

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO**

**APLICAÇÃO PARA MEMORIZAÇÃO UTILIZANDO**  
**MÉTODO SRS**

**VITOR AMARILDO KRÜGER REITER**

**BLUMENAU**  
**2020**

**VITOR AMARILDO KRÜGER REITER**

# **APLICAÇÃO DE MEMORIZAÇÃO UTILIZANDO MÉTODO**

## **SRS**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Profa. Joyce Martins, Mestre - Orientadora

**BLUMENAU  
2020**

# APLICAÇÃO PARA MEMORIZAÇÃO UTILIZANDO MÉTODO SRS

Por

**VITOR AMARILDO KRÜGER REITER**

Trabalho de Conclusão de Curso aprovado  
para obtenção dos créditos na disciplina de  
Trabalho de Conclusão de Curso II pela banca  
examinadora formada por:

Presidente:   
Prof(a). Joyce Martins – Orientador(a), FURB

Membro: \_\_\_\_\_  
Prof(a). Alexander Roberto Valdameri – FURB

Membro: \_\_\_\_\_  
Prof(a). Andreza Sartori – FURB

Blumenau, 17 de dezembro de 2020

Dedico este trabalho à minha família, a todos que contribuíram de alguma forma para a realização do mesmo e à minha orientadora que me auxiliou sem medir esforços nas diversas etapas deste trabalho.

## **AGRADECIMENTOS**

À minha família, em especial aos meus pais, por sempre me apoiarem e auxiliarem nos momentos em que eu mais precisei.

À minha namorada Maria Alice, por todo o carinho, incentivo e compreensão nos momentos mais difíceis.

Aos meus amigos, que sempre me deram motivos para continuar toda a caminhada universitária.

À minha orientadora Joyce Martins, por sua dedicação, paciência, cobranças, conselhos e o auxílio excepcional prestado no desenvolvimento deste trabalho.

Aqui no entanto nós não olhamos para trás por muito tempo. Nós continuamos seguindo em frente, abrindo novas portas e fazendo coisas novas. Porque somos curiosos e a curiosidade continua nos conduzindo por novos caminhos. Siga em frente.

Walt Disney

## RESUMO

Este trabalho apresenta o processo de especificação, desenvolvimento e operacionalidade de uma aplicação web para memorização de vocábulos e outros assuntos, que utiliza o método Spaced Repetition System (SRS), denominada Flashcard App. A metodologia escolhida permite que assuntos estudados sejam revisados antes mesmo que o cérebro perca essas informações. Para o uso da aplicação o usuário deve registrar-se e, após o cadastro prévio de categorias de assuntos e cartões de conteúdo, pode realizar lições a fim de memorizar de maneira contínua os conteúdos estudados. A aprendizagem e a memorização se dão por meio de cartões de conteúdo (*flashcards*) que possuem em uma face uma pergunta e na outra sua respectiva resposta com o áudio correspondente. Após estudar um conteúdo, o algoritmo agenda intervalos ideais de tempo para ocorrer a revisão do mesmo. Os objetivos deste trabalho foram alcançados uma vez que foi possível revisar vocábulos dentro de intervalos de tempo previstos pelo método SRS.

Palavras-chaves: Memorização. SRS. *Flashcards*.

## **ABSTRACT**

This paper presents the specification, development and operation process of a web application for word memorization and other subjects, which uses the Spaced Repetition System (SRS) method, called Flashcard App. The chosen methodology allows subjects studied to be reviewed before the brain even loses this information. To use the application the user must register and after a previous registration of subject categories and flashcards, can take lessons to memorize continuously the contents studied. Learning and memorization can be done by the use of flashcards that have questions on one side and their respective answers on the other, with the corresponding audios. After studying a content, the algorithm schedules the ideal time intervals for the content to be reviewed. The objectives of this work were accomplished, since it was possible to review words with time intervals provided by the SRS method.

**Keywords:** Memorization. SRS. Flashcards.

## LISTA DE FIGURAS

Figura 1 – Curva de esquecimento de Ebbinghaus.....	16
Figura 2 – Exemplificação do sistema Leitner .....	16
Figura 3 – Satoru: cadastro de cursos e lições.....	19
Figura 4 – Satoru: 1ª rodada de uma lição.....	20
Figura 5 – Duolingo: lições e exercícios específicos de uma lição .....	21
Figura 6 – Anki: cartão com questão e avaliação da resposta .....	22
Figura 7 – MosaLingua: categorias de vocábulos .....	24
Figura 8 – MosaLingua: frente e verso do cartão ( <i>flashcard</i> ) .....	24
Figura 9 – MosaLingua: revisão .....	25
Figura 10 – Diagrama de casos de uso .....	27
Figura 11 – Diagrama de atividades .....	28
Figura 12 – Flashcard App: tela inicial para cadastro de usuário.....	34
Figura 13 – Flashcard App: informações sobre o método.....	34
Figura 14 – Flashcard App: login .....	35
Figura 15 – Flashcard App: erros de acesso .....	35
Figura 16 – Flashcard App: <i>dashboard</i> .....	36
Figura 17 – Flashcard App: identificação do usuário logado .....	36
Figura 18 – Flashcard App: categorias .....	37
Figura 19 – Flashcard App: formulário para nova categoria.....	37
Figura 20 – Flashcard App: mensagem de exclusão de categoria.....	38
Figura 21 – Flashcard App: <i>flashcards</i> .....	38
Figura 22 – Flashcard App: formulário para novo <i>flashcard</i> .....	39
Figura 23 – Flashcard App: mensagem de exclusão de <i>flashcard</i> .....	39
Figura 24 – Flashcard App: lições .....	39
Figura 25 – Flashcard App: mensagem de lição sem <i>flashcards</i> na categoria .....	40
Figura 26 – Flashcard App: área de lições .....	40
Figura 27 – Flashcard App: <i>flashcards</i> em lições .....	41
Figura 28 – Flashcard App: mensagem ao término de lições.....	41
Figura 29 – Flashcard App: revisões .....	42
Figura 30 – Flashcard App: mensagens para as revisões .....	43
Figura 31 – Teste Flashcard App: Geografia 01 .....	44

Figura 32 – Teste Flashcard App: Geografia 02 .....	45
Figura 33 – Teste Flashcard App: Italiano 01 .....	45
Figura 34 – Teste Flashcard App: Italiano 02 .....	46

## LISTA DE QUADROS

Quadro 1 – Cálculo do intervalo de tempo para revisões.....	17
Quadro 2 – Inserção de novo cartão para revisão.....	31
Quadro 3 – Seleção de cartões para revisão .....	31
Quadro 4 – Agendamento da próxima revisão de cartões de conteúdo.....	32
Quadro 5 – Atualização do fator de facilidade (EF).....	32
Quadro 6 – Função que retorna o intervalo de dias .....	33
Quadro 7 – Comparativo entre os trabalhos correlatos e o projeto realizado.....	43
Quadro 8 – UC01: Cadastrar usuário .....	51
Quadro 9 – UC02: Efetuar login/logout .....	51
Quadro 10 – UC03: Manter categorias .....	52
Quadro 11 – UC04: Manter flashcards .....	52
Quadro 12 – UC05: Fazer lições .....	53
Quadro 13 – UC06: Realizar revisões .....	53
Quadro 14 – Tabela tb_users .....	54
Quadro 15 – Tabela tb_categories .....	54
Quadro 16 – Tabela tb_flashcards .....	54
Quadro 17 – Tabela tb_lessons .....	55

## **LISTA DE ABREVIATURAS E SIGLAS**

API – Application Programming Interface

EF – Easiness Factor (ou E-Factor)

EJS – Embedded JavaScript

HTML – Hypertext Markup Language

RF – Requisito Funcional

RNF – Requisito Não Funcional

SM-2 – SuperMemo 2.0

SRS – Spaced Repetition System

TOEFL – Test of English as a Foreign Language

UC – Use Case

UML – Unified Modeling Language

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>13</b>
1.1 OBJETIVOS.....	13
1.2 ESTRUTURA.....	14
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>15</b>
2.1 SISTEMA DE REPETIÇÃO ESPAÇADA (SRS).....	15
2.2 CARTÕES DE CONTEÚDO (FLASHCARDS).....	16
2.3 ALGORITMO PARA CÁLCULO DA REALIZAÇÃO DE REVISÕES.....	17
2.4 TRABALHOS CORRELATOS .....	18
2.4.1 SATORU.....	18
2.4.2 DUOLINGO.....	20
2.4.3 ANKI.....	22
2.4.4 MOSALINGUA .....	23
<b>3 DESENVOLVIMENTO.....</b>	<b>26</b>
3.1 LEVANTAMENTO DE REQUISITOS .....	26
3.2 ESPECIFICAÇÃO .....	27
3.2.1 Casos de uso.....	27
3.2.2 Diagrama de atividades .....	28
3.2.3 Organização e manipulação de banco de dados .....	29
3.3 IMPLEMENTAÇÃO .....	29
3.3.1 Técnicas e ferramentas utilizadas.....	29
3.3.2 Organização lógica e funcionamento do agendamento de revisões do método implementado.....	30
3.3.3 Operacionalidade da implementação .....	33
3.4 RESULTADOS E DISCUSSÕES.....	43
<b>4 CONCLUSÕES.....</b>	<b>47</b>
4.1 EXTENSÕES .....	47
<b>REFERÊNCIAS .....</b>	<b>49</b>
<b>APÊNDICE A – DESCRIÇÃO DOS CASOS DE USO .....</b>	<b>51</b>
<b>APÊNDICE B – DICIONÁRIO DE DADOS.....</b>	<b>54</b>

## 1 INTRODUÇÃO

O domínio de uma língua estrangeira no mundo globalizado tornou-se um requisito, seja no âmbito acadêmico ou profissional, pois, conforme afirma Lomb (2008, p. 12, tradução nossa), poliglota em dezesseis idiomas diferentes, “[...] devemos aprender idiomas porque um idioma é a única coisa que vale o conhecimento, mesmo que superficialmente”. Quando se estuda uma língua estrangeira, determinadas habilidades tornam-se primordiais para o indivíduo, como ler, escrever, falar e compreender. Contudo, de acordo com Garcia (2014), um dos fatores que deve ser levado em consideração para avaliação do nível de fluência de alguém em uma língua é o tamanho de seu vocabulário. Dessa maneira, quanto mais palavras souber, mais fluente a pessoa será.

O processo de aprendizagem, seja do vocabulário de um idioma ou de conceitos de uma área de conhecimento, poderia ser menos árduo se as pessoas pudessem aprender qualquer coisa uma única vez (GARCIA, 2014). O problema é que o cérebro humano não funciona assim, as informações assimiladas no cotidiano possuem um prazo de validade. Logo, quando algo relativamente novo é aprendido, o cérebro guarda a informação em conexões neurais, mas, como descreve Garcia (2014, p. 1), “[...] essas conexões primeiramente costumam ser muito fracas e em pouco tempo tendem a ser desfeitas. Resultado: nós esquecemos.”. Para que isso não aconteça, torna-se necessário revisar (ou usar) a informação aprendida, reforçando as conexões neurais. O uso de revisões no processo de aprendizagem é a base do método Spaced Repetition System (SRS).

O método SRS, também conhecido como Sistema de Repetições Espaçadas, é uma técnica de memorização de conteúdo baseada na curva de esquecimento humano em que, de acordo com Lameri (2017), os atos da revisão são espaçados em intervalos que aumentam com o sucesso e diminuem com o insucesso. A metodologia consiste em a pessoa revisar conteúdos num tempo ideal, antes mesmo que o cérebro perca a informação. A memorização do conteúdo é dada por meio de *flashcards*, isto é, cartões de conteúdo em que se anota a pergunta na frente do cartão e a resposta no verso de tal forma que o indivíduo tente recordar a resposta antes de visualizá-la (SANTOS; BARROS, 2018). Diante do exposto, este trabalho descreve o desenvolvimento de uma aplicação web para memorização de vocabulário de um idioma, utilizando o SRS.

### 1.1 OBJETIVOS

O objetivo principal deste trabalho é desenvolver uma aplicação web para memorização de vocabulário por meio do método SRS.

Os objetivos específicos do trabalho são:

- a) permitir a criação de cartões de conteúdo com o vocabulário estrangeiro a ser estudado;
- b) possibilitar o uso de texto e áudio para criar os cartões de conteúdo;
- c) validar a aplicação para memorização de vocabulário.

## 1.2 ESTRUTURA

Este trabalho foi estruturado em quatro capítulos. O primeiro capítulo aborda a introdução do trabalho desenvolvido, os objetivos e como está organizado. No segundo capítulo, são apresentados os conceitos e fundamentos teóricos para o desenvolvimento da aplicação proposta, que se utiliza do método SRS, juntamente com os trabalhos correlatos. O terceiro capítulo traz os requisitos necessários para a implementação da aplicação, técnicas e ferramentas utilizadas, a operacionalidade da implementação, bem como descreve os resultados obtidos. Por fim, são apresentadas as conclusões, assim como sugeridas extensões para serem implementadas no futuro.

