



COORDENADORIA DE ENGENHARIA DE COMPUTAÇÃO

VINÍCIUS VIEIRA TOZZI

**APLICATIVO PARA APRENDIZADO DE LÍNGUAS UTILIZANDO
REPETIÇÃO ESPAÇADA**

Sorocaba/SP

2019

Vinícius Vieira Tozzi

**APLICATIVO PARA APRENDIZADO DE LÍNGUAS UTILIZANDO
REPETIÇÃO ESPAÇADA**

Trabalho de conclusão de curso apresentado ao Centro Universitário Facens como exigência parcial para a obtenção do diploma de graduação em Engenharia de Computação.

Orientador: Wilson Roberto Marcondes
Oliveira Júnior

**Sorocaba/SP
2019**

Vieira Tozzi, Vinícius

Aplicativo para aprendizado de línguas utilizando repetição espaçada/ por
Vinícius Vieira Tozzi. – Sorocaba/SP, 2019

Trabalho de Conclusão de Curso (Graduação) – Centro Universitário Facens
– Curso de Engenharia da Computação, 2018.

Orientador: Prof. Wilson Roberto Marcondes Oliveira Júnior

1. Aprendizado. 2. Memória. 2. SRS. I. Trânsito. II. Wilson Roberto Marcondes
Oliveira Júnior. III. Centro Universitário Facens.

CDD XXXX

Vinícius Vieira Tozzi

Aplicativo para aprendizado de línguas utilizando repetição espaçada

Trabalho de conclusão de curso apresentado ao Centro Universitário Facens como exigência parcial para a obtenção do diploma de graduação em Engenharia de Computação.

Sorocaba/SP, / / .

Orientador **Wilson Roberto Marcondes
Oliveira Júnior**

Professor Convidado 1

Professor Convidado 2

AGRADECIMENTOS

RESUMO

ABSTRACT

LISTA DE ILUSTRAÇÕES

Figura 3.1 – Curva de esquecimento	15
Figura 3.2 – Aprendizado espaçado aumenta sobrevivência de células no hipocampo	16
Figura 3.3 – Método das caixas	17
Figura 3.4 – Quantidade da população que possui <i>Smartphones</i>	18
Figura 4.1 – Duolingo	20
Figura 4.2 – Super Memo	21
Figura 4.3 – Anki	22
Figura 5.1 – <i>Programação declarativa</i>	25
Figura 5.2 – <i>BLoC Pattern</i>	25
Figura 5.3 – Arquitetura Geral	26
Figura 5.4 – Redux	26

LISTA DE ABREVIATURAS E SIGLAS

ETS	<i>Educational Testing Service</i>
TOEFL	<i>Test of English as a Foreign Language</i>
SRS	<i>Spaced Repetition System</i>
SM	<i>Super Memo</i>
CALL	<i>Computer Assisted Language Learning</i>
MALL	<i>Computer Assisted Language Learning</i>
SDK	<i>Software Development Kit</i>
UI	<i>User Interface</i>
BLOC	<i>Business Logic Component</i>
API	<i>Application Programming Interface</i>

SUMÁRIO

1	INTRODUÇÃO	11
2	APRENDIZADO DE UMA NOVA LÍNGUA	12
2.1	Dificuldades Ao Aprender Uma Nova Língua	12
3	METODOLOGIAS PARA MEMORIZAÇÃO DE VOCABULÁRIO . . .	14
3.1	Método Dos <i>Flashcards</i>	14
3.2	Curva De Esquecimento E Aprendizado	14
3.3	SRS - <i>Spaced Repetition System</i>	16
3.4	CALL - Computer Assisted Language Learning	17
3.5	Aprendizado Intencional De Vocabulário E Aprendizado Involun- tário	19
4	PROPOSTA DA PLATAFORMA	20
4.1	Aplicações Semelhantes	20
4.1.1	Duolingo	20
4.1.2	Super Memo	21
4.1.3	Anki	21
4.2	Flashcard com leitura interativa	22
5	ESPECIFICAÇÕES TÉCNICAS E ARQUITETURA	24
5.1	Flutter	24
5.2	Dart	24
5.3	Arquitetura	24
5.4	Backend	26
6	CONCLUSÃO	28

1 INTRODUÇÃO

2 APRENDIZADO DE UMA NOVA LÍNGUA

O aprendizado de línguas estrangeiras é bastante relevante no mundo moderno graças a fatores como a globalização, internet e mercado de trabalho. Pesquisas indicam que a procura pelo aprendizado de novas línguas está aumentando nos últimos anos no âmbito escolar. Dados providos pelo *Institute of Education Sciences* (2002) mostram que escolas primárias aumentaram em 40% o número de horas dedicadas ao ensino de uma língua estrangeira entre os anos 1990 e 2000.

