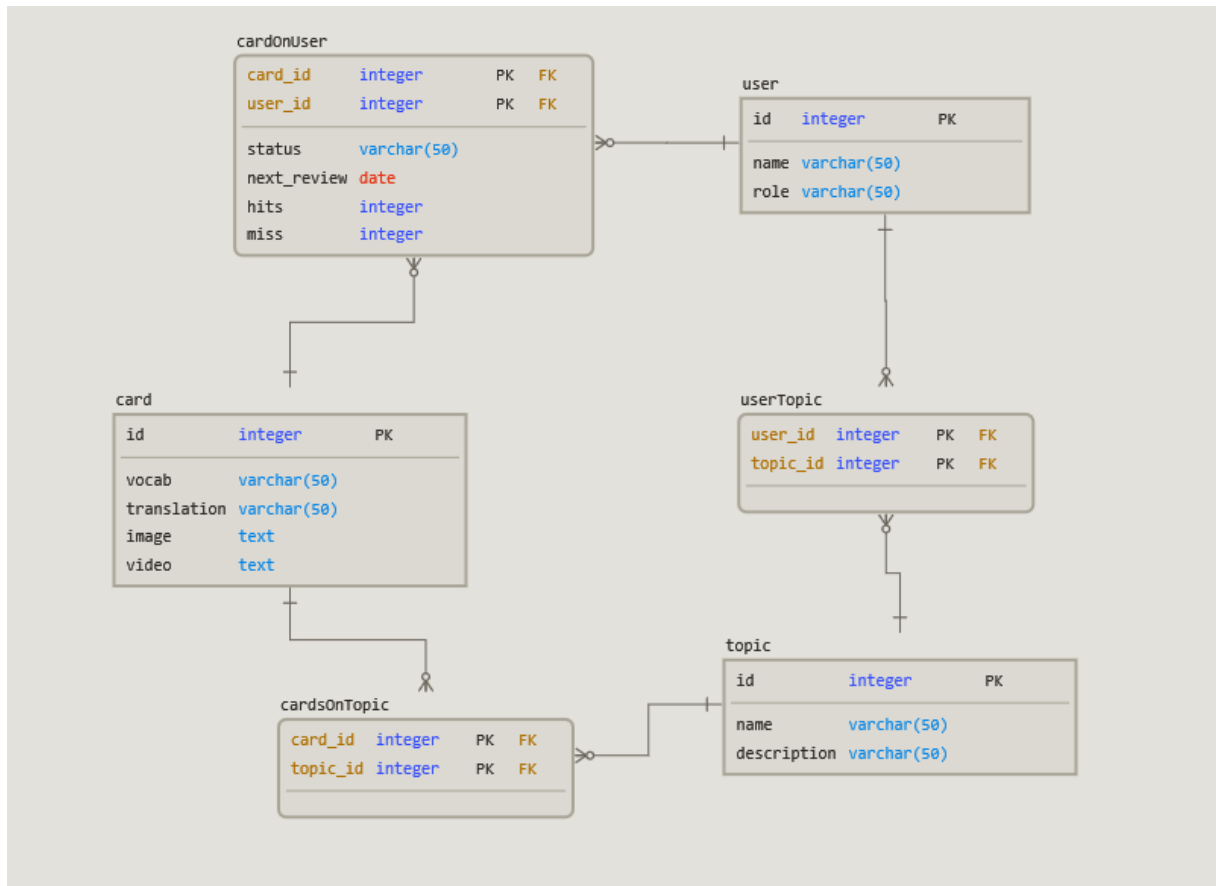
O modelo do banco de dados foi criado baseado nas funcionalidades descritas acima e seu modelo relacional pode ser visto na Figura 12.

Como cada usuário pode possuir mais de um tópico, e como cada *card* deve conter informações de acertos e erros de cada usuário, foi necessário criar um relacionamento com cardinalidade N:N entre duas entidades. Este relacionamento entre *cards* e *user* pode ser visto na Figura 12, representado pela tabela de nome *cardsOnUser*.

Após a conclusão da etapa de modelagem a plataforma *Heroku*¹ foi utilizada para hospedar o banco de dados.

¹ www.heroku.com

Figura 12 – Diagrama Entidade Relacionamento



Fonte: Autoria própria (2022).