## 2 FUNDAMENTAÇÃO TEÓRICA

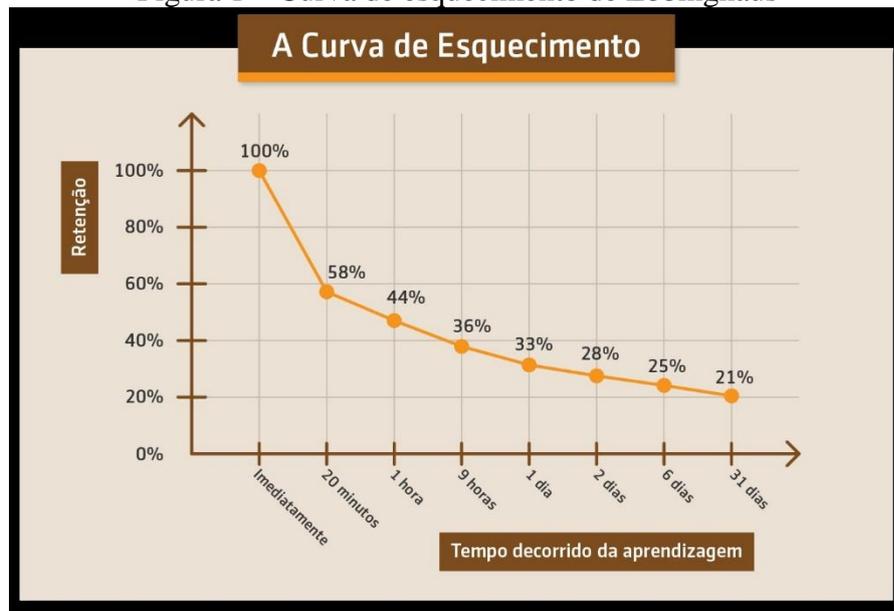
Neste capítulo são descritos os assuntos que fundamentaram o desenvolvimento da aplicação descrita, incluindo o método SRS e o uso cartões na memorização de conteúdo, bem como o cálculo da realização de revisões do estudo. Também são apresentados trabalhos correlatos a esse projeto.

### 2.1 SISTEMA DE REPETIÇÃO ESPAÇADA (SRS)

Ao final da década de 1800, o psicólogo e cientista alemão Hermann Ebbinghaus realizou um experimento analítico sobre a memória, usando uma lista de sílabas aleatórias a fim de mensurar quanto tempo levaria para esquecer o que havia sido estudado e depois reaprendê-lo novamente (SANTOS; OLIVEIRA JUNIOR, 2018). O estudo de Ebbinghaus demonstrou a capacidade da memória humana de retenção da informação com o passar do tempo, pois aproximadamente 2,5 sílabas por segundo foram repetidas diversas vezes, mantendo assim, um ritmo mental atlético. Santos e Oliveira Junior (2018) relatam que esse e outros experimentos foram repetidos pelo cientista por mais três anos com o propósito de mostrar que os resultados obtidos não eram sorte ou acidente. Após a obtenção dos mesmos resultados do primeiro estudo, em 1885, Ebbinghaus publicou um estudo com a representação dos resultados em uma curva, que veio a ser conhecida como a Curva de Esquecimento de Ebbinghaus, visualizada na Figura 1. Nesse sentido, as evidências encontradas por Ebbinghaus sugerem que para existir a fixação de uma informação na memória de longo prazo, é preciso estimular a memória em determinados intervalos de tempo, pois após 20 minutos que determinado conteúdo foi aprendido, o percentual de retenção cai para 58% (MICHELOT, 2018a).

Esta descoberta originou o método SRS que se baseia em três etapas: a codificação, a consolidação e a recuperação. Na codificação é feita a organização e processamento de uma informação de primeiro contato. Em seguida, a consolidação corresponde ao momento em que esta informação é colocada na memória. Por fim, a etapa de recuperação está relacionada à possibilidade de acessar essa informação, o que só é possível se a informação for repetida quantas vezes for necessária e no momento correto, garantindo sua integração à memória de longo prazo (MICHELOT, 2018a).

Figura 1 – Curva de esquecimento de Ebbinghaus



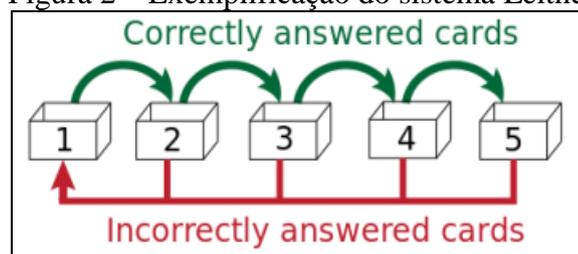
Fonte: Cassimiro (2018).

## 2.2 CARTÕES DE CONTEÚDO (FLASHCARDS)

No século XIX, conforme Santos e Oliveira Junior (2018) descrevem, professores já usavam como método de apoio à aprendizagem o uso de cartões de papel, que consistia em colocar em um dos lados do cartão o conteúdo a ser estudado, através de uma pergunta, e no verso do cartão a resposta correspondente. Com o passar de anos, o método foi aperfeiçoado por diversos estudiosos, como o caso do jornalista alemão Sebastian Leitner que na década de 1970 criou um processo de agendamento de cartões, com perguntas na frente e respostas no verso.

O Sistema Leitner agrupa os cartões em diferentes caixas ou blocos de cartões, em função dos acertos ou erros do estudante. Assim, o conteúdo a ser memorizado passa a ser separado em caixas com uma periodicidade diferente, onde ao revisar determinado cartão, o mesmo é movido para a próxima caixa com um intervalo maior em caso de sucesso (acerto do estudante) ou retrocede a uma caixa de intervalo menor em caso de insucesso (erro do estudante) (LAMERI, 2017). A Figura 2 exemplifica o Sistema Leitner.

Figura 2 – Exemplificação do sistema Leitner



Fonte: Lameri (2017, p. 23).

### 2.3 ALGORITMO PARA CÁLCULO DA REALIZAÇÃO DE REVISÕES

Em 1987, o pesquisador polonês Piotr Wozniak utilizou a abordagem do Sistema Leitner para desenvolver um algoritmo denominado SuperMemo, que posteriormente teve sua lógica difundida em diversos outros programas de computador, propondo auxiliar no processo de memorização de conteúdos considerados muito difíceis (SANTOS; OLIVEIRA JUNIOR, 2018). O algoritmo de Wozniak apresenta, a partir da curva de esquecimento, a informação a ser revisada em um tempo ideal, levando em consideração as dificuldades da pessoa em reter a informação (MICHELOT, 2018b).

A primeira implementação do SuperMemo foi desenvolvida em Turbo Pascal 3.0 em dezembro de 1987 e foi nomeada de SuperMemo 2.0 ou SM-2 (WOZNIAK, 2018). O intervalo de tempo para realizar as revisões é calculado a partir de um intervalo anterior e sua respectiva classificação de dificuldade, informada pelo usuário, denominada de qualidade na resposta, organizada na seguinte escala: (5) resposta imediata e correta; (4) resposta correta com hesitação; (3) resposta correta com dificuldade séria; (2) resposta incorreta, onde a resposta correta é de fácil lembrança; (1) resposta incorreta, a resposta correta é relembrada; (0) esquecimento total. O cálculo do intervalo de tempo para revisões segue os passos descritos no Quadro 1.

Quadro 1 – Cálculo do intervalo de tempo para revisões

<p><b>1° passo:</b> para cada cartão novo associar um fator de facilidade (<i>E-Factor</i> ou EF) igual a 2,5. Podendo este valor diminuir ou aumentar de acordo no decorrer das repetições;</p> <p><b>2° passo:</b> repetir cada cartão em intervalos (I) expressos da seguinte forma:  <math>I(1) := 1</math> dia  <math>I(2) := 6</math> dias  para cada <math>n &gt; 2</math> faça  <math>I(n) := I(n-1) * EF</math>, onde <math>I(n)</math> representa o intervalo em dias da <math>n</math>ésima repetição e EF o fator de facilidade de cada cartão</p> <p><b>3° passo:</b> solicitar ao usuário a qualidade (q) da resposta da repetição na escala de 0 a 5, sendo 0 esquecimento total e 5 perfeita recordação;</p> <p><b>4° passo:</b> caso a qualidade (q) da resposta seja menor que 3, ou seja, o estudante não lembra do conteúdo, retornar as repetições do cartão ao seu estado inicial para que o cartão seja memorizado novamente; caso contrário, alterar o fator de facilidade (EF) de cada repetição usando a seguinte fórmula:  <math>EF := EF + (0,1 - (5 - q) * (0,08 + (5 - q) * 0,02))</math>  caso EF seja menor do que 1,3, manter o valor em 1,3 para que as repetições sejam realizadas de maneira constante;</p> <p><b>5° passo:</b> após cada sessão de repetição de um determinado dia, repetir todos os itens com uma pontuação abaixo de 4 na avaliação da qualidade. Continuar as repetições até que todos os cartões obtenham pontuação 4.</p>
---

Fonte: adaptado de Wozniak (2018).

## 2.4 TRABALHOS CORRELATOS

A utilização do SRS como método de memorização encontra-se presente em trabalhos acadêmicos e em aplicações comerciais. Os trabalhos a seguir foram selecionados para fins de comparação e de embasamento desse estudo. Na próxima seção é descrita uma plataforma web desenvolvida para aprendizagem do idioma japonês criada por Lameri (2017). As seções seguintes abrangem alguns aplicativos/softwarewares comerciais para o ensino de línguas, incluindo eventualmente a descrição das tecnologias usadas.

### 2.4.1 SATORU

O Satoru, desenvolvido em meio acadêmico por Lameri (2017), é uma plataforma web para aprendizado do japonês, com foco em proporcionar o entendimento das estruturas de dois dos três diferentes alfabetos existentes: o Katakana e o Hiragana. Além disso, como as frases deste idioma são construídas com um padrão inexistente na língua portuguesa, utilizou-se a metodologia SRS a fim de descomplicar a vida do estudante na assimilação do japonês. As tecnologias utilizadas no desenvolvimento não foram descritas.

Na aplicação de Lameri (2017), o estudante inicia seus estudos pelos alfabetos Katakana e Hiragana, denominados de cursos, proporcionando um conhecimento inicial básico do idioma. Para tanto, tem-se listas de palavras escritas (lições) no alfabeto a ser aprendido, sendo cada palavra associada à imagem correspondente e ao áudio da sua pronúncia. A cada lição podem ser associados um ou mais conteúdos, ou seja, expressões em japonês relacionadas ao assunto que está sendo aprendido. Por exemplo, para a lição *Obrigado* e *tchau* do curso Hiragana, *arigatou* e *sayonara*, entre outros, são dois conteúdos relacionados.

Após o usuário efetuar o login na aplicação, ele é direcionado a um *dashboard*. Existem dois tipos de usuário na aplicação: o administrador e o estudante. O usuário administrador possui acesso ao cadastro de cursos e lições (Figura 3, menu à esquerda), podendo também fazer associações de conteúdos às lições para que quando um conteúdo seja revisado, os conteúdos relacionados possam ser adicionados à revisão.

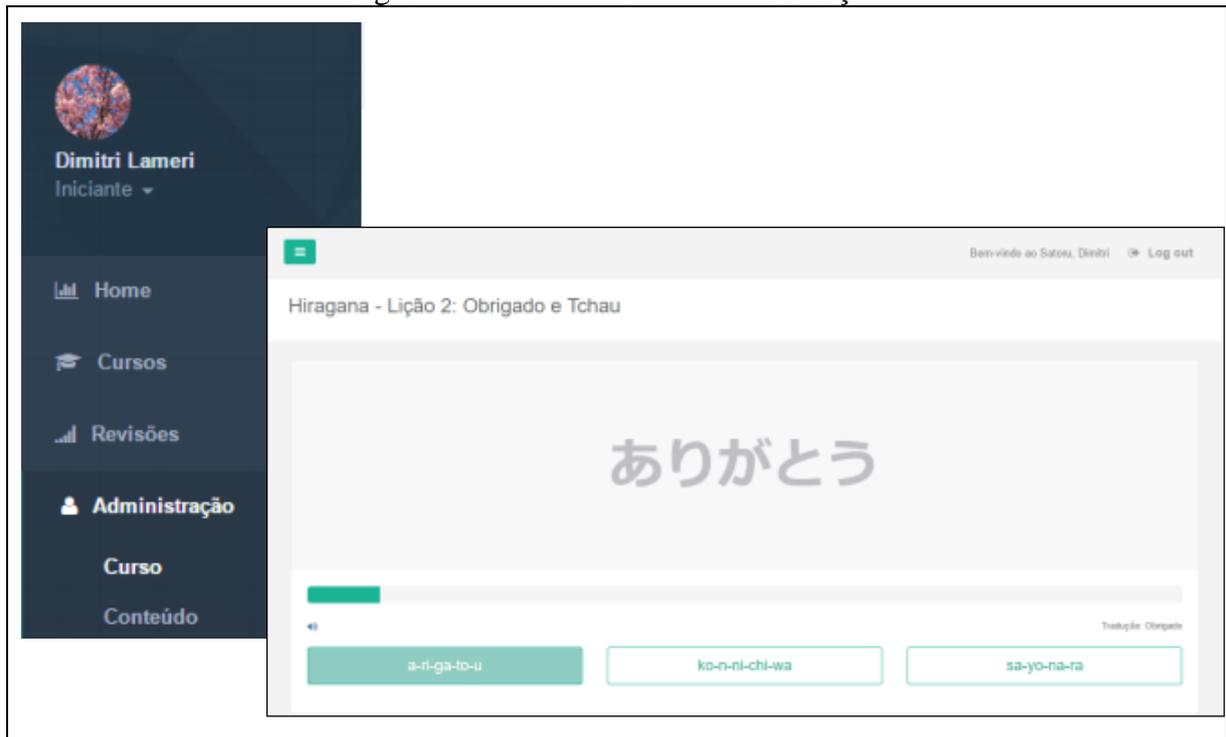
Figura 3 – Satoru: cadastro de cursos e lições

The image shows two overlapping screenshots of the Satoru application interface. The top screenshot displays the 'Curso' (Course) management page, which includes a 'Listagem' (List) table with columns for 'Nome' (Name), 'Descrição' (Description), and 'Ações' (Actions). A single entry for 'Hiragana' is visible. A 'Criar Curso' (Create Course) button is located in the top right corner. The bottom screenshot shows the 'Lições de Hiragana' (Hiragana Lessons) page, also featuring a 'Listagem' table with the same columns. Two lesson entries are listed: 'Quem sou eu?' (Who am I?) with the description 'Pronomes pessoais' (Personal pronouns), and 'Obrigado e Tchau' (Thank you and Bye) with 'Palavras simples' (Simple words). A 'Criar Lição' (Create Lesson) button is in the top right. On the left side of the bottom screenshot, a dark navigation menu is visible with options for 'Administração' (Administration), 'Curso' (Course), and 'Conteúdo' (Content).