Com o mundo cada vez mais globalizado e inúmeras empresas multinacionais, falar uma ou mais línguas estrangeiras tornou-se uma vantagem competitiva no mercado de trabalho, pois indica uma maior capacidade de comunicação e permite maiores possibilidades de estudo e capacitação em instituições de ensino em países estrangeiros. Além de tais vantagens, estudos mostram que existem diferenças fisiológicas no cérebro de pessoas que aprendem uma segunda língua. Um estudo publicado na revista científica *Brain and Language* (2014) mostra que existe um melhor desenvolvimento do cérebro quando se aprende uma língua estrangeira e outro estudo publicado na revista *NeuroImage* (2012) mostra que há diferenças estruturais na fisiologia do cérebro de pessoas enquanto elas estão aprendendo uma nova língua.

2.1 DIFICULDADES AO APRENDER UMA NOVA LÍNGUA

Aprender uma língua é uma tarefa complexa que envolve estudo de conceitos teóricos tais como a gramática, conceitos temporais, casos e exceções, e também exige um número relevante de horas práticas e treino para estimular a memorização do vocabulário. Para muitas instituições e escolas de proficiência como a ETS (*Educational Testing Service*) que providencia o TOEFL (*Test of English as a Foreign Language*) a fluência em uma língua está diretamente ligada ao seu domínio dela em quatro aspectos: escrita, fala, interpretação de texto e compreensão auditiva.

Alguns dos desafios mais comuns e que se tornam empecilhos no processo de aprendizagem são as regras gramaticais e as semânticas, que são basicamente um conjunto de regras que o aluno deve conhecer para saber como a linguagem funciona estruturalmente, além é claro, do conhecimento necessário de vocabulário, que por sua vez requer memorização de uma grande quantidade de palavras.

É importante mencionar que não é necessário conhecer todas as palavras da língua para ser fluente, mas ainda assim é um longo percurso a se percorrer, uma vez que no dicionário Oxford da Língua Inglesa, por exemplo, há 133.826 palavras, excluindo as palavras que não são mais utilizadas. Uma pesquisa feita pela equipe

de pesquisas independentes criadora do site *Test Your Vocab* (2013) mostra que uma pessoa fluente na língua inglesa conhece em torno de 20.000 a 35.000 palavras, o que pode ser considerado um número considerável de palavras a serem memorizadas.

3 METODOLOGIAS PARA MEMORIZAÇÃO DE VOCABULÁRIO

Existem diversos métodos para auxiliar na memorização de novo vocabulário como por exemplo: realizar a leitura da definição de uma palavra repetidas vezes, escrever múltiplas vezes ou até mesmo repetir em voz alta essa palavra e seu significado. O que algumas pesquisas indicam é que repetir o vocabulário enquanto executa algum tipo de exercício físico (escrita ou fala por exemplo) aumenta a capacidade de retenção dos significados das palavras.

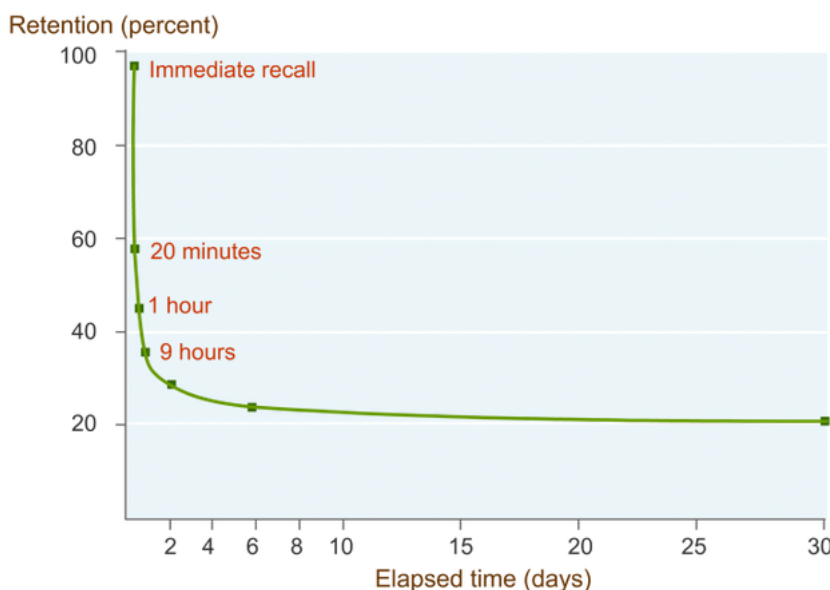
3.1 MÉTODO DOS *FLASHCARDS*

Um dos métodos mais populares atualmente se baseia em utilizar cartões ou pedaços de papel (*flashcards*) onde de um lado do papel está escrito a palavra que se deseja memorizar na língua estrangeira e do outro lado do papel está a tradução da palavra. A ideia é que o estudante tente lembrar da tradução da palavra sem precisar virar o cartão e para verificar se está correto ele pode facilmente consultar a resposta.