Após a modelagem do banco de dados, se deu início a construção da interface web. Esta foi construída utilizando o *framework Next.js*², por ser código aberto, por ser feito para ser utilizado com a linguagem *javascript*, a mesma da *Web Speech API*, como também a sua facilidade de configuração. Em seguida a ferramenta foi hospedada na plataforma *Vercel*³, que facilita a hospedagem de projetos criados com o *Next.js*, por ser a idealizadora do mesmo. O acesso a ferramenta é feito pelos usuários por meio do seguinte endereço eletrônico <https://pronun-master.vercel.app/>⁴.

4.4 Algoritmo de *Spaced Repetition*

O algoritmo de *Spaced Repetition* utilizado na ferramenta é uma adaptação do algoritmo utilizado no *Leitner System*, empregando o mesmo conceito de separação dos *cards* em grupos, porém cada grupo possui três subgrupos. Esta adaptação visa ser mais tolerante a erros cometidos em *cards* que se encontram em grupos mais avançados, visto que ao errar um *card*

² <https://nextjs.org/>

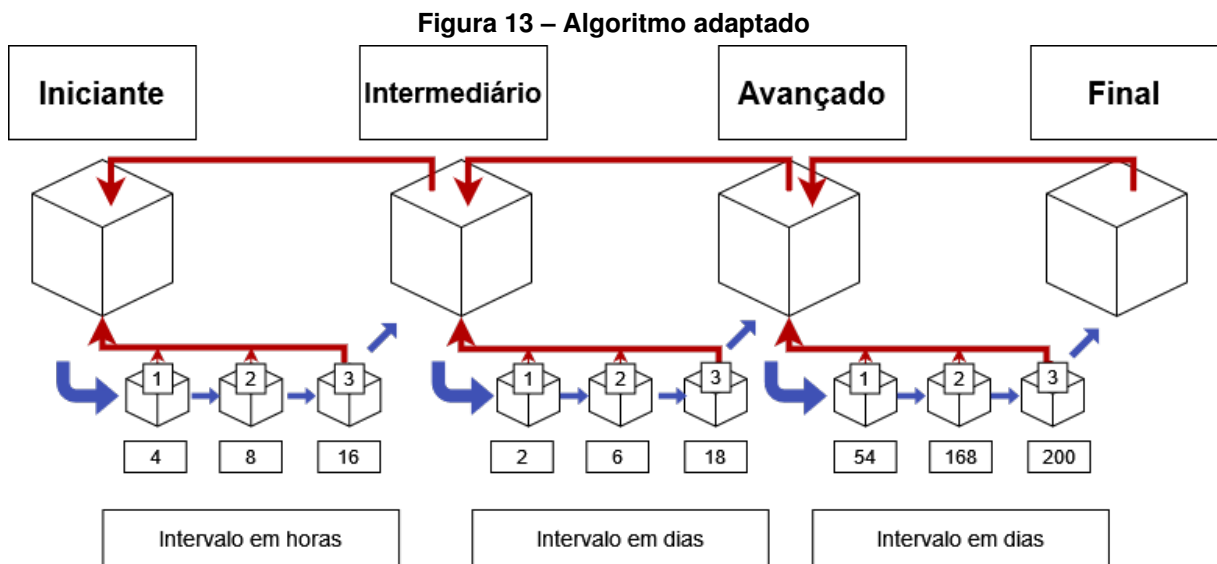
³ <https://vercel.com/>

⁴ <https://pronun-master.vercel.app/>

do penúltimo grupo o mesmo não retorna para o grupo inicial. Desta forma mantém os primeiros grupos com os *cards* que o usuário possui mais dificuldade.

A Figura 13 corresponde ao algoritmo adaptado, contendo quatro grupos em que o espaçamento de cada revisão é dobrado a cada resposta marcada como correta para os subgrupos do primeiro grupo ex: 4 horas, 8 horas, 16 horas. Para os outros grupos este espaçamento é triplicado a cada resposta correta.

Quando uma revisão acontece existem quatro possibilidades para o *card*, caso a resposta esteja correta este pode: ir para o próximo subgrupo; caso já esteja no último subgrupo e é movido para o primeiro subgrupo do próximo grupo. Se a resposta for incorreta o *card* pode: ser movido para o grupo anterior caso estiver no primeiro subgrupo do grupo atual; se não estiver no primeiro então é movido para o primeiro subgrupo. Quando o *card* chegar no último grupo não há mais necessidade de ser revisado.



Fonte: Autoria própria (2022).

O Algoritmo 4 apresenta uma iteração do processo de revisão de um *card*.