Fonte: adaptado de Lameri (2017, p. 38).

O estudante tem acesso aos cursos e respectivas lições já cadastrados, assim como o seu progresso com informações sobre os estudos e as revisões feitas, graficamente mostradas no *dashboard*. O estudante deve selecionar o curso desejado acessando-o no menu à esquerda (Figura 4). São apresentadas as diversas lições, as quais devem ser efetuadas em ordem para que as próximas sejam liberadas. Qualquer lição após sua conclusão continua disponível para ser refeita a qualquer momento. Uma lição consiste em uma lista de palavras ou expressões a serem estudadas. Cada palavra ou expressão da lição é apresentada até que o estudante acerte quatro vezes. Na primeira rodada, conforme visualizado na Figura 4, é exibida a palavra ou a expressão em japonês com o áudio da sua pronúncia, assim como a opção correta selecionada e a tradução para a língua portuguesa, servindo para apresentar o vocábulo. Nas rodadas seguintes, o áudio não é iniciado automaticamente, ficando apenas disponível para consulta, e a opção correta não se encontra mais destacada. Na última rodada, o estudante apenas escuta o áudio e visualiza palavras ou expressões escritas em japonês sem qualquer auxílio (tradução para a língua portuguesa ou imagem correspondente), sendo necessário selecionar a opção correta. No final de cada lição, o conteúdo aprendido é adicionado a uma lista de revisão.

Figura 4 – Satoru: 1ª rodada de uma lição



Fonte: adaptado de Lameri (2017, p. 41).

A revisão no Satoru utiliza-se do método SRS para indicar quais conteúdos necessitam ser revisados diariamente, funcionando como a última rodada de uma lição, conforme descrito anteriormente. As diferenças são: (a) tem-se mais de um conteúdo agendado para o mesmo dia; (b) não existe mais dica de qual é a opção correta.

Por fim, Lameri (2017) sugere como melhorias para a plataforma: (a) o uso dos desenhos dos ideogramas japoneses; (b) a captura do áudio de falantes nativos; (c) o uso de uma Application Programming Interface (API) adequada para a reprodução da pronúncia das palavras e expressões, já que a aplicação utiliza-se de uma API instável.

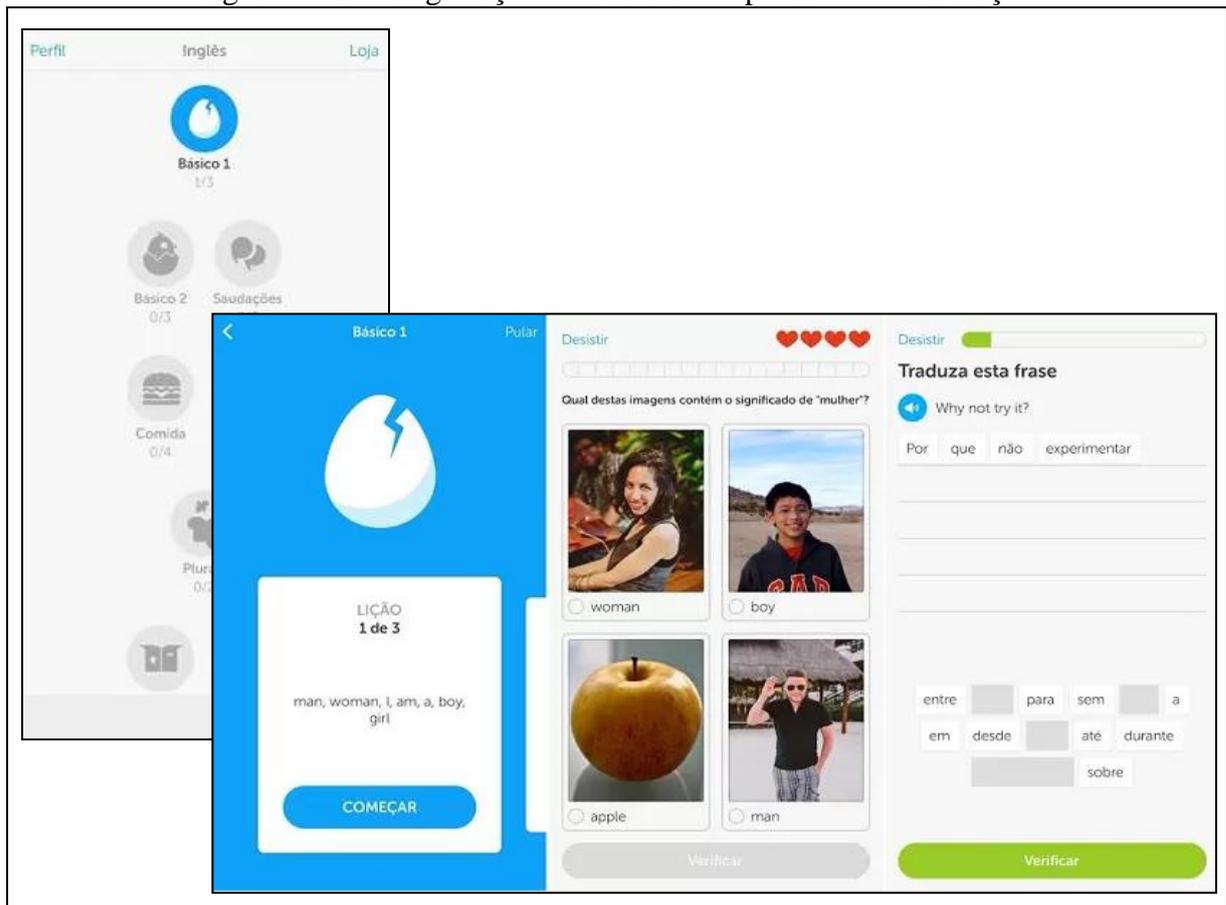
#### 2.4.2 DUOLINGO

Um dos aplicativos mais utilizados para o ensino de idiomas é o Duolingo, uma plataforma online que possui versões para diversos dispositivos. Por se tratar de uma plataforma online, o servidor *back-end* foi escrito na linguagem de programação Python e o *front-end* foi escrito em Backbone.js, Mustache, JQuery e Bootstrap (MEEDER, 2013).

O aplicativo foi concebido como um jogo onde se pode perder vidas (corações) a cada resposta errada ou conquistar os chamados “lingots”, dinheiro do Duolingo que pode ser trocado por recompensas na plataforma, conforme for passando de fase (lições). O conteúdo no Duolingo é organizado de forma hierárquica, compondo uma árvore de lições/fases (Figura 5, à esquerda) que são desbloqueadas à medida que o usuário avança (BARROS, 2016). Os

curso são criados também por contribuição de falantes nativos. No entanto, a disponibilidade de idiomas estrangeiros para aprendizagem é diferente dependendo da versão do aplicativo. A metodologia de ensino correlaciona as palavras a serem assimiladas com imagens, sons ou mesmo traduções de frases (Figura 5, à direita). Assim, o usuário escolhe/escreve as respostas nos diferentes exercícios que uma lição pode ter, o Duolingo verifica a resposta e, à medida que o usuário acerta, liberam-se novos exercícios até a conclusão da lição (BARROS, 2016).

Figura 5 – Duolingo: lições e exercícios específicos de uma lição



Fonte: Barros (2016).

O processo de revisão, por sua vez, não é diário, como no método SRS, tendo uma periodicidade calculada pela própria plataforma, havendo a possibilidade de o usuário revisar determinado conteúdo quantas vezes achar necessário. O algoritmo do Duolingo utiliza-se de *deep learning* no reforço das lições. Dessa forma, através de *deep learning* ou aprendizado profundo, é possível utilizar-se de redes neurais que imitam o comportamento cerebral para processamento de linguagem natural, analisando o histórico das respostas dos usuários para prever a probabilidade de obterem uma resposta correta. Essas previsões são a base para personalizar o teste de aprendizagem adaptável e o conteúdo de aprendizagem da aplicação (PERANANDAM, 2018).

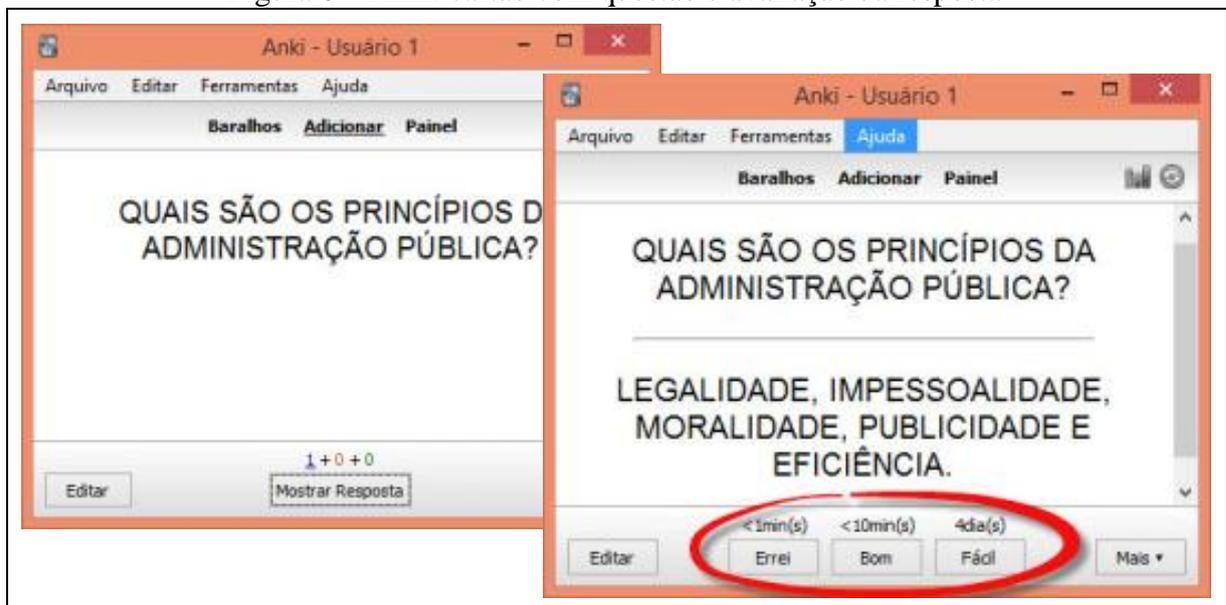
### 2.4.3 ANKI

Anki é uma aplicação *open source*, escrita em Python. Foi projetada inicialmente para o aprendizado de idiomas, mas foi adaptada para qualquer assunto e está disponível em diversas plataformas, tais como Android, iOS, Mac, Linux ou mesmo Windows (ANKI 2.1..., 2018).

Uma vez que a aplicação esteja instalada, o usuário pode criar um baralho de cartões de conteúdo, com os assuntos que deseja memorizar, sendo que cada cartão contém uma pergunta e a respectiva resposta. Os cartões podem ser associados a imagens, vídeos, áudios, texto Hypertext Markup Language (HTML) ou marcação científica usando LaTeX. O usuário pode também baixar um dos baralhos disponíveis no site do Anki, os quais estão listados por diferentes categorias, para então importá-lo e começar os estudos. Por padrão, Anki escolhe vinte cartões do baralho selecionado para serem estudados por dia. Contudo, este e outros recursos podem ser customizados, para baralhos diversos.

Durante os estudos, como Manfrin (2015) exemplifica usando um baralho de Direito Constitucional, quando um cartão é aberto na ferramenta, o usuário vê a pergunta, tenta recordar-se da resposta e em seguida clica na única opção disponível, *Mostrar Resposta* (Figura 6, à esquerda).

Figura 6 – Anki: cartão com questão e avaliação da resposta



Fonte: Manfrin (2015).

O usuário então compara a resposta apresentada e a resposta por ele dada e define entre *Errei*, *Bom* e *Fácil* (Figura 6, à direita), o quão recorrente será a revisão daquele cartão estudado. Em uma sessão de estudos, existem três tipos de cartões: (a) cartões novos que não foram estudados; (b) cartões vistos uma vez; (c) cartões que precisam ser revisados, sendo o

tempo para revisão calculado em função da classificação da resposta dada pelo usuário (Errei, Bom, Fácil) (MANFRIN, 2015).