3.2 CURVA DE ESQUECIMENTO E APRENDIZADO

Em seu livro *Über das Gedächtnis - Memory: A Contribution to Experimental Psychology* o autor Hermann Ebbinghaus (1885 apud TUMELERO, 2018) analisa de forma empírica como as pessoas esquecem a informação depois de adquiri-la. A conclusão obtida foi de que o tempo de esquecimento se comporta de forma exponencial, também chamada de curva de esquecimento.

Figura 3.1 – Curva de esquecimento



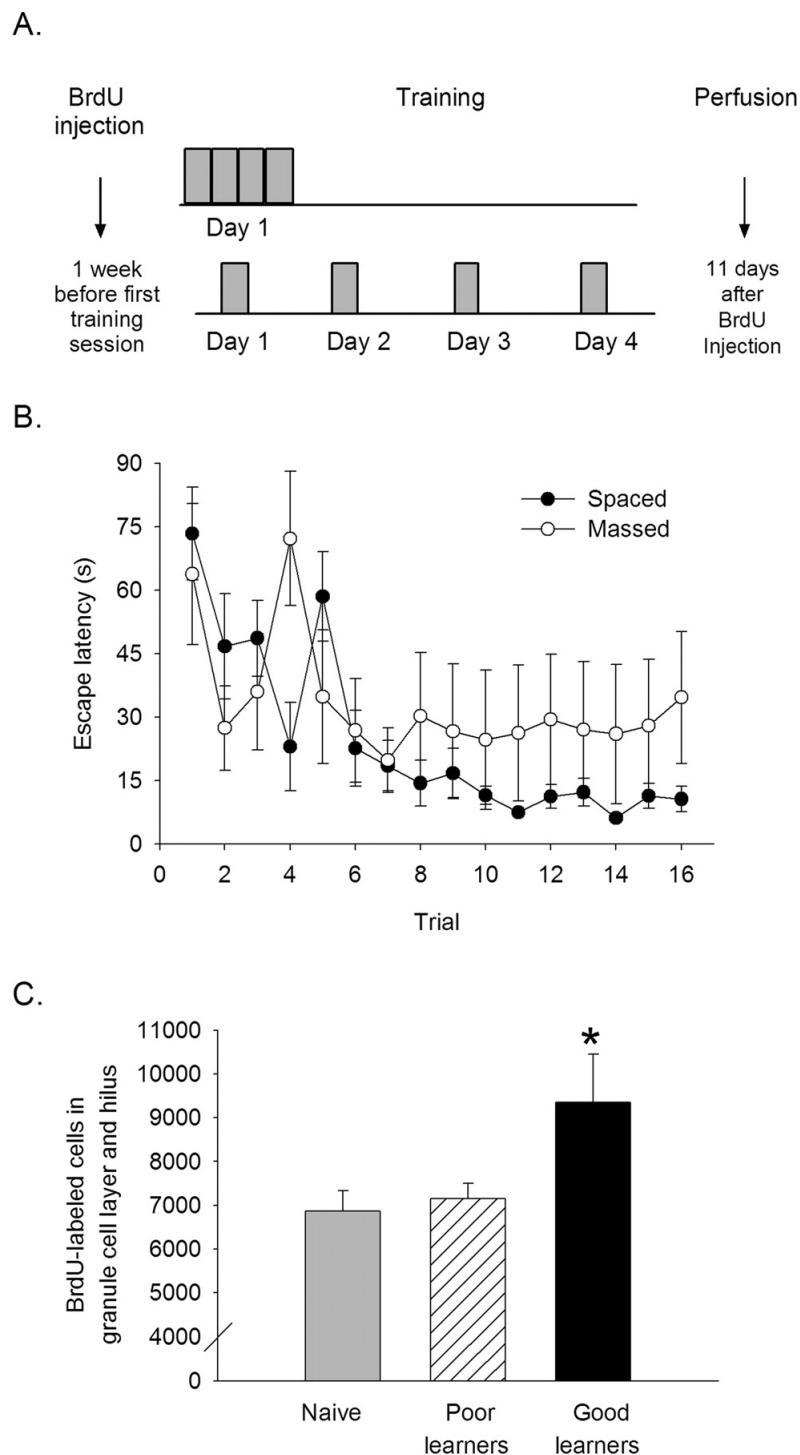
Fonte:Disponível em: <https://learning-evolution.com>. Acesso em: 04 set. 2019.

Pode-se notar também que a velocidade com que um estudante esquece as informações depende de múltiplos fatores como dificuldade do tema a ser estudado, como a informação está sendo apresentada e o quão fácil é para o estudante relacionar o conteúdo com algum conhecimento prévio.

O autor ressalta que apesar da velocidade de esquecimento variar, a curva exponencial mantém-se em todas os casos. Baseando-se nessa informação, é também mencionado o efeito causado pelo espaçamento do conteúdo para otimização do aprendizado de algo novo. O estudo revelou que reforçar a informação após um determinado intervalo de tempo é bem mais efetivo do que tentar memorizar uma grande quantidade de informação em um curto período de tempo.