Algoritmo 4 – Pseudo código do algoritmo Adaptado

inserir *NGrupos, Intervalo*

```
1: se Intervalo == IntervaloDoSGrupo então  
2:   Fazer revisão de um Flashcard  
3:   se Resposta == Correta então  
4:     se posicaoSubGrupo != ultimaPosicao então  
5:       ProximoSubGrupo := flashcard  
6:     senão,  
7:       ProximoGrupo := flashcard  
8:     finaliza se  
9:   senão,  
10:    se posicaoSubGrupo == primeiraPosicao então  
11:      GrupoAnterior := flashcard  
12:    senão,  
13:      subGrupoAnterior := flashcard  
14:    finaliza se  
15:  finaliza se  
16: finaliza se
```

Fonte: Autoria própria (2022).

5 RESULTADOS

Como o sistema é baseado em *spaced repetition* e o uso de *flashcards*. O site foi desenvolvido de forma a fazer o uso de *flashcards* em formato digital, tendo como inspiração outras ferramentas semelhantes existentes como o ANKI.

A interface inicial após o acesso autenticado do *Pronun-Master* exibida na Figura 14. Esta interface apresenta ao usuário uma breve explicação das funcionalidades da ferramenta. A explicação descreve o que é e como funciona *Spaced Repetition*, como escolher um tópico dentro da ferramenta. E quando e como realizar a revisão dos *cards*. E também serve para acessar as outras funcionalidades disponíveis.

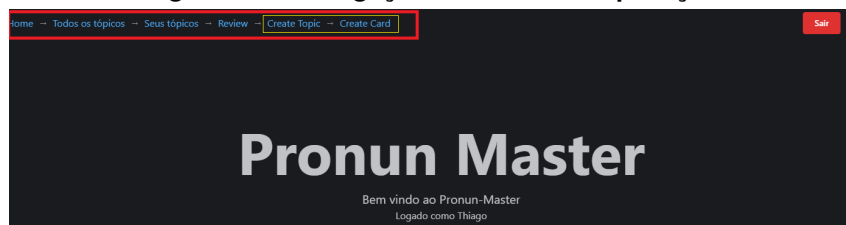
Figura 14 – Interface inicial Pronun-Master



Fonte: Autoria própria (2022).

No canto superior esquerdo da interface se encontra o menu de navegação do *Pronun-Master*, destacado em vermelho na Figura 15. Em amarelo são destacadas apenas funcionalidades disponíveis para os administradores. Todas as interfaces da ferramenta possuem esta navegação.

Figura 15 – Navegação estrutural da aplicação

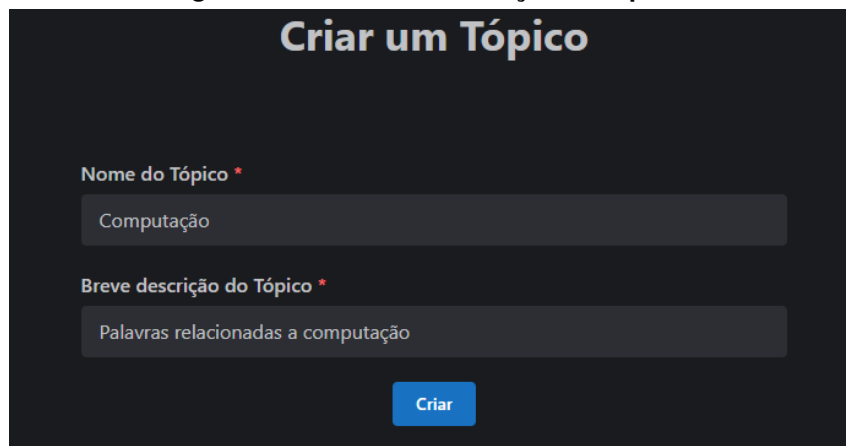


Fonte: Autoria própria (2022).

As funcionalidades de criação de *cards* e tópicos são exclusivas para os administradores da ferramenta. A Figura 16 apresenta um tópico sendo criado. Sendo necessário apenas o nome do tópico e uma descrição.

A criação de um *card* se dá de forma parecida a criação de um tópico. Um administrador deve preencher os campos obrigatórios, sinalizados por um asterisco em vermelho. A Figura 17 demonstra este processo criação para o termo *keyboard*.