O algoritmo por traz do agendamento das revisões no Anki foi baseado no SuperMemo ou SM-2, descrito anteriormente. O intervalo de tempo das revisões é calculado da seguinte forma: (1) para a opção *Errei*, o cartão será revisto em 1 minuto; (2) para *Bom*, será revisto nos próximos 10 minutos; (3) para *Fácil*, por padrão, o cartão será exibido novamente 4 dias depois e, após isso, com intervalos cada vez maiores. Todavia, a definição desses intervalos pode ser customizada. Caso não haja outros cartões a serem mostrados, será repetido o aprendizado dos cartões mesmo que o intervalo de tempo não tenha sido concluído completamente. Assim, a ferramenta entende que pode ser necessário revisar um cartão várias vezes até sua memorização (ANKI 2.1..., 2018).

#### 2.4.4 MOSALINGUA

O MosaLingua é uma plataforma de ensino de idiomas criada para smartphones, disponibilizando aplicativos tanto para Android quanto iOS, com mais de sete milhões de usuários no mundo. A plataforma possui aplicativos específicos para o ensino de idiomas, focados no ensino do vocabulário necessário para avaliações como o Test Of English as a Foreign Language (TOEFL) ou voltado à área de negócios (MATHILDE, 2018).

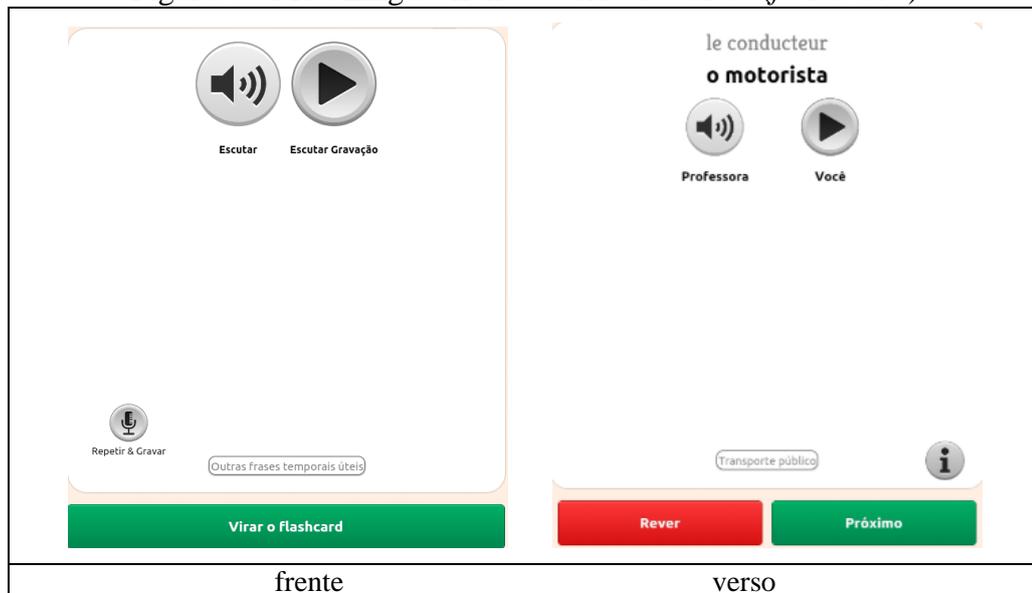
MosaLingua conta com mais de três mil cartões de conteúdo selecionados pela própria equipe de desenvolvimento, onde as palavras e frases são divididas em torno de quinze categorias (*Compras, Turismo, Lazer, Social, entre outras*) (Figura 7). Assim, ao instalar o aplicativo, o usuário deve escolher algum objetivo, como viagem ou negociações corporativas, por exemplo, justamente para que a aplicação apresente as listas de vocábulos de acordo com o objetivo escolhido, fazendo com que o usuário não perca tempo com palavras inúteis (MATHILDE, 2018).

O ensino neste aplicativo é feito através da opção *Aprender* (Figura 7, menu à esquerda) onde estão os cartões relacionados às categorias mais compatíveis com o objetivo previamente escolhido. Cada sessão de aprendizagem possui um número limite de cartões, podendo haver inclusão/exclusão de mais cartões se o usuário desejar. Os cartões dentro do aplicativo são associados não só as palavras, mas também ao áudio e à escrita do vocábulo. Ao virar um cartão, o usuário possui a opção de continuar aprendendo as demais palavras da lista ou eventualmente revisar aquele cartão (Figura 8).

Figura 7 – MosaLingua: categorias de vocábulos



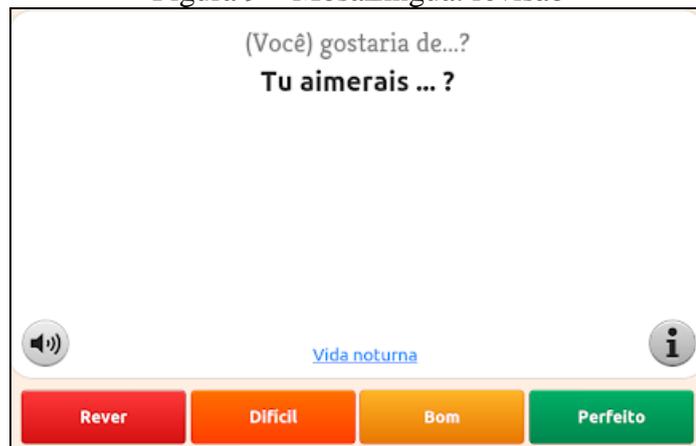
Fonte: Mathilde (2018).

Figura 8 – MosaLingua: frente e verso do cartão (*flashcard*)

Fonte: adaptado de Martins (2015).

O algoritmo de revisão do MosaLingua também provém dos estudos da criação do SuperMemo capaz de efetuar os cálculos de datas ideais para a revisão, seguindo a autoavaliação do usuário em seus cartões (MANFRIN, 2018). A revisão de cartões é feita cotidianamente, podendo este cartão ganhar um intervalo de revisão maior de dias ou não, em virtude da autoavaliação do usuário, conforme pode ser observado na Figura 9 (MARTINS, 2015).

Figura 9 – MosaLingua: revisão



Fonte: adaptado de Martins (2015).

### 3 DESENVOLVIMENTO

Este capítulo compreende o desenvolvimento da aplicação. A seção 3.1 lista os requisitos funcionais e não funcionais identificados no presente trabalho. Na seção 3.2 são apresentados os artefatos gerados durante o processo de diagramação e modelagem do software. A seção 3.3 aborda a implementação e, por fim, na seção 3.4 são mostrados os resultados alcançados com este projeto.

#### 3.1 LEVANTAMENTO DE REQUISITOS

A aplicação desenvolvida deve:

- a) permitir que o usuário efetue um cadastro, fornecendo nome, e-mail e senha (Requisito Funcional - RF);
- b) permitir que o usuário cadastre categorias de assuntos a serem estudados, informando o idioma para o áudio correspondente às respostas (RF);
- c) permitir que o usuário forneça vocábulos (pergunta / resposta) para cartões de conteúdo, indicando a categoria de assunto a que se relacionam (RF);
- d) criar os cartões de conteúdo a partir dos vocábulos informados, buscando automaticamente o áudio correspondente à resposta (RF);
- e) permitir que o usuário estude os cartões de conteúdo de um determinado assunto, apresentando-os aleatoriamente (RF);
- f) permitir que o usuário classifique os cartões estudados indicando se acertou ou errou a resposta (RF);
- g) permitir que o usuário realize revisões dos cartões de conteúdo, classificando-os de acordo com o grau de dificuldade ao recordar o conteúdo (rever, difícil, bom, perfeito) (RF);
- h) estar disponível em uma plataforma web (Requisito Não Funcional - RNF);
- i) utilizar a API `SpeechSynthesisUtterance` de síntese de voz para reprodução do áudio correspondente às respostas (RNF);
- j) persistir as informações em banco de dados MySQL (RNF);
- k) usar o método SRS para determinar o intervalo de tempo nas revisões do conteúdo estudado (RNF);
- l) acessar o conteúdo das lições e revisões apenas por meio de e-mail e senha previamente cadastrados (RNF);
- m) ter como opções Português, Inglês, Italiano, Alemão, Espanhol e Francês para cadastro de idiomas em categoria (RNF);

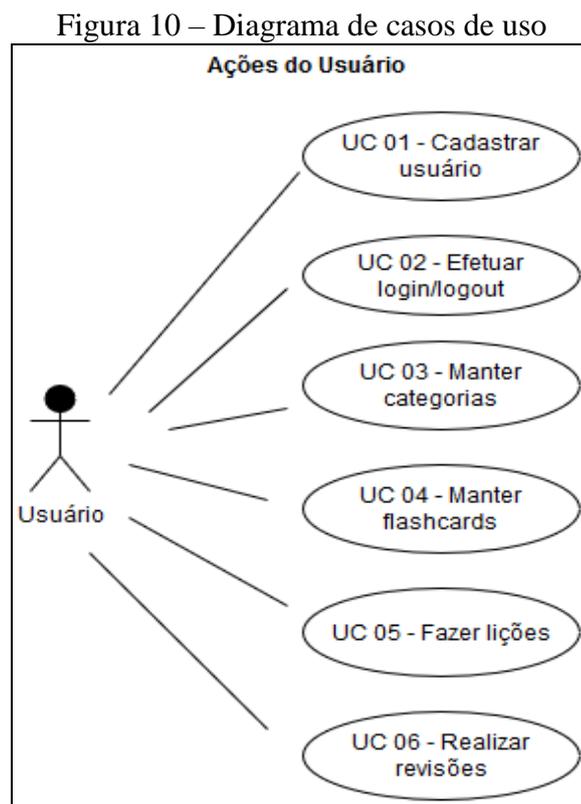
- n) ter suporte aos navegadores Chrome (versão 33 ou superior), Firefox (versão 49 ou superior) e Safari (versão 7 ou superior) (RNF);
- o) ter o *back-end* implementado em JavaScript usando NodeJS (RNF);
- p) ter o *front-end* implementado em Javascript usando EJS junto com o *framework* AdminLTE (RNF).

## 3.2 ESPECIFICAÇÃO

Esta seção aborda os principais artefatos, como diagramas de caso de uso e atividades da UML, que compõem a especificação da solução desenvolvida.

### 3.2.1 Casos de uso

Neste projeto foram identificados seis casos de uso, em Inglês *Use Case* – UC (Figura 10), detalhados no Apêndice A.



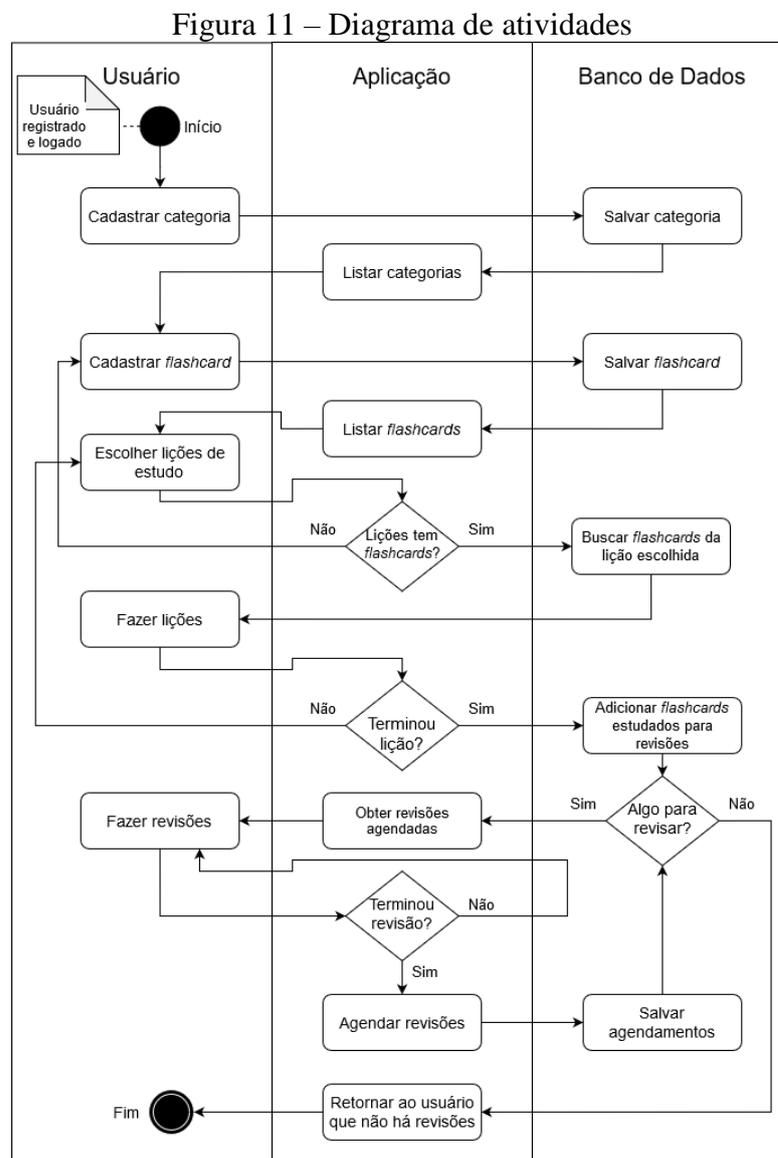
Fonte: elaborado pelo autor.