Em uma pesquisa efetuada em 2007 pela equipe de neurocientistas da *Learning & Memories*, foi avaliado sob perspectiva biológica e fisiológica a aplicação do princípio de aprendizado espaçado, e percebeu-se que há um maior desenvolvimento dos neurônios na região do hipocampo, que é a região responsável por armazenar memórias a longo prazo.

Figura 3.2 – Aprendizado espaçado aumenta sobrevivência de células no hipocampo



Fonte: Learning & Memories - Neurogenesis and the spacing effect: Learning over time enhances memory and the survival of new neurons, 2007.

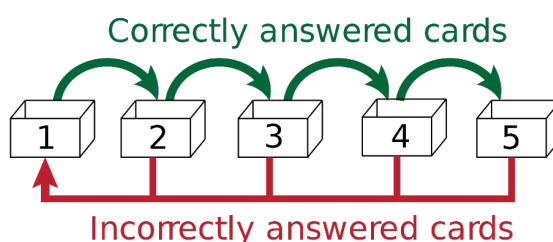
3.3 SRS - SPACED REPETITION SYSTEM

Como mencionado anteriormente, o método dos *flashcards* são apenas uma ferramenta para a repetição de conteúdo, o potencial desse método vem de técnicas e

algoritmos que utilizam o conceito de aprendizado espaçado. Idealmente o estudante deve estudar de forma otimizada e não repetindo todas as palavras o tempo inteiro, mas apenas repetir as palavras que estiver prestes a esquecer.

Um dos primeiros métodos que utilizou esse princípio para tornar *flashcards* um processo eficiente foi criado por um cientista alemão chamado Sebastian Leitner no ano de 1970 e é conhecido como o Leitner System ou método das caixas. A ideia do método de Leitner era separar esses cartões em diferentes caixas, onde cada caixa é reavaliada de acordo com a familiaridade com os cartões ali contidos. Para cada *flashcard*, se o estudante não lembrar com sucesso da resposta, o cartão é colocado em uma primeira caixa, caso contrário o *flashcard* irá para a caixa seguinte, indicando que o conteúdo foi memorizado. Para cada caixa há um tempo maior no qual o aluno deve revisitar, fazendo assim com que o aluno reavalie com mais frequência os cartões que ele possui mais dificuldade e com menos frequência aqueles que ele já possui domínio.

Figura 3.3 – Método das caixas



Fonte: Leitner System, 1970.

Durante os anos vários algoritmos foram desenvolvidos para otimizar o uso de *flashcards* nos estudos. Alguns dos métodos mais famosos são: Pimsleur, DASH e múltiplas versões do algoritmo SM. Para esse projeto o algoritmo escolhido para ser utilizado será o SM-2 que é o algoritmo de agendamento de *flashcards* mais utilizado na indústria atualmente.

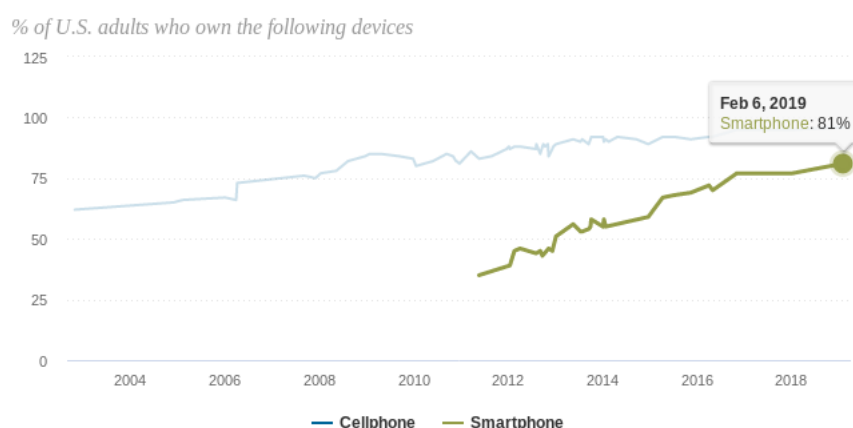
3.4 CALL - COMPUTER ASSISTED LANGUAGE LEARNING

CALL é a sigla para *Computer Assisted Language Learning* ou “Aprendizado de Línguas Auxiliado por Computadores” (tradução livre) é um termo que vem sendo utilizado desde os anos 60. Michael Levy descreve em seu livro chamado *Computer-Assisted Language Learning Context and Conceptualization* a origem e a natureza desse conceito e como softwares podem auxiliar professores e alunos ao aprender

uma língua estrangeira. O autor argumenta que softwares não podem substituir todo o ensino da linguagem, o professor ainda tem um papel importante no ensino, contudo, softwares podem ser um grande fator auxiliar nessa tarefa já que permitem novas abordagens para o professor.