Figura 16 – Interface de criação de tópicos



Criar um Tópico

Nome do Tópico *

Computação

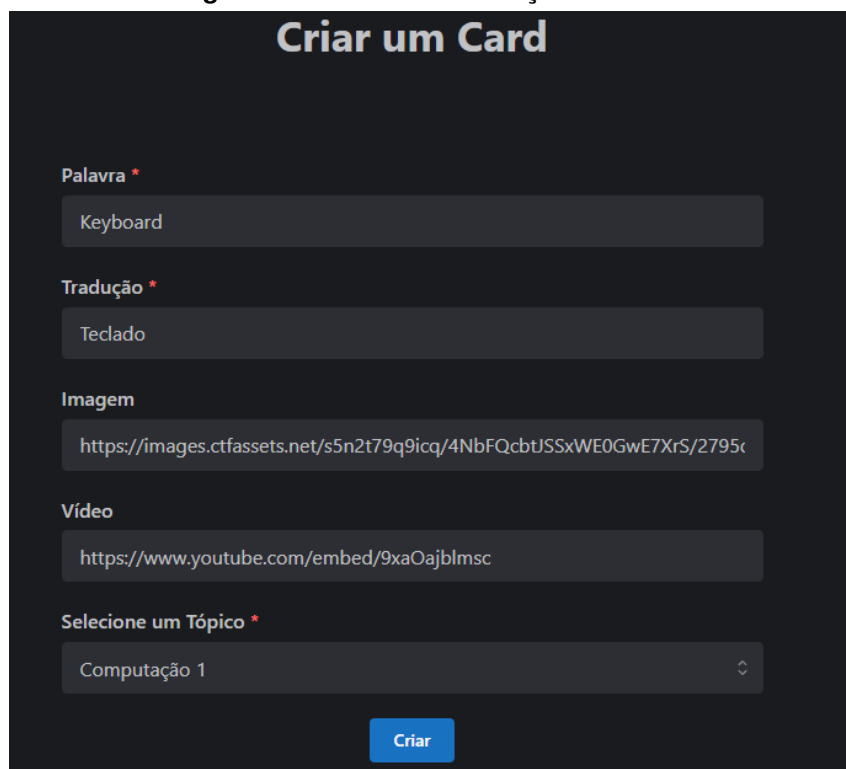
Breve descrição do Tópico *

Palavras relacionadas a computação

Criar

Fonte: Autoria própria (2022).

No campo de imagem o usuário fornece um *link* de onde a imagem está hospedada. O mesmo acontece para o campo de vídeo, porém neste caso apenas *links* do *YouTube*¹ são aceitos pela ferramenta. Um *card* precisa sempre ser associado a um tópico já existente.

Figura 17 – Interface de criação de *cards*

Criar um Card

Palavra *

Keyboard

Tradução *

Teclado

Imagem

<https://images.ctfassets.net/s5n2t79q9icq/4NbFQcibtUSSxWE0GwE7XrS/2795c>

Vídeo

<https://www.youtube.com/embed/9xaOajblmsc>

Selecione um Tópico *

Computação 1

Criar

Fonte: Autoria própria (2022).

Os administradores também podem fazer modificações no *cards* e em tópicos, como demonstrado na Figura 18 que demonstra a edição do *card* de *Cloud Computing*. Porém não é possível mover os *cards* para outros tópicos, porque esta ação pode fazer com que usuários tenham *cards* de tópicos que eles não possuem na coleção.

¹ <https://www.youtube.com/>

Figura 18 – Edição de um *card*

Fonte: Autoria própria (2022).

Outra funcionalidade que está disponível apenas para os administradores da ferramenta, é a possibilidade de visualizar quais os *cards* que cada usuário possui maior dificuldade. Na Figura 19 a interface exibe um seletor, que ao selecionar um usuário informa cinco *cards*. Sendo estes os *cards* em que o usuário possui a maior taxa de pronúncia marcadas como incorretas.

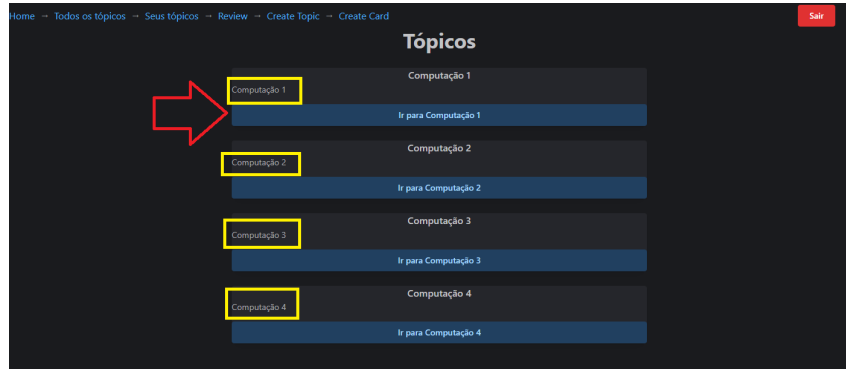
Figura 19 – Interface status de erros dos usuários

Fonte: Autoria própria (2022).