Neste diagrama, os casos de uso foram atrelados a um ator, denominado *Usuário*. Este, por sua vez, consegue efetuar seu cadastro para acessar aplicação através do caso de uso *Cadastrar usuário*. Os casos de uso *Efetuar login/logout*, *Manter categorias* e *Manter flashcards* permitem que o usuário realize login/logout após informar suas credenciais e mantenha tanto as categorias de assuntos, como os cartões a serem estudados.

Os dois últimos casos de uso, *Fazer lições* e *Realizar revisões*, estão relacionados com a experiência de estudo do usuário, que, após realizar as lições de um determinado conteúdo, poderá revê-las em determinados intervalos de tempo, de acordo com a classificação do grau de dificuldade que o usuário teve ao recordar o conteúdo dos cartões.

### 3.2.2 Diagrama de atividades

A Figura 11 mostra o processo de interação entre o usuário e a aplicação, bem como as etapas necessárias para estudar um novo conteúdo. Após o usuário estar registrado e fazer o login corretamente, inicia a interação com o cadastro de categorias, seguido do cadastro de *flashcards* (cartões de conteúdo a serem estudados). Na sequência, o usuário poderá realizar lições e revisões.



Fonte: elaborado pelo autor.

### 3.2.3 Organização e manipulação de banco de dados

O banco de dados deste projeto possui quatro tabelas, detalhadas no Apêndice B. Todas as tabelas utilizadas armazenam um campo `id` que possui uma chave única, tornando cada registro único no banco de dados, bem como um campo com data e hora da inserção para manter um controle do registro dos dados. Além disso, não é permitida a inserção com valores nulos (`null`).

Uma das tabelas é `tb_users`, que é populada com nome, e-mail e senha de novos usuários. Outra tabela denomina-se `tb_categories` e possui as categorias de assuntos estudados (campos `name` e `description`), juntamente com o idioma (campo `lang`) a ser utilizado para o áudio das respostas às perguntas de cada cartão daquela categoria. Os campos `name` e `lang` são posteriormente utilizados para tabela de *flashcards*. Já a tabela `tb_flashcards` armazena a categoria, a língua, a frente (pergunta) e o verso (resposta) de cada cartão cadastrado. As informações contidas nessa tabela são listadas e filtradas por sua categoria para a realização das lições, sendo que os cartões de conteúdo apresentados ao usuário devem ser por ele classificados como corretos ou incorretos. Os cartões classificados como incorretos permanecem registrados nesta tabela, ao passo que, os classificados como corretos são inseridos na tabela `tb_lessons`.

A tabela `tb_lessons` comporta todas as informações dos cartões respondidos de forma correta nas lições, juntamente com os campos necessários para persistência das informações para uma revisão contínua. Durante as revisões, o conteúdo desta tabela é filtrado de acordo com as datas das revisões (campo `nextRevision`). À medida que os cartões são classificados nesta etapa, os campos contendo o número de vezes que o cartão foi revisado, a data da próxima revisão e o fator de facilidade com que o cartão foi recordado (EF) são atualizados e salvos, até que o usuário revise-os novamente, repetindo o processo.

## 3.3 IMPLEMENTAÇÃO

Esta seção relaciona as principais ferramentas utilizadas durante o desenvolvimento do projeto. A organização lógica e o funcionamento do agendamento de revisões do método SRS também são mostrados, bem como, a operacionalidade da implementação.

### 3.3.1 Técnicas e ferramentas utilizadas

No desenvolvimento desta aplicação foram utilizadas as seguintes bibliotecas e ferramentas:

- a) Visual Studio Code: ambiente de desenvolvimento multiplataforma gratuito, criado e mantido pela Microsoft, utilizado para codificação da aplicação na linguagem JavaScript;
- b) NodeJS versão 12.14.1: plataforma de programação criada a partir da Runtime V8 do navegador Chrome, usada como servidor *back-end*;
- c) Redis versão 3.2.1: base de dados *in-memory* de armazenamento chave-valor, funcionando como um *middleware* de sessão da aplicação para permitir a comunicação com o banco de dados;
- d) AdminLTE versão 2.4.13: *framework* para *templates* web, feito com Bootstrap 3, empregado no layout das telas;
- e) EJS versão 2.6.1: *template engine* para gerar conteúdo HTML dinâmico com sintaxe JavaScript, usada nas páginas web;
- f) SpeechSynthesisUtterance versão 1.0.2: biblioteca experimental de áudio empregada para converter conteúdo de texto em fala, mas que não é suportada por todos os navegadores;
- g) MySQL versão 8.0.21: banco de dados utilizado para armazenar as informações dos usuários cadastrados, tais como nome, e-mail e senha. Também é usado para persistir cartões de conteúdo e lições, assim como o progresso do usuário vinculado a um assunto estudado.

### 3.3.2 Organização lógica e funcionamento do agendamento de revisões do método implementado

O usuário deve inicialmente cadastrar pelo menos uma categoria sobre um assunto a ser estudado. Após, cartões de conteúdo devem ser cadastrados para serem estudados, associando-os a uma categoria, o que define também uma lição. O usuário deve, então, escolher por meio da categoria, qual o assunto que deseja aprender e efetuar a lição correspondente. No momento em que a lição estiver concluída, os cartões de conteúdo que foram classificados como corretos prosseguem para a segunda etapa, para estarem em constante revisão, como prevê o método SRS. Em contra partida, os cartões classificados como incorretos, permanecem associados à lição até que sejam considerados corretos pelo usuário.

A fim de seguir com as etapas do método, todo cartão de conteúdo registrado na tabela de lições (`tb_lessons`) do banco de dados, recebe valor do fator de facilidade equivalente a 2.5 (linha 03, Quadro 2), número de repetições igual a 0 (linha 04, Quadro 2), bem como a

data da próxima revisão (linhas 05 a 07, Quadro 2). O Quadro 2 apresenta um trecho de código no *back-end* que contém a atribuição desses valores a cada novo cartão a ser revisado.

Quadro 2 – Inserção de novo cartão para revisão

```

01 saveLessons(fields){
02   return new Promise((resolve, reject) =>{
03     let ef = 2.5;
04     let repetitions = 0;
05     let currentDate = new Date();
06     let formatDate = `${currentDate.getFullYear()}-
07     ${currentDate.getMonth()+1}-${currentDate.getDay()+2}`;
08
09     conn.query(`INSERT INTO tb_lessons (id, category, front, back,
10     timesRepeated, nextRevision, ef) VALUES(?,?,?,?,?,?,?)`, [fields.id,
11     fields.category, fields.front, fields.back, repetitions, formatDate, ef],
12     (err, results) =>{ if(err){
13       console.log(err);
14       reject(err);
15     }else{
16       console.log(results);
17       resolve(results);
18     }
19     });
20   });
21 }

```

Fonte: elaborado pelo autor.

Na etapa de revisão, os cartões são apresentados diariamente de acordo com a data de revisão determinada anteriormente (Quadro 2). Cartões com data anterior ou igual a data atual, serão revisados naquela sessão (linhas 09 e 10, Quadro 3). No Quadro 3, é possível visualizar o código que realiza a seleção dos cartões para cada sessão de revisão.

Quadro 3 – Seleção de cartões para revisão

```

01 function selectCardsForReview(array){
02   let filteredCards = array.filter((card) =>{
03     let today = new Date();
04     //transforma string date do array em date
05     let stringDate = card.nextRevision.split('T');
06     let dtParts = stringDate[0].split('-');
07     let dateCard = new Date(dtParts[0], dtParts[1] - 1, dtParts[2]);
08
09     if(dateCard<= today)
10       return card;
11   });
12   return filteredCards;
13 }

```

Fonte: elaborado pelo autor.

Uma vez que os cartões de conteúdo foram selecionados, dá-se início à sessão de revisão, na qual os cartões são novamente apresentados, devendo o usuário classificar a qualidade de sua recordação. Esta classificação de qualidade é feita ao selecionar uma entre as opções rever, difícil, bom e perfeito, que atribuem, respectivamente, os valores 1, 2, 4 e 5, usados para determinar a data da próxima revisão (Quadro 4).

Quadro 4 – Agendamento da próxima revisão de cartões de conteúdo

```

01 function scheduleReview(quality, card){
02   let currentDate = new Date();
03
04   if(quality < 3){
05     ef = 2.5;
06     repetitions = 1;
07     card.ef = ef;
08     return addDays(currentDate, nextReview(repetitions, ef));
09   }else{
10     ef = card.ef;
11     ef = updateEffort(quality, ef);
12     card.ef = ef;
13     return addDays(currentDate, nextReview(card.timesRepeated, ef));
14   }
15 }

```

Fonte: elaborado pelo autor.

O Quadro 4 apresenta o cálculo para o agendamento da próxima revisão de cada cartão. Uma vez que a qualidade informada seja inferior a 3 (linha 04, Quadro 4), ou seja, o usuário não recordou ou recordou com dificuldade o conteúdo mostrado, o fator de facilidade é reiniciado com 2.5 e o número de repetições é setado com 1 (linhas 05 a 07, Quadro 4). Caso a qualidade informada seja igual ou superior a 3, ou seja, o usuário recordou ou recordou com facilidade o conteúdo mostrado, o valor do fator de facilidade (EF) é atualizado (linhas 10 a 12, Quadro 4). O fator de facilidade (EF) e o número de repetições do cartão são usados para determinar a data da próxima revisão (linhas 08 e 13, Quadro 4).

O fator de facilidade (EF) foi criado por Wozniak (2018) com o intuito de medir uma aproximação ou não da dificuldade de recordar um item estudado. Quanto maior for o valor de EF, mais fácil será a retenção e recordação do conteúdo. Para que haja atualização deste valor, utiliza-se uma fórmula construída heurísticamente por ele. Esta fórmula foi transcrita para uma função, apresentada no Quadro 5, que tem como parâmetros a qualidade informada pelo usuário e o fator de facilidade anteriormente definido (linha 11, Quadro 4). No decorrer das repetições, Wozniak (2018) observou que este valor poderia gradualmente reduzir em caso de problemas na recordação dos conteúdos. Assim, ele determinou que valor do fator de facilidade (EF) não poderia ficar abaixo de 1,3 (linhas 06 a 08, Quadro 5), sendo que cartões com valores inferiores a 1,3 seriam repetidos com frequência.

Quadro 5 – Atualização do fator de facilidade (EF)

```

01 function updateEffort(quality, ef){
02   let newEf = 0;
03
04   newEf = parseFloat(ef)+(0.1-(5-quality*(0.08+(5-quality*0.02)));
05
06   if(newEf < 1.3){
07     newEf = 1.3;
08   }
09
10   return newEf;
11 }

```

Fonte: elaborado pelo autor.

Para o retorno em dias, que define o intervalo espaçado para a revisão de cada cartão, leva-se em consideração o número de repetições e fator de facilidade (EF) que o cartão obteve. Cartões com número de repetições igual a 1, devem ser revisados em 1 dia (linhas 02 e 03, Quadro 6). Já cartões com número de repetições iguais a 2, têm um intervalo de 6 dias até a próxima revisão (linhas 04 e 05, Quadro 6), enquanto cartões com número de repetições acima de 2, têm o número de dias contabilizado através da chamada recursiva expressa na linha 07 do Quadro 6, em que o valor final é arredondado.

Quadro 6 – Função que retorna o intervalo de dias

```

01 function nextReview(cardRepetitions, ef){
02   if(cardRepetitions == 1){
03     return 1;
04   }else if(cardRepetitions == 2){
05     return 6;
06   }else{
07     let nextReviewDays = Math.round(nextReview(cardRepetitions-1) * ef);
08     return parseInt(nextReviewDays);
09   }
10 }

```

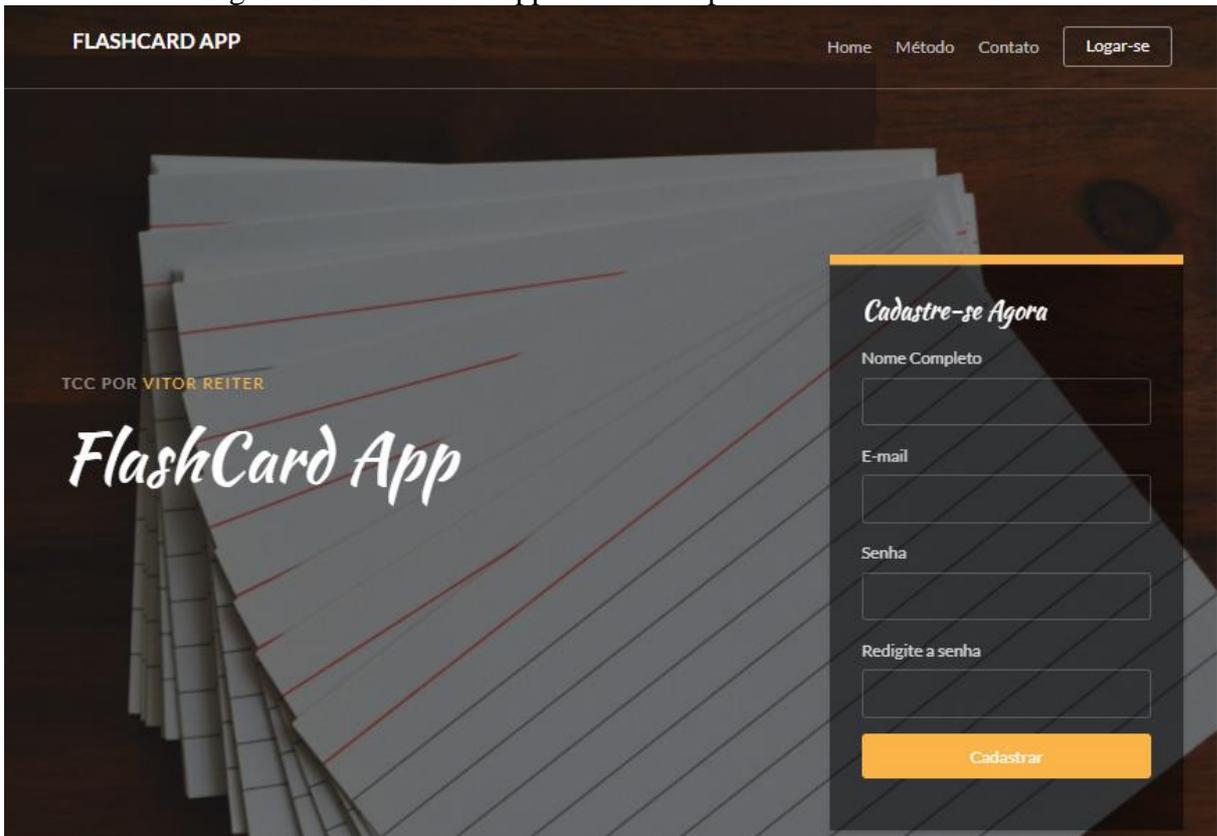
Fonte: elaborado pelo autor.