A partir desse conceito, George M. Chinnery cria o conceito de MALL (*Mobile Assisted Language Learning*) que em tradução livre pode ser lido como "Aprendizado de línguas auxiliado por dispositivos móveis". Considerando o número crescente de pessoas que utilizam smartphones e dispositivos móveis diariamente, faz sentido utilizar essas tecnologias a nosso favor e tornar o acesso ao aprendizado de línguas estrangeiras mais acessível. De acordo com Pesquisas feitas pelo *Pew Research Center* (2011) apenas 35% da população dos Estados Unidos possuíam um smartphone enquanto em 2019 esse número subiu para 81%, ou seja, a maior parte da população norte-americana possui um dispositivo móvel que pode ser utilizado para o aprendizado de uma língua estrangeira.

Figura 3.4 – Quantidade da população que possui *Smartphones*



Fonte: Disponível em: <https://www.pewinternet.org/fact-sheet/mobile/>. Acesso em: 30 set. 2019.

Em seu artigo *Emerging Technologies - Going to the MALL: Mobile Assisted Language Learning*, Chinnery (2006) descreve que dispositivos móveis podem não ser a melhor plataforma para o aprendizado de novos conceitos, mas funcionam muito bem como uma ferramenta de revisão, o que colabora com o conceito inicial de CALL onde o software deve ser uma ferramenta de apoio para o estudante e não para substituir métodos tradicionais de ensino.

3.5 APRENDIZADO INTENCIONAL DE VOCABULÁRIO E APRENDIZADO INVOLUNTÁRIO

Ao aprender um novo vocabulário, um possível método de estudo é traduzir palavra por palavra através de sinônimos em sua língua nativa, esse método é chamado de aprendizado intencional de vocabulário ou aprendizado por substituição de palavras. Outra abordagem seria tentar inferir o significado da palavra dentro de um contexto, por exemplo ao ler o texto o leitor tenta através de outras palavras da frase interpretar ou pressupor o significado da palavra sem necessariamente achar uma palavra de referência em sua língua nativa, o nome dessa abordagem é aprendizado involuntário de vocabulário.

No artigo *Intentional Vs. Incidental Vocabulary Learning* escrito por Jameel Ahmad (2012) o autor discursa sobre a efetividade de cada método ao se aprender novo vocabulário. As estatísticas fornecidas pelo estudo mostram que o método involuntário funciona significativamente melhor, pois requer do estudante um processo mental mais profundo e além de auxiliar na memorização de palavras também auxilia o aluno a reconhecer elementos semânticos da linguagem e associar como adjetivos e substantivos interagem na língua a ser estudada. Naturalmente não é sempre possível inferir o significado de uma palavra a partir do contexto, nesses casos deve-se utilizar o método de sinônimos.

Após a leitura desses estudos é possível concluir que utilizar *flashcards* através de um sistema de repetição espaçada é bastante efetivo para aprender novo vocabulário, mas sua efetividade aumenta se utilizado dentro de um contexto textual onde é possível inferir o significado de algumas palavras, e quando não possível, as palavras com seus sinônimos na língua nativa podem ser adicionados ao sistema para melhor memorização.

4 PROPOSTA DA PLATAFORMA

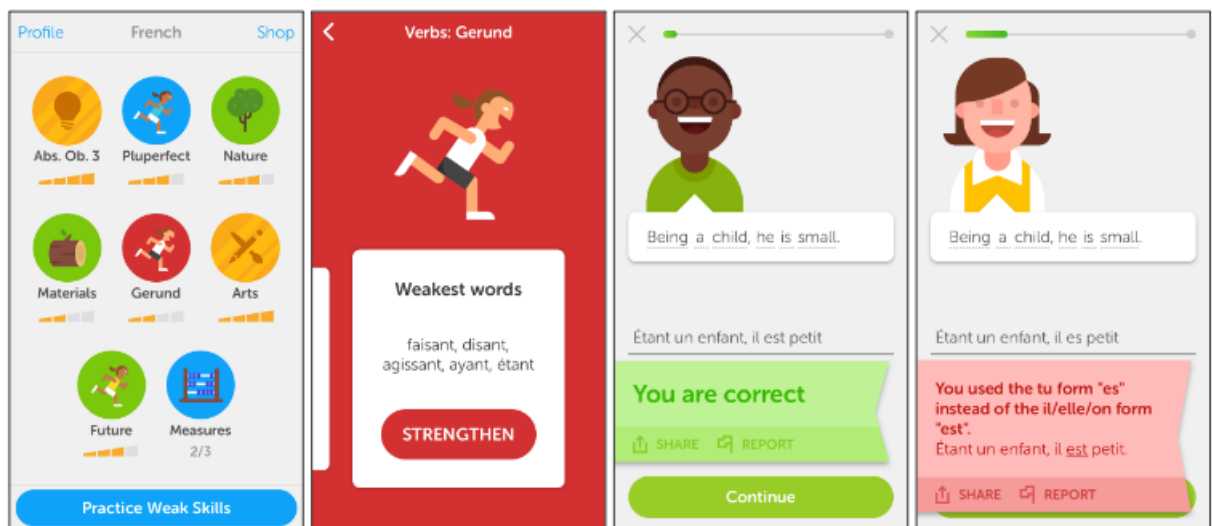
Esse projeto propõe o desenvolvimento de uma plataforma com o objetivo de auxiliar o aprendizado de novo vocabulário. Existem múltiplas aplicações com o mesmo fim no mercado atualmente, cada uma delas com suas particularidades, a seguir tem-se uma breve descrição de alguns exemplos:

4.1 APLICAÇÕES SEMELHANTES

4.1.1 Duolingo

Duolingo é uma plataforma que contém um aplicativo *mobile* e um sistema web para o aprendizado de línguas estrangeiras utilizando repetição espaçada. A plataforma utiliza-se de conceitos de gamificação para manter o usuário engajado na plataforma. O usuário progride através de de uma árvore de desafios e a dificuldade das sentenças a serem aprendidas aumentam gradualmente. Não é possível criar *flashcards* customizados e dessa forma o usuário fica limitado ao vocabulário que é escolhido pelo Duolingo para o aprendizado da linguagem.

Figura 4.1 – Duolingo

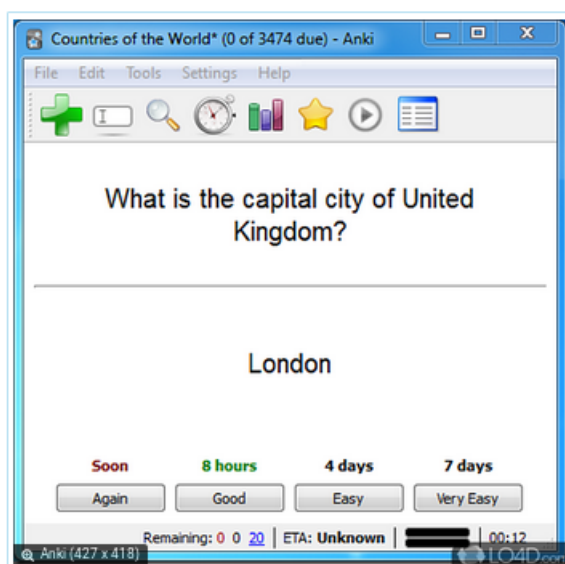


Fonte: A Trainable Spaced Repetition Model for Language Learning, 2016

O algoritmo utilizado para repetição espaçada no aplicativo é chamado de HLR ou *Half-Life Regression* que utiliza conceitos do método de Leitner juntamente com

se deve ao fato que possui embasamento científico (ANKI, 2019) e também é código aberto, que permite a distribuição do Anki como *software* livre e de código aberto para todos seus usuários.

Figura 4.3 – Anki



Fonte: Disponível em: <https://anki.en.lo4d.com/screenshots>. Acesso em: 01 out. 2019.

4.2 FLASHCARD COM LEITURA INTERATIVA

A proposta desse projeto é criar uma plataforma que permita ao usuário aprimorar seu vocabulário através do método de repetição espaçada e *flashcards* em conjunto com leitura de textos como artigos e notícias. Como foi descrito anteriormente no capítulo 3, o denominado aprendizado involuntário de novo vocabulário através um contexto textual é mais efetivo. A plataforma desenvolvida é composta de uma aplicação mobile para Android e iOS e um *backend* para sincronizar dados entre as plataformas (mais detalhes sobre implementação no Capítulo 5). A escolha da plataforma *mobile* deve-se ao fato de que a aplicação não requer nenhum *input* complexo e a interface de *touchscreen* é bastante intuitiva para o layout do aplicativo. Outro motivo que favoreceu a escolha da plataforma mobile é que o usuário pode praticar vocabulário mesmo não tendo acesso a um computador, sendo assim possível utilizar o aplicativo fora de casa ou no transporte público. No aplicativo o usuário pode escolher a partir de uma lista de notícias uma que seja de seu interesse e na língua que está sendo estudada e a partir desse artigo ele pode selecionar as palavras que tiver dificuldade e diretamente ver a tradução dessas palavras, se desejado o usuário pode adicionar a palavra selecionada a uma lista de *flashcards* para estudar posteriormente. O usuário pode também criar listas de *flashcards* customizados caso ele queira estudar palavras

que não necessariamente pertencem a uma notícia fornecida pelo aplicativo. Como descrito anteriormente, o algoritmo de repetição espaçada escolhido para esse projeto foi o SM-2. Um dos motivos para a escolha desse algoritmo é o fato de ser bastante performático e pode rodar direto no dispositivo do usuário, outro motivo foi o fato de que o algoritmo possui código aberto e possui licença livre. Quando o usuário decidir revisar uma lista de *flashcards*, o software calcula quais *flashcards* devem ser mostrados e com qual frequência baseado na performance do usuário sem sessões anteriores. O aplicativo também mostrará notificações para o usuário lembrar-se de revisar os *flashcards* quando for o melhor momento baseado na curva de esquecimento descrita no Capítulo 3.