Na interface de “Todos os tópicos” são exibidos os tópicos disponíveis para os usuários. A Figura 20 mostra uma lista contendo todos os tópicos. Destes cada um possui um nome e

uma breve descrição, destacado em amarelo. O usuário pode ver os *cards* contidos em um tópico clicando no botão de “ir para o tópico”, destacado com uma seta vermelha.

Figura 20 – Lista de Tópicos

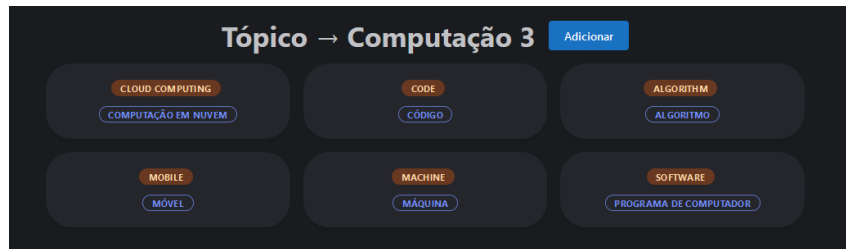


Fonte: Autoria própria (2022).

A Figura 21 apresenta os detalhes do tópico “Computação 3”. A interface apresenta um botão em azul ao lado do nome do tópico, que ao pressionar este botão o usuário salva o tópico em sua coleção. Quando um tópico é adicionado, todos os *cards* associados a ele ficam aptos para serem revisados pelo usuário no mesmo instante.

Os *cards* listados dentro de um tópico apresentam o nome do termo em laranja e sua tradução em azul. Ao clicar em um dos *cards*, o usuário é redirecionado para uma interface contendo uma visão detalhada do mesmo.

Figura 21 – Tópico Computação 3 e seus cards



Fonte: Autoria própria (2022).

A Figura 22 apresenta uma visão detalhada de um *card*, *Cloud Computing*. Nesta interface são exibidas informações como: vocabulário, quantidade de vezes que o usuário acertou e errou a pronúncia do termo, data da próxima revisão, tradução do termo, como também a imagem ilustrativa do termo e o vídeo com a pronúncia correta.

Na página de *reviews* a aplicação recupera do banco de dados todos os *cards* prontos para serem revisados que o usuário possui. Apenas algumas informações do *card* aparecem como: Vocabulário, imagem, quantidade de vezes que ele acertou e errou uma revisão deste *card* como mostra a Figura 23.

O reconhecimento de voz só começa quando o botão em verde “Revisar palavra” é clicado. Neste momento um indicador animado em formato de microfone aparece com a instrução para que o usuário comece a falar, essa interface é demonstrada na Figura 24 com a revisão

Figura 22 – Visão detalhada de um cards

Fonte: Autoria própria (2022).

do termo *Cloud Computing*. Neste momento se o usuário não falar nada, ou não estiver com um microfone conectado a revisão é marcada como uma falha, e conta no *card* como uma resposta errada.

Figura 23 – Revisão

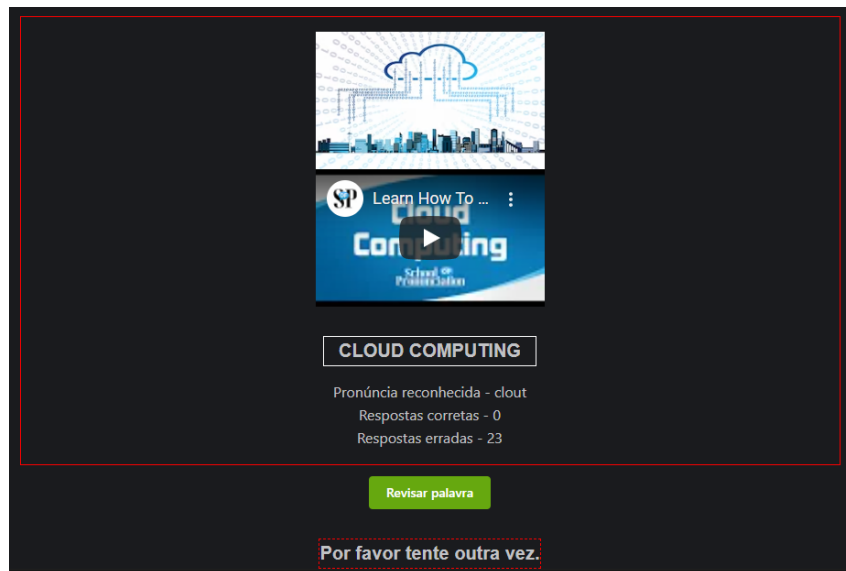
Fonte: Autoria própria (2022).