Ao obter o intervalo de dias de um cartão, este é acrescido à data atual da sessão de revisão (linhas 08 e 13, Quadro 4), que resulta na próxima data em que aquele conteúdo deve ser revisado.

### 3.3.3 Operacionalidade da implementação

O software descrito neste trabalho foi desenvolvido como uma aplicação web. Portanto, se o mesmo estiver hospedado em algum servidor, basta apenas acessá-lo pelo navegador, sem a necessidade de instalação para ser utilizado. Primeiramente, a partir da tela inicial apresentada na Figura 12, o usuário consegue realizar seu cadastro. Além disso, mesmo se não estiver logado, pode ainda navegar para outras páginas (indicadas no menu superior direito) que permitem ler sobre o método SRS (Figura 13, assim como visualizar informações de contato ou acessar a área de login (Figura 14).

Figura 12 – Flashcard App: tela inicial para cadastro de usuário



Fonte: elaborado pelo autor.

Figura 13 – Flashcard App: informações sobre o método

**Método Spaced Repetition System (SRS)**

...também conhecido como Sistema de Repetições Espaçadas, é uma técnica de memorização de conteúdo baseada na curva de esquecimento humano, onde o objetivo é prolongar o conhecimento adquirido!

 <b>Crie FlashCards</b> Monte seu próprio baralho de conteúdo a ser estudado	 <b>Aprendizagem contínua</b> Tenha revisões do que estudou, no momento certo	 <b>Memória duradoura</b> Faça com que o conhecimento fique em sua memória de longo prazo, para não esquecer o que aprendeu
 <b>Acertos</b> Prolongue o tempo do conhecimento que você assimilou	 <b>Erros</b> Revise constantemente o que ainda não memorizou	 <b>Curva do esquecimento</b> Todas as informações adquiridas possuem uma data de validade, este método procura prolongar esta data

Fonte: elaborado pelo autor.

Figura 14 – Flashcard App: login

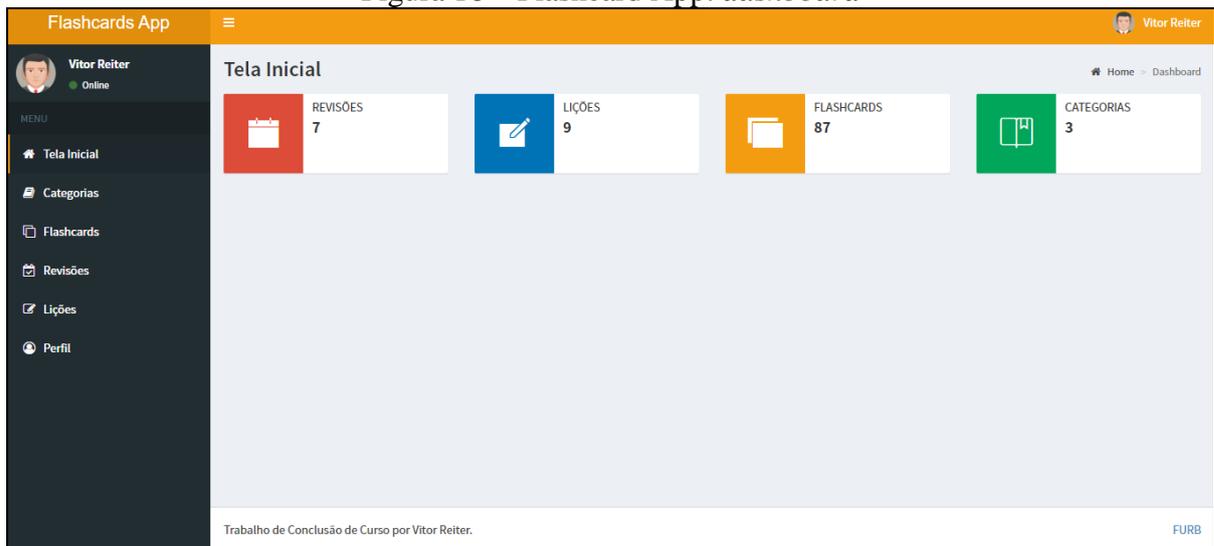
Fonte: elaborado pelo autor.

Se houver tentativa de acesso não contendo e-mail ou senha, é indicado o campo faltante ao usuário (Figura 15, à esquerda). Caso o usuário informe e-mail ou senha de forma incorreta, é mostrada uma mensagem de erro (Figura 15, à direita).

Figura 15 – Flashcard App: erros de acesso

Fonte: elaborado pelo autor.

Quando o usuário informa um e-mail e senha válidos, ele é redirecionado para o *dashboard* da aplicação (Figura 16). Aí encontram-se informações relevantes para o uso da ferramenta, tais como: a quantidade atual de cartões para serem revisados, a quantidade de cartões que já foram corretamente estudados nas lições, a quantidade de *flashcards*/cartões de conteúdo e de categorias registrados. Então, na Figura 16 é possível observar que 3 categorias (CATEGORIAS = 3) e 87 cartões de conteúdo (FLASHCARDS = 87) foram registrados, sendo que dos 9 cartões corretamente estudados nas lições (LIÇÕES = 9), 7 precisam ser revisados na data atual (REVISÕES = 7).

Figura 16 – Flashcard App: *dashboard*

Fonte: elaborado pelo autor.

O direcionamento para as outras páginas pode ser feito através das opções do menu lateral esquerdo da Figura 16, ou mesmo, ao clicar sobre qualquer um dos quatro *cards* coloridos (REVISÕES, LIÇÕES, FLASHCARDS, CATEGORIAS). Durante toda a sessão em que o usuário estiver logado, ele pode se desconectar da aplicação por meio da barra fixa de navegação com seu nome, no canto superior direito (Figura 16). Ao clicar sobre o nome, um *drop-down* revela duas opções: *Perfil* (botão à esquerda, Figura 17), permitindo configurar o perfil do usuário, e *Sair* (botão à direita, Figura 17) para encerrar a sessão.

Figura 17 – Flashcard App: identificação do usuário logado

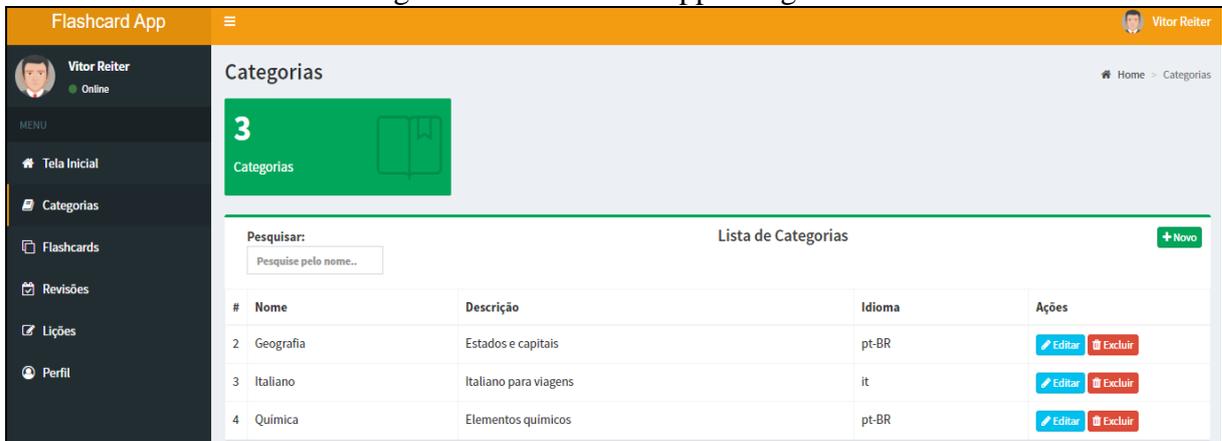


Fonte: elaborado pelo autor.

Para definir um conteúdo a ser estudado, inicialmente o usuário deve cadastrá-lo como uma categoria, fazendo o acesso à página de *Categorias*. É mostrada a quantidade de categorias cadastradas na aplicação, juntamente com as informações sobre cada categoria (Figura 18). Tem-se: Nome, Descrição, Idioma e Ações. A categoria *Geografia*, por exemplo, trata dos estados e suas capitais, sendo que o idioma das respostas é português (pt-

BR). A coluna **Ações** contém as opções: **Editar** e **Excluir**, permitindo que o usuário altere ou exclua qualquer categoria que desejar.

Figura 18 – Flashcard App: categorias



Fonte: elaborado pelo autor.

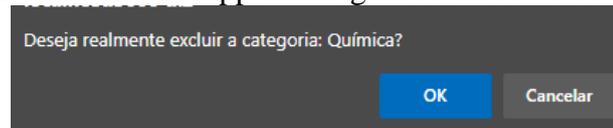
Além disso, é disponibilizada uma caixa de texto intitulada **Pesquisar** (Figura 18), que possibilita que uma pesquisa seja realizada pelo nome da categoria. Ao pressionar o botão **Novo**, é possível preencher um formulário (Figura 19) para inserir uma nova categoria no banco de dados. Ao pressionar o botão **Editar**, é possível alterar uma categoria existente. Tanto num quanto no outro caso, o usuário precisa informar **Nome** e **Descrição**, assim como selecionar um **Idioma** para o áudio das respostas, dentre os disponíveis, sendo eles: Alemão, Espanhol, Francês, Inglês, Italiano e Português.

Figura 19 – Flashcard App: formulário para nova categoria

Fonte: elaborado pelo autor.

Ao pressionar no botão **Excluir**, o usuário é questionado se a categoria selecionada deve mesmo ser excluída, ou seja, é necessário confirmar a ação, como mostrado na Figura 20.

Figura 20 – Flashcard App: mensagem de exclusão de categoria



Fonte: elaborado pelo autor.

Dado que o usuário cadastrou a categoria do conteúdo desejado corretamente, a próxima etapa é cadastrar um ou mais *flashcards*, associando-os ao conteúdo correspondente. Para tanto, o usuário deve acessar a página de *Flashcards*. De forma similar às categorias, conforme visualizado na Figura 21, é mostrada a quantidade de *flashcards* cadastrados na aplicação, juntamente com as informações de cada cartão de conteúdo, tais como: *Categoria*, *Frente* (do cartão ou pergunta) *Verso* (do cartão ou resposta) e *Ações*.

Figura 21 – Flashcard App: *flashcards*

 A imagem é uma captura de tela da interface de usuário da aplicação. No topo, há uma barra de navegação laranja com o nome "Flashcard App" e o perfil do usuário "Vitor Reiter". À esquerda, há um menu lateral com opções como "Tela Inicial", "Categorias", "Flashcards", "Revisões", "Lições" e "Perfil". O conteúdo principal mostra a seção "FlashCards" com um contador "87" e uma lista de cartões. Cada cartão tem uma aba de pesquisa "Pesquisar:" e uma lista de cartões com as seguintes colunas: "#", "Categoria", "Frente", "Verso" e "Ações".
 

#	Categoria	Frente	Verso	Ações
2	Geografia	Santa Catarina	Florianópolis	<a href="#">Editar</a> <a href="#">Excluir</a>
29	Geografia	Sergipe	Aracaju	<a href="#">Editar</a> <a href="#">Excluir</a>
30	Geografia	Tocantins	Palmas	<a href="#">Editar</a> <a href="#">Excluir</a>
5	Geografia	Acre	Rio Branco	<a href="#">Editar</a> <a href="#">Excluir</a>
6	Geografia	Alagoas	Maceió	<a href="#">Editar</a> <a href="#">Excluir</a>

Fonte: elaborado pelo autor.