5 ESPECIFICAÇÕES TÉCNICAS E ARQUITETURA

5.1 FLUTTER

O aplicativo foi desenvolvido utilizando o SDK de desenvolvimento Flutter. Flutter é um conjunto de ferramentas de código aberto desenvolvido pelo Google. Sua primeira versão foi lançada em 2018. O objetivo do Flutter é permitir a criação de aplicativos Android e iOS onde o resultado seja equivalente a um aplicativo desenvolvido nativamente mas com uma base de código em comum para as duas plataformas. O SDK utiliza internamente a biblioteca gráfica Skia para renderizar o aplicativo.

5.2 DART

Dart é a linguagem de programação utilizada em Flutter. Dart é uma linguagem desenvolvida e otimizada para se desenvolver interfaces de usuário interativas. A linguagem compila para ARM e código de máquina x64, que são as arquiteturas mais utilizadas atualmente. Também é possível compilar Dart para Javascript e utilizá-lo na Web. É possível compilar Dart de duas formas: *Ahead of Time* (AOT) e *Just in Time* (JIT). Utiliza-se AOT ao gerar o código que irá rodar em produção pois ele contém mais verificações e bloqueia a compilação em tempo de execução. Já para código em desenvolvimento utiliza-se JIT, o que permite um ciclo de desenvolvimento mais ágil já que nem toda a aplicação precisa ser reconstruída de uma vez só. Devido a essa flexibilidade, é possível utilizar a técnica chamada de *Hot reload* ou carregamento quente (tradução livre) onde o desenvolvedor pode modificar apenas um arquivo e quase instantaneamente verificar o resultado na aplicação.

5.3 ARQUITETURA

Comumente em desenvolvimento *mobile*, como por exemplo no Android SDK ou iOS UIKit, o desenvolvimento acontece de forma imperativa, ou seja, aquilo que o usuário está vendo em determinado momento é definido a partir de uma sequência de instruções dadas pelo desenvolvedor. Já em Flutter a aplicação é desenvolvida de forma declarativa, portanto aquilo que o usuário vê não é construído a partir de uma sequência de instruções, em vez disso representa um estado que pode ser representado por uma variável, uma estrutura de dados ou uma classe.

Se há uma mudança no estado da aplicação o *framework* atualiza a interface de usuário e redesenha todos os componentes para representar o estado atual. Não é possível acessar um único componente e modificá-lo diretamente, em vez disso

é necessário atualizar o estado da aplicação. Devido a essa natureza declarativa, gerenciamento de estados é um fator importante no desenvolvimento em Flutter.

Figura 5.1 – Programação declarativa

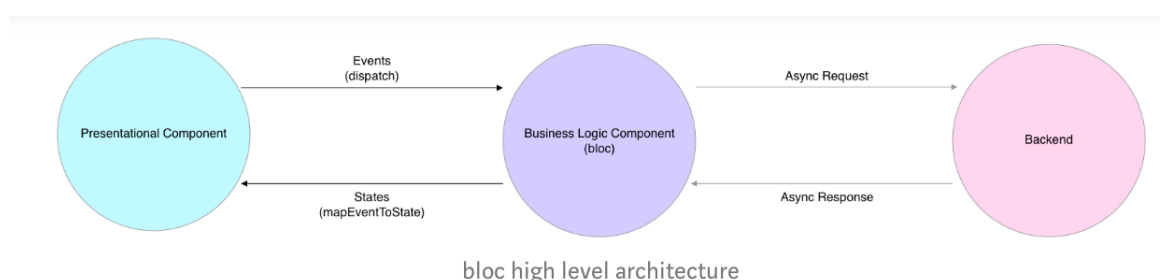


Fonte: Disponível em: <https://flutter.dev/docs/development/data-and-backend/state-mgmt/declarative>
Acesso em: 01 out. 2019.

Há várias bibliotecas e arquiteturas para facilitar esse processo, algumas delas específicas para Flutter, por exemplo: Provider, BLoC e Redux. O método utilizado nesse projeto é uma mistura de BLoC com Redux. BLoC ou *Business Logic Component* (Componente de lógica de negócios) baseia-se em separar o código de interface de usuário da lógica de negócios.