Figura 24 – Revisão

Fonte: Autoria própria (2022).

A Figura 25 apresenta um caso no qual o usuário fez a revisão de um *card*, porém a pronúncia não foi aceita como correta para o termo. A transcrição da palavra reconhecida pela *Web Speech API* é apresentada na interface. Assim o usuário sabe a razão de ter falhado na revisão. Após a falha o mesmo *card* é mantido na interface. No entanto, neste momento o vídeo com a pronúncia considerada correta é apresentado para que o usuário.

Figura 25 – Pronúncia não aceita



Fonte: Autoria própria (2022).

5.1 Discussões

A ferramenta está hospedada em um servidor web e faz chamadas para a *Web Speech API*, além de armazenar os dados dos usuários, assim para o uso da ferramenta é preciso que o usuário tenha acesso ao computador com internet.

A ferramenta exige também o uso de um microfone para as revisões. Wallace (2016) recomenda o uso de um bom microfone para evitar a captação de sons indesejados que possam atrapalhar o reconhecimento feito pela API. A configuração de *Speech-to-Text* da *Web Speech API* foi feita de modo a reconhecer palavras em inglês padrão dos Estados Unidos. Portanto a pronúncia considerada correta pela ferramenta é baseada neste sotaque e forma de falar.

6 CONCLUSÃO

Neste trabalho foi criada uma ferramenta que utiliza *Speech-to-Text* e *Spaced repetition* para ajudar no ensino de pronúncia de palavras de língua inglesa, por meio de *flashcards* digitais. A ferramenta oferece de forma gratuita uma maneira para que estudantes possam aprender e melhorar a pronúncia de palavras em inglês da área da computação. Para professores, a ferramenta fornece um meio de identificar possíveis dificuldades de pronúncias.

Por ser online, a ferramenta requer apenas um computador com acesso a internet e um microfone. Além de permitir que os usuários acessem e façam o uso da ferramenta em qualquer horário. Como os *cards* podem ser divididos em diversos tópicos, um professor pode criar vários tópicos específicos para diferentes turmas e diferentes interesses.

Em trabalhos futuros, a ferramenta pode ser utilizada para pesquisas com ênfase em termos técnicos de outra área, ou também para o ensino de pronúncia de palavras do dia a dia da língua inglesa.

Visto que a *Web Speech API* permite que a mesma seja configurada para reconhecer palavras em outros idiomas, pesquisas que exploram o uso de outros idiomas também podem ser realizadas. Dos idiomas que a *Web Speech API* é capaz de transcrever, um deles é o português, portanto trabalhos futuros não precisam ser limitados somente ao ensino de inglês como segundo idioma. Tais trabalhos podem ser realizados como o ensino de português para estrangeiros. E até mesmo na área de saúde, mais especificamente fonoaudiologia, em que um profissional pode utilizar a ferramenta para ajudar um paciente a melhorar a sua pronúncia.

Outros futuros trabalhos podem também abranger a utilização de outros algoritmos de *Spaced Repetition*. Como por exemplo, pode-se adaptar o algoritmo de SM-2 para que o parâmetro de qualidade no qual determina o intervalo para a próxima revisão, seja o grau de confiança retornado pela *Web Speech API* a cada palavra testada para o reconhecimento.

A ferramenta também pode ser modificada de modo a comportar outros algoritmos de *Spaced Repetition*. Assim uma pesquisa para analisar qual algoritmo é melhor ou mais eficiente, no ensino de pronúncias de um segundo idioma.