Também é disponibilizada uma caixa de texto intitulada *Pesquisar* (Figura 21) para realizar pesquisa por cartões de conteúdo pertencentes a uma categoria informada. Ao inserir ou alterar um *flashcard*, o usuário precisa escolher uma categoria em *Selecionar Categoria* entre as disponíveis, bem como informar o conteúdo da *Frente* e do *Verso* do cartão (Figura 22). Caso não tenha sido cadastrada uma categoria anteriormente, não será possível efetuar o cadastro de *flashcards*.

Figura 22 – Flashcard App: formulário para novo *flashcard*

Novo FlashCard

Selecionar Categoria

Escolha categoria..

Frente

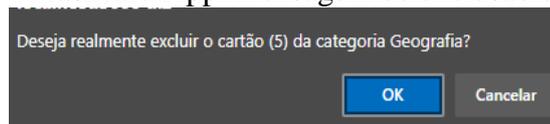
Verso

Cancelar Salvar

Fonte: elaborado pelo autor.

Para excluir um *flashcard*, o usuário deve sempre confirmar a ação, como mostrado na Figura 23.

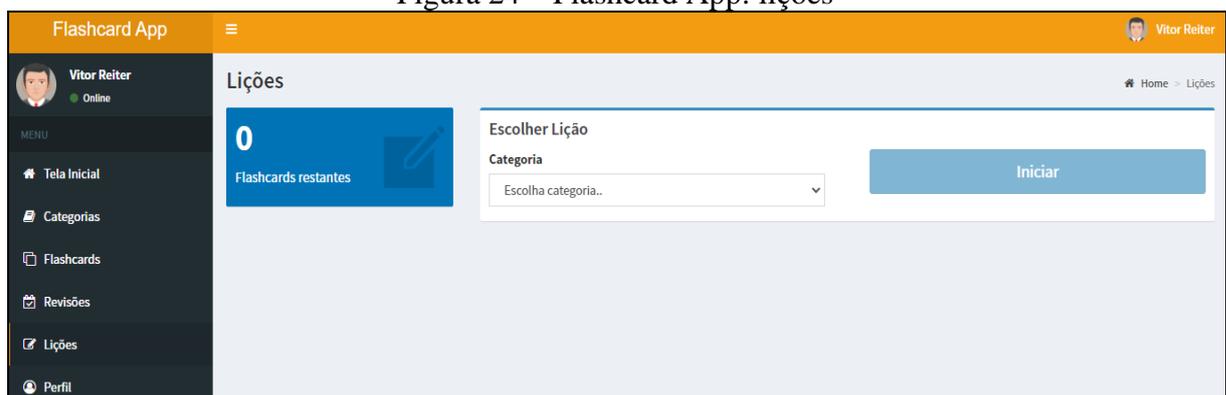
Figura 23 – Flashcard App: mensagem de exclusão de *flashcard*



Fonte: elaborado pelo autor.

Após cadastrar categorias e cartões de conteúdo, o usuário pode fazer lições. Ao acessar a opção *Lições*, o usuário é redirecionado à tela correspondente (Figura 24). Enquanto uma lição não for iniciada, nenhum cartão de conteúdo é apresentado (0 *Flashcards* restantes). Para iniciar uma lição, é necessário que o usuário informe a categoria que deseja estudar em *Escolher Lição*, selecionando algum item em *Categoria*. Ao escolher a categoria, caso a mesma possua *flashcards* associados, o botão *Iniciar* ficará disponível para começar a lição e a quantidade de cartões associados que devem ser estudados é apresentada em *Flashcards* restantes, sendo decrementada à medida que o usuário avança nos conteúdos estudados.

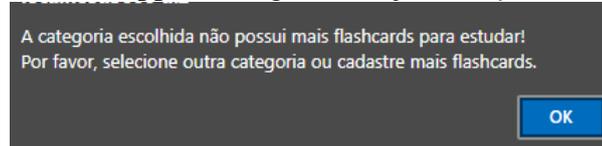
Figura 24 – Flashcard App: lições



Fonte: elaborado pelo autor.

No entanto, caso a categoria informada pelo o usuário, não possua *flashcards* associados, é apresentada uma mensagem solicitando que sejam associados cartões a esta categoria, como mostrado na Figura 25.

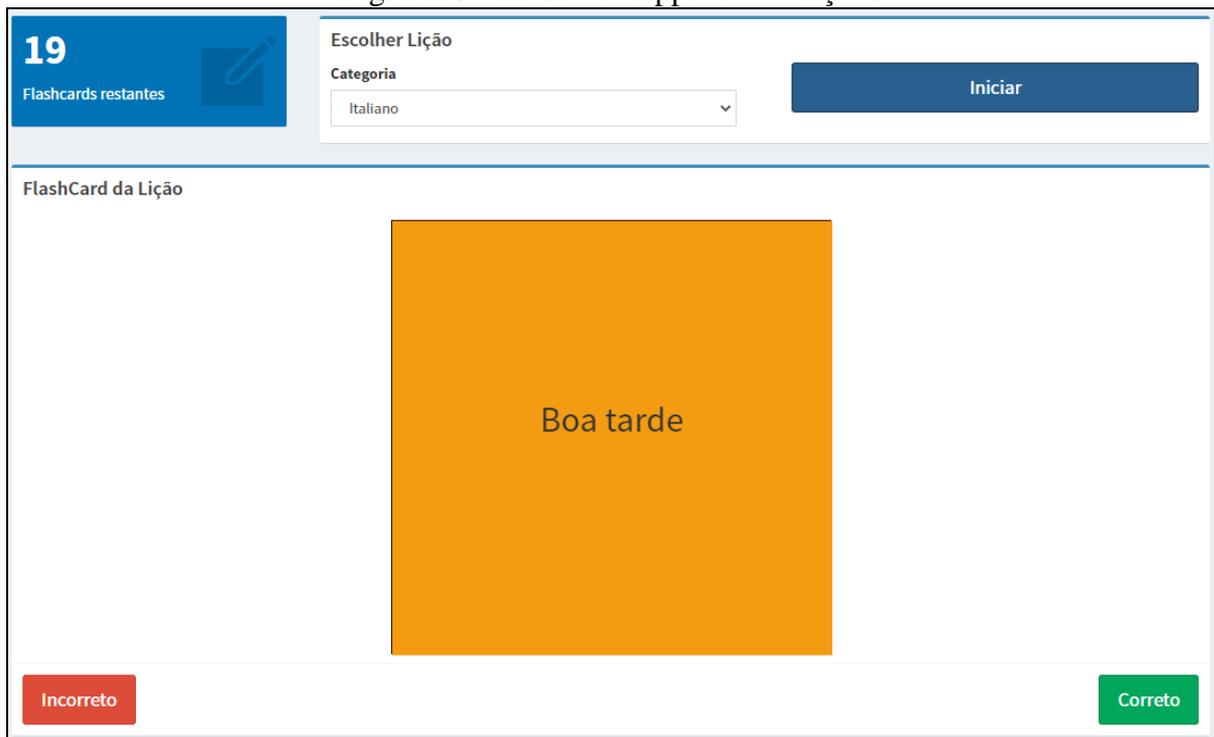
Figura 25 – Flashcard App: mensagem de lição sem *flashcards* na categoria



Fonte: elaborado pelo autor.

Ao iniciar uma lição, certos componentes em tela são atualizados e apresentados ao usuário, como a área de FlashCard da Lição, que mostra a frente (ou o verso) do cartão, dois botões, *Incorreto* e *Correto*, para a classificação do cartão, e a quantidade de *flashcards* restantes da lição indicada em *Flashcards restantes* (canto superior esquerdo, Figura 26). Os cartões de conteúdo são aleatoriamente embaralhados, após o usuário selecionar a categoria que deseja estudar.

Figura 26 – Flashcard App: área de lições



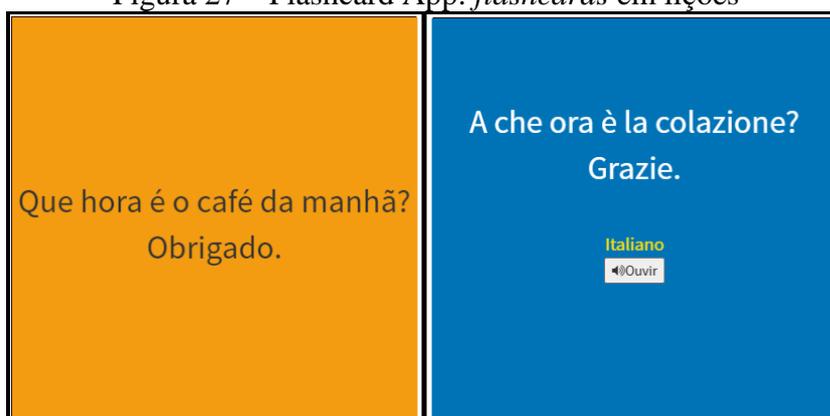
Fonte: elaborado pelo autor.

Para esta lição, foi selecionada a categoria *Italiano* (Figura 26), voltada à memorização de vocabulário para viagens à Itália. A frente de cada cartão contém o vocábulo no idioma nativo do estudante e o verso contém o vocábulo correspondente em italiano. Para prosseguir com a memorização e revisão deste conteúdo, é necessário que o estudante se recorde da resposta no verso, antes de virar o cartão. Caso o usuário tenha acertado a resposta,

basta classificar o cartão como *Correto* (botão à direita, Figura 26). Do contrário, se o usuário não acertar a resposta no verso, deve classificar como *Incorreto* (botão à esquerda, Figura 26). O próximo cartão da lição é então apresentado. Cartões classificados como corretos durante uma lição seguem para revisão, cartões classificados como incorretos são mantidos na lição para uma próxima sessão de estudos.

Para alternar entre a frente (na cor laranja) e o verso (na cor azul) dos cartões, deve-se passar com a seta do *mouse* sobre o mesmo revelando as informações dispostas no verso. Um cartão de conteúdo contém, em seu verso, a resposta, a categoria do conteúdo correspondente e um botão denominado *Ouvir*, que reproduz em áudio a informação do verso de acordo com o idioma cadastrado na categoria. Assim, com exemplo, a Figura 27 apresenta o cartão com a informação *Que hora é o café da manhã? Obrigado* (à esquerda) em sua frente e a resposta em italiano *A che ora è la colazione? Grazie* (à direita), dispondo juntamente com a categoria *Italiano* e o botão *Ouvir*.

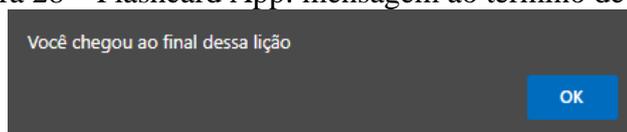
Figura 27 – Flashcard App: *flashcards* em lições



Fonte: elaborado pelo autor.

Após a passagem de todos os cartões da lição, o usuário é notificado de que aquela sessão terminou, como apresentado na Figura 28. Ao concluir uma lição, os cartões classificados corretamente aparecem somente nas sessões de revisão tendo os dias agendados como prevê o método SRS.

Figura 28 – Flashcard App: mensagem ao término de lições

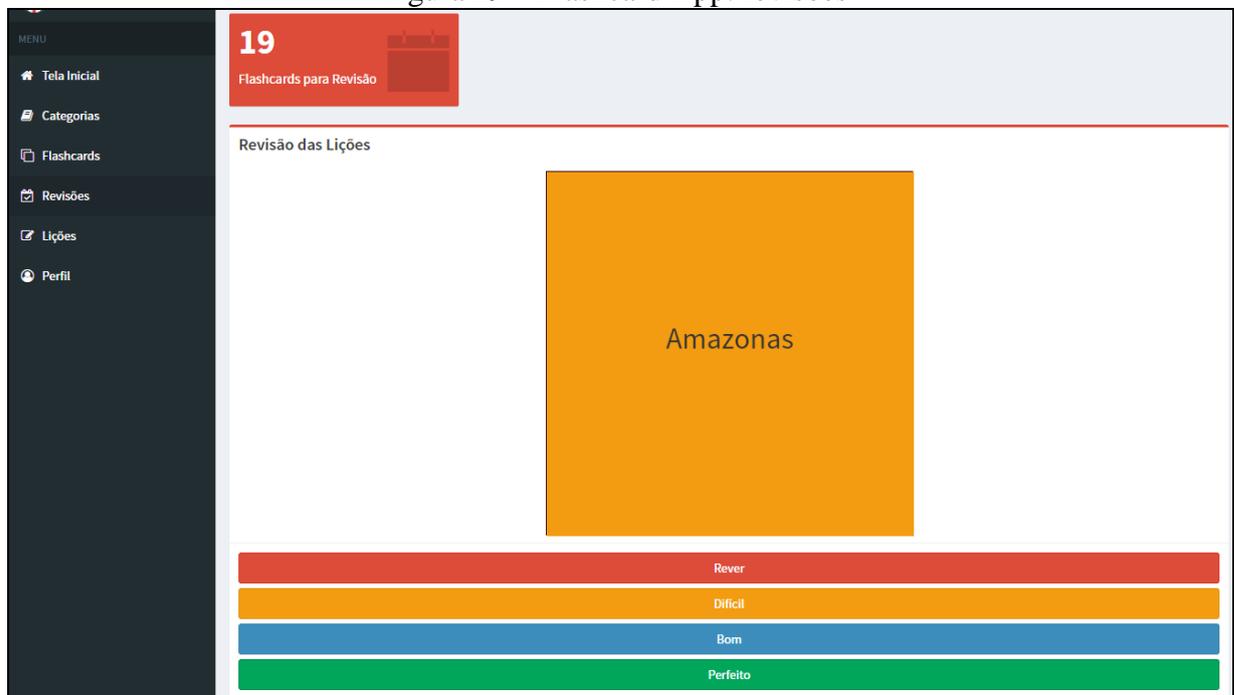


Fonte: elaborado pelo autor.