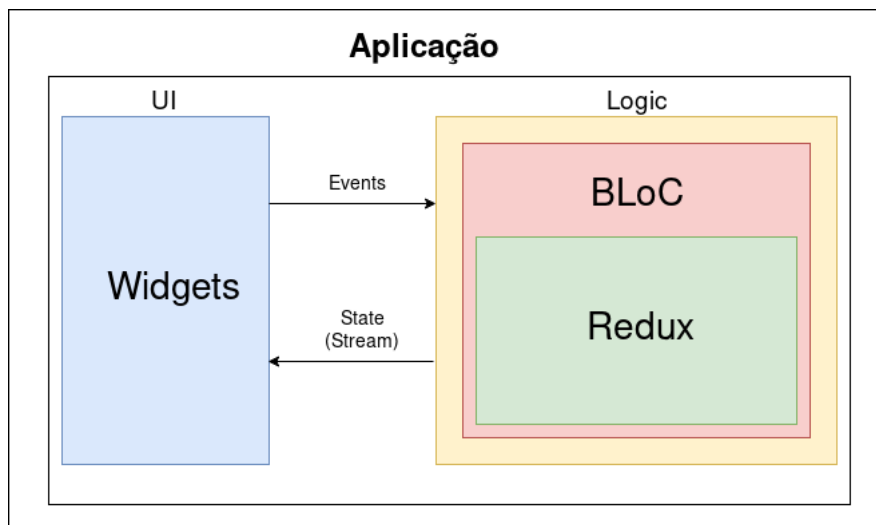
Redux é um *framework* desenvolvido por Dan Abramov primeiramente para desenvolvimento web e eventualmente adaptado para outras plataformas. Geralmente Redux é apresentado como um container de estados previsíveis, na prática isso significa uma maneira de gerenciar a interface de usuário da aplicação a partir de um estado global. Neste projeto o estado é construído na camada de lógica de negócios e os *Widgets* do flutter reagem a qualquer mudança que ocorra no objeto de estado. Uma visão geral da arquitetura do projeto pode ser vista nas figuras a seguir:

Figura 5.2 – BLoC Pattern



Fonte: Disponível em: <https://pub.dev/packages/bloc> Acesso em: 01 out. 2019.

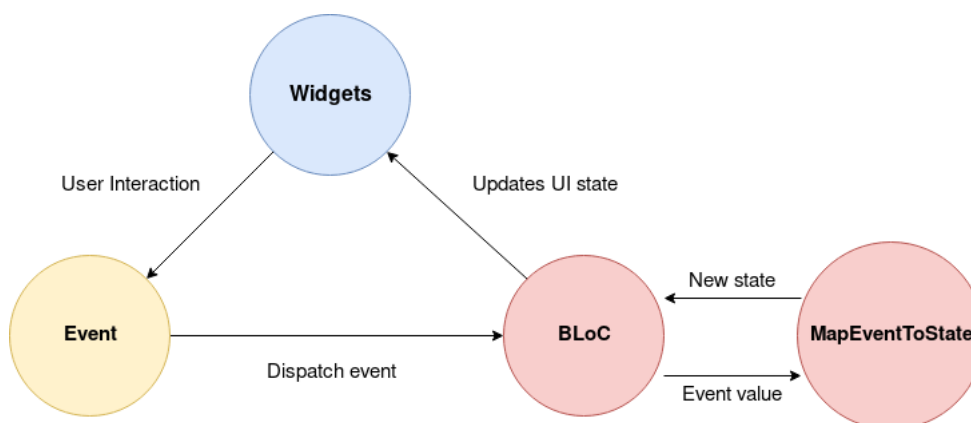
Figura 5.3 – Arquitetura Geral



Fonte: Elaborado pelo autor

Sempre que o usuário interagir com o aplicativo, essa interação emite eventos a partir da camada de UI através dos *Widgets* para a camada de lógica onde o *Redux* mapeia esse evento para um possível estado da aplicação. A camada de interface escuta por mudanças no estado e renderiza novamente quando há uma variação no estado.

Figura 5.4 – Redux



Fonte: Elaborado pelo autor

5.4 BACKEND

Foi desenvolvido um *backend* para a aplicação com o objetivo de gerenciar os dados e as sessões de usuário. Utilizou-se ASP.NET Core para desenvolver uma

RESTful API. .NET Core é a reimplementação de código aberto da plataforma .NET Framework criada pela Microsoft. ASP.NET Core é uma boa opção para o desenvolvimento de aplicações *backend* pois é eficiente, altamente escalável e rápido. De acordo com testes efetuados pela empresa *TechEmpower* ASP.NET Core está entre as aplicações para *WebAPI* mais eficientes. Aplicações desenvolvidas com ASP.NET Core também podem utilizar o serviço de hospedamento Azure da Microsoft gratuitamente.

PostgreSQL foi o banco de dados utilizado no desenvolvimento da aplicação. PostgreSQL é um banco de dados relacional de código aberto que utiliza linguagem SQL para operar em dados, funciona na maioria dos sistemas operacionais atualmente e segue todas as convenções de banco de dados, sendo as mesmas: atomicidade, consistência, isolamento e durabilidade.

6 CONCLUSÃO