REFERÊNCIAS

- ASSIS, A. B. G. d. **Adaptações fonológicas na pronúncia de estrangeirismos do Inglês por falantes de Português Brasileiro**. 2007. Dissertação (Mestrado) — Universidade Estadual Paulista (UNESP), 2007.
- ATKINSON, R. C. Optimizing the learning of a second-language vocabulary. **Journal of experimental psychology**, American Psychological Association, v. 96, n. 1, p. 124, 1972.
- BATURAY, M.; YILDIRIM, S.; DALOĞLU, A. Effects of web-based spaced repetition on vocabulary retention of foreign language learners. **Eurasian Journal of Educational Research (EJER)**, n. 34, 2009.
- BLOOM, K. C.; SHUELL, T. J. Effects of massed and distributed practice on the learning and retention of second-language vocabulary. **The Journal of Educational Research**, Taylor & Francis, v. 74, n. 4, p. 245–248, 1981.
- CHUKHAREV-HUDILAINEN, E.; KLEPIKOVA, T. A. The effectiveness of computer-based spaced repetition in foreign language vocabulary instruction: A double-blind study. **calico journal**, v. 33, n. 3, p. 334, 2016.
- EBBINGHAUS, H. Memory: A contribution to experimental psychology. **Annals of neurosciences**, Karger Publishers, v. 20, n. 4, p. 155, 2013, reimpressão.
- FINI, A. A. S.; ZAINALIPOUR, H.; MOZAFARI, S. H. Comparing the effect of mass and distributed study techniques on remembering level and its relation with the exam date. **Procedia-Social and Behavioral Sciences**, Elsevier, v. 5, p. 1751–1759, 2010.
- GODWIN-JONES, R. Mobile apps for language learning. **Language Learning & Technology**, University of Hawaii National Foreign Language Resource Center, v. 15, n. 2, p. 2–11, 2011.
- GOLDSTINE, H. H.; GOLDSTINE, A. The electronic numerical integrator and computer (eniac). **Mathematical Tables and Other Aids to Computation**, JSTOR, v. 2, n. 15, p. 97–110, 1946.
- GRADDOL, D. **The future of English?: a guide to forecasting the popularity of the English language in the 21st century**. [S.l.]: The British Council, 2000.
- GURU, S. M. **Leitner system**. 2018. **Disponível em:** https://supermemo.guru/wiki/Leitner_system. **Acesso em:** 10 fev. 2022.
- HINCKS, R. Speech technologies for pronunciation feedback and evaluation. **ReCALL**, Cambridge University Press, v. 15, n. 1, p. 3–20, 2003.
- KANG, S. H. Spaced repetition promotes efficient and effective learning: Policy implications for instruction. **Policy Insights from the Behavioral and Brain Sciences**, SAGE Publications Sage CA: Los Angeles, CA, v. 3, n. 1, p. 12–19, 2016.
- KEDER, D. **Computer-assisted language learning using spaced repetition**. 2009. Tese (Doutorado) — Masarykova univerzita, Fakulta informatiky, 2009.
- KRUG, D.; DAVIS, T. B.; GLOVER, J. A. Massed versus distributed repeated reading: A case of forgetting helping recall? **Journal of Educational Psychology**, American Psychological Association, v. 82, n. 2, p. 366, 1990.

LANTEIGNE, B. Common, persistent errors in English by Brazilian Portuguese speakers. **TEFL Web Journal**, v. 4, n. 1, p. 80, 2006.

LEITNER, S. **So Lernt Man Lernen**. [S.l.: s.n.], 1972.

LINDSEY, R. V. *et al.* Improving students' long-term knowledge retention through personalized review. **Psychological science**, Sage Publications Sage CA: Los Angeles, CA, v. 25, n. 3, p. 639–647, 2014.

MARAGLIA, P. H.; PEIXOTO, M. A. P. **Aprendiz em Saúde**. 2017. **Disponível em:** <https://oaprendizemsaude.wordpress.com/2017/03/22/o-que-sao-flashcards-parte-i/>. **Acesso em:** 14 dez. 2021.

METZLER-BADDELEY, C.; BADDELEY, R. J. Does adaptive training work? **Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition**, Wiley Online Library, v. 23, n. 2, p. 254–266, 2009.

NERI, A.; CUCCHIARINI, C.; STRIK, H. Automatic speech recognition for second language learning: how and why it actually works. *In: Proceedings of international congress of phonetic science*. [S.l.: s.n.], 2003. p. 1157–1160.

PIMSLEUR, P. A memory schedule. **The Modern Language Journal**, JSTOR, v. 51, n. 2, p. 73–75, 1967.

POSTGRESQL. 2022. **Disponível em:** <https://www.postgresql.org/>. **Acesso em:** 04 abr. 2022.

SCHMIDT, R. A.; WRISBERG, C. A. **Motor learning and performance: A situation-based learning approach**. [S.l.]: Human kinetics, 2008.

SELLINK, A.; VERHOEF, C. *et al.* **Reflections on the evolution of COBOL**. [S.l.]: Universiteit van Amsterdam. Programming Research Group, 1997.

SUPERMEMO. **SM-2 Super Memo**. 2015. **Disponível em:** <https://www.supermemo.com/en/archives1990-2015/english/ol/sm2>. **Acesso em:** 10 fev. 2022.

TABIBIAN, B. *et al.* Enhancing human learning via spaced repetition optimization. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 116, n. 10, p. 3988–3993, 2019.

WALLACE, L. Using google web speech as a springboard for identifying personal pronunciation problems. *In: Proceedings of the 7th Pronunciation in Second Language Learning and Teaching Conference, ISSN*. [S.l.: s.n.], 2016. p. 2380–9566.

WEXELBLAT, R. L. **History of programming languages**. [S.l.]: Academic Press, 2014.

WRANX. **What is the Ebbinghaus forgetting curve?** 2019. **Disponível em:** <https://blog.wranx.com/ebbinghaus-forgetting-curve>. **Acesso em:** 10 dez. 2021.

APÊNDICE A – Lista inicial de palavras usadas na ferramenta

Tabela 1 – Lista de termos.

	Termo	Tradução
1	Stack	Pilha
2	Queue	Fila
3	Insert	Inserir
4	Thread	-
5	Struct	-
6	Class	Classe
7	Exception	Exceção
8	Cloud Computing	Computação em nuvem
9	Backup	Cópia de segurança
10	Thermal Throttling	Controle térmico
11	Database	Banco de dados
12	Dataset	Conjunto de dados
13	Query	Consulta
14	Bug	-
15	Foreign key	Chave estrangeira
16	Begin	Ínicio
17	End	Fim
18	Block	Bloco
19	Indentation	Indentação
20	Repeat	Repetir
21	Semicolon	Ponto e vírgula
22	Parentheses	Parênteses
23	Underline	Sublinhado
24	Boldface	Negrito
25	Italic	Itálico
26	Font	Fonte textual
27	Timeline	Linha do tempo
28	Follow	Seguir
29	Follower	Seguidor
30	Procedure	Procedimento
31	Function	Função
32	Parameter	Parâmetro
33	Command line	Linha de comando
34	Graphical	Gráfico
35	User	Usuário
36	Window	Janela
37	Menu	Menu
38	Status	Status
39	Button	Botão
40	Check	Verificar
41	Radio button	Botão de opção
42	Form	Formulário
43	Submit	Enviar
44	Cancel	Cancelar
45	Manager	Gerenciador
46	Management	Gerenciamento
47	System	Sistema
		Continua na próxima página

Tabela 1 – continuação da página anterior

	Termo	Tradução
48	Client	Cliente
49	Server	Servidor
50	Communication	Comunicação
51	Protocol	Protocolo
52	While	Enquanto
53	Do	Faça
54	For	Para cada
55	Each	Cada
56	Selection	Seleção
57	Data Structure	Estrutura de dados
58	Computer	Computador
59	Mouse	-
60	Keyboard	Teclado
61	Setting	Configuração
62	Fan	Ventoinha
63	Memory	Memória
64	Code	Código
65	Algorithm	Algoritmo
66	Program	Programa
67	Icon	Ícone
68	Language	Linguagem
69	Screen	Tela
70	Touch screen	Tela sensível ao toque
71	Sensor	Sensor
72	Comment	Comentário
73	Test	Teste
74	Review	Revisão
75	Version	Versão
76	Schedule	Agenda
77	Scheduler	Escalonador
78	Pipeline	-
79	Processor	Processador
80	Instruction	Instrução
81	Set	Colocar
82	Null	Nulo
83	Pointer	Ponteiro
84	Void	Vazio
85	Garbage	Lixo
86	Method	Método
87	Attribute	Atributo
88	Integer	Inteiro
89	Real	Número Real
90	Float	Número de ponto flutuante
91	Upload	-
92	Download	-
93	Hacker	-
94	Bandwidth	Largura de banda
		Continua na próxima página

Tabela 1 – continuação da página anterior

	Termo	Tradução
95	Network	Rede de computadores
96	Hardware	-
97	Software	Programa de computador
98	Cluster	-
99	Link	Link
100	Click	Clique
101	List	Lista
102	Greedy	(Algoritmo) Guloso
103	Dynamic programming	Programação dinâmica
104	State	Estado
105	Machine	Máquina
106	Statement	Declaração
107	Scientific	Científico
108	Interface	Interface
109	Human	Humano
110	Drive	Unidade de armazenamento
111	Disk	Disco
112	Hard disk	Unidade de disco rígido
113	Artificial intelligence	Inteligência artificial
114	Security	Segurança
115	Router	Roteador
116	Proxy	-
117	Address	Endereço
118	Smartphone	-
119	Mobile	Móvel
120	Folder	Pasta
121	Copy	Cópia
122	Paste	Colar
123	Cut	Recortar
124	Layer	Camada
125	Device	Dispositivo