O acesso às revisões de conteúdo pode ser feito através da opção *Revisões* (menu à esquerda, Figura 29). Ao iniciar uma sessão, certos componentes em tela são atualizados e apresentados ao usuário, tais como: a área de *Revisão das Lições*, que possui um cartão

com frente e verso; os botões *Rever*, *Difícil*, *Bom* e *Perfeito*, para a classificação da recordação do conteúdo; e a quantidade de *flashcards* agendados para sessão atual, indicada em *Flashcards para Revisão* (canto superior esquerdo, Figura 29). O agendamento do intervalo de dias, nas revisões de cada cartão, é calculado a partir da classificação que o usuário faz quanto à dificuldade de recordar o conteúdo (botões *Rever*, *Difícil*, *Bom* e *Perfeito*). Os cartões apresentados para o usuário numa sessão de revisão estão dispostos seguindo a ordem alfabética de categorias estudadas e são inicialmente filtrados pela data de agendamento, sendo que cartões com data inferior ou igual a data da sessão de revisão, são apresentados para revisão.

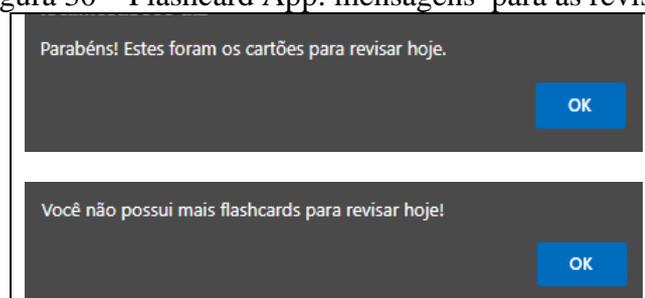
Figura 29 – Flashcard App: revisões



Fonte: elaborado pelo autor.

Os cartões em *Revisões* (Figura 29) possuem as mesmas informações apresentadas nas *lições*. À medida que os cartões são classificados, o número apresentado de *Flashcards para Revisão* (canto superior esquerdo, Figura 29) é decrementado até que a sessão encerre. Ao terminar de revisar todos os cartões agendados, o usuário é notificado de que a sessão acabou (mensagem superior, Figura 30). Uma notificação também é apresentada caso não haja cartões agendados para *rever* no dia (mensagem inferior, Figura 30).

Figura 30 – Flashcard App: mensagens para as revisões



Fonte: elaborado pelo autor.

### 3.4 RESULTADOS E DISCUSSÕES

O Quadro 7 traz um comparativo entre os trabalhos correlatos apresentados anteriormente e o projeto realizado, onde nas linhas encontram-se as características e nas colunas as aplicações descritas.

Quadro 7 – Comparativo entre os trabalhos correlatos e o projeto realizado

correlatos / características	Satoru	Duolingo	Anki	MosaLingua	Flashcard App
plataforma	web	web / aplicativo móvel	desktop / aplicativo móvel	aplicativo móvel	web
uso de cartões de conteúdo	sim	não	sim	sim	sim
recursos utilizados nos cartões/lições	texto, imagem, áudio	texto, imagem, áudio	texto, imagem, áudio, vídeo, texto HTML, marcação científica	texto, áudio	texto, áudio
conteúdo	japonês	idiomas (jogo)	qualquer assunto	palavras e frases de categorias	qualquer assunto
customização dos cartões	sim	não	sim	não	não
método de revisão	SRS	<i>deep learning</i>	SRS	SRS	SRS

Fonte: elaborado pelo autor.

Observa-se no Quadro 7 alguns pontos. Em primeiro lugar três correlatos são aplicativos móveis enquanto a plataforma de dois deles é web, assim como o projeto desenvolvido, sendo estes disponíveis a um grupo maior de usuários. Em segundo lugar, três correlatos e o Flashcard App utilizam-se de cartões de conteúdo para o que se propõem, considerando que a não utilização pode comprometer a memorização do conteúdo, já que lições podem ser muito mais extensas se comparadas a cartões. Outro ponto é em relação aos recursos usados. Todos usam texto associado a áudio, o que determinou com que o Flashcard App também atendessem a este quesito. Três aplicativos usam imagens, auxiliando ainda mais o usuário na memorização do conteúdo. Apenas o Anki faz uso de outros recursos, como vídeo,

texto HTML e marcação científica LaTeX. Todos voltam-se para o ensino de idiomas, sendo que o primeiro correlato se destina apenas ao ensino de japonês, o segundo propõe o ensino de idiomas de maneira lúdica e gamificada e o quarto segue o estudo de vocábulos de idiomas organizados em categorias pré-definidas. Já o Anki e o Flashcard App podem ser usados para estudar diversos assuntos, pois permitem que o usuário cadastre uma categoria de estudo com o assunto que desejar aprender. No Quadro 7, nota-se ainda que a possibilidade de customização nos cartões está presente apenas no Satoru e no Anki, sendo uma vantagem possibilitar que o usuário personalize seus estudos. Quatro ferramentas apresentadas utilizam-se do método de revisão SRS, sendo que apenas o Duolingo utiliza *deep learning* para memorização a longo prazo.

Para validar a eficácia do método SRS no Flashcard App, foram realizados testes com dois conteúdos distintos, sendo eles Geografia e Italiano, a fim de simular os agendamentos propostos pelo o algoritmo. O teste foi realizado localmente, já que o Flashcard App não se encontra hospedado em um servidor externo. O resultado do teste foi positivo, ou seja, o algoritmo criou intervalos de tempo espaçados a cada repetição dos vocábulos. Com o objetivo de visualizar os agendamentos feitos pelo algoritmo, foram capturadas imagens do banco de dados. As revisões em ambas as categorias iniciaram a partir do dia 25 de novembro, como pode ser verificado na coluna `register` das Figuras 31 a 34. Os cartões não foram cotidianamente revisados, o que gerou determinados acúmulos de cartões durante as sessões, mas comprovou a eficácia no filtro de cartões com datas inferiores a data atual da sessão.

Para cartões da categoria de Geografia, o objetivo era lembrar-se da capital (coluna `back`, Figura 31), tendo como informação o estado brasileiro (coluna `front`) apresentado. Após a recordação do conteúdo ter sido classificada com perfeita para todos os cartões agendados na sessão de revisão de 2020-12-06 (coluna `nextRevision`), obteve-se os resultados mostrados na Figura 32.

Figura 31 – Teste Flashcard App: Geografia 01

id	category	lang	front	back	timesRepeated	nextRevision	ef	register
2	Geografia	pt-BR	Santa Catarina	Florianópolis	3	2020-12-06	2.80	2020-11-28 16:15:28
7	Geografia	pt-BR	Amapá	Macapá	3	2020-12-06	2.70	2020-11-25 16:59:06
8	Geografia	pt-BR	Amazonas	Manaus	2	2020-12-06	2.60	2020-11-28 18:28:46
9	Geografia	pt-BR	Bahia	Salvador	3	2020-12-11	2.80	2020-11-28 16:16:23
10	Geografia	pt-BR	Ceará	Fortaleza	1	2020-12-06	2.50	2020-12-04 10:59:58
11	Geografia	pt-BR	Distrito Federal	Brasília	2	2020-12-06	2.70	2020-11-28 18:27:31
12	Geografia	pt-BR	Espírito Santo	Vitória	0	2020-12-06	2.50	2020-12-05 16:46:32
13	Geografia	pt-BR	Goiás	Goiânia	2	2020-12-06	2.70	2020-11-28 18:27:49
20	Geografia	pt-BR	Paraná	Curitiba	2	2020-12-06	2.70	2020-11-28 18:27:35
21	Geografia	pt-BR	Pernambuco	Recife	1	2020-12-06	2.50	2020-12-04 10:59:54
23	Geografia	pt-BR	Rio de Janeiro	Rio de Janeiro	2	2020-12-06	2.70	2020-11-28 16:16:32
25	Geografia	pt-BR	Rio Grande do...	Porto Alegre	1	2020-12-06	2.50	2020-12-04 11:00:39
28	Geografia	pt-BR	São Paulo	São Paulo	2	2020-12-06	2.70	2020-11-25 16:59:02
30	Geografia	pt-BR	Tocantins	Palmas	1	2020-12-06	2.60	2020-11-28 18:28:50

Fonte: elaborado pelo autor.

Figura 32 – Teste Flashcard App: Geografia 02

id	category	lang	front	back	timesRepeated	nextRevision	ef	register
2	Geografia	pt-BR	Santa Catarina	Florianópolis	4	2020-12-23	2.90	2020-11-28 16:15:28
7	Geografia	pt-BR	Amapá	Macapá	4	2020-12-23	2.80	2020-11-25 16:59:06
8	Geografia	pt-BR	Amazonas	Manaus	3	2020-12-12	2.70	2020-11-28 18:28:46
9	Geografia	pt-BR	Bahia	Salvador	3	2020-12-11	2.80	2020-11-28 16:16:23
10	Geografia	pt-BR	Ceará	Fortaleza	2	2020-12-07	2.60	2020-12-04 10:59:58
11	Geografia	pt-BR	Distrito Federal	Brasília	3	2020-12-12	2.80	2020-11-28 18:27:31
12	Geografia	pt-BR	Espírito Santo	Vitória	1	2020-12-07	2.60	2020-12-05 16:46:32
13	Geografia	pt-BR	Goiás	Goiânia	3	2020-12-12	2.80	2020-11-28 18:27:49
20	Geografia	pt-BR	Paraná	Curitiba	3	2020-12-12	2.80	2020-11-28 18:27:35
21	Geografia	pt-BR	Pernambuco	Recife	1	2020-12-07	2.50	2020-12-04 10:59:54
23	Geografia	pt-BR	Rio de Janeiro	Rio de Janeiro	3	2020-12-12	2.80	2020-11-28 16:16:32
25	Geografia	pt-BR	Rio Grande do...	Porto Alegre	2	2020-12-07	2.60	2020-12-04 11:00:39
28	Geografia	pt-BR	São Paulo	São Paulo	3	2020-12-12	2.80	2020-11-25 16:59:02
30	Geografia	pt-BR	Tocantins	Palmas	2	2020-12-07	2.70	2020-11-28 18:28:50

Fonte: elaborado pelo autor.

Comparando os registros das duas figuras, tem-se que cartões com repetições (coluna `timesRepeated`) iguais a 0 e 1 (registros com `id` igual a 10, 12, 21, 25 e 30, Figura 31) adicionaram um agendamento de 1 dia (coluna `nextRevision`, Figura 32). Por sua vez, cartões repetidos duas vezes, com os respectivos `ids` 8, 11, 13, 20, 23 e 28 (Figura 31), obtiveram um agendamento de 6 dias (Figura 32). Já os cartões repetidos pela terceira vez, `ids` 2 e 7 (Figura 31), receberam um intervalo espaçado de tempo de 17 dias (Figura 32), sendo este o maior intervalo adquirido durante os testes.

Para cartões da categoria Italiano, o objetivo era recordar-se da palavra ou sentença em italiano (coluna `back`, Figura 33), dada como informação a palavra ou sentença no idioma nativo do estudante (coluna `front`). Após a sessão de revisão agendada em 2020-12-06 (coluna `nextRevision`), com classificações entre Bom e Perfeito, observou-se os resultados computados na Figura 34.

Figura 33 – Teste Flashcard App: Italiano 01

id	category	lang	front	back	timesRepeated	nextRevision	ef	register
4	Italiano	it	Olá, meu nome é Vitor, sou brasileiro. Prazer!	Ciao, mi chiamo Vitor, sono brasiliano. Piacere.	1	2020-12-06	2.50	2020-11-28 18:20:09
60	Italiano	it	Quero uma passagem para a estação central de...	Vorrei un biglietto per Milano Centrale, solo and...	0	2020-12-06	2.50	2020-12-05 16:43:12
63	Italiano	it	Bom dia	Buongiorno	2	2020-12-06	2.60	2020-11-25 17:59:36
64	Italiano	it	Boa noite	Buonasera	2	2020-12-06	2.60	2020-11-25 17:43:23
69	Italiano	it	Posso provar?	Posso provare?	1	2020-12-06	2.50	2020-11-28 18:23:24
70	Italiano	it	Quanto custa?	Quanto viene?	2	2020-12-06	2.50	2020-11-25 17:42:30
77	Italiano	it	Que hora é o café da manhã? Obrigado.	A che ora è la colazione? Grazie.	0	2020-12-08	2.50	2020-12-05 07:49:59
80	Italiano	it	Oi e tchau	Ciao	2	2020-12-06	2.60	2020-11-25 18:01:56
81	Italiano	it	Muito obrigado	Grazie mille	2	2020-12-06	2.60	2020-11-25 17:44:34
82	Italiano	it	Por favor	Per favore	0	2020-12-06	2.50	2020-12-05 16:30:41
83	Italiano	it	Restaurante	Ristorante	2	2020-12-06	2.60	2020-11-28 16:19:56
84	Italiano	it	Não entendo italiano.	Non capisco italiano.	0	2020-12-06	2.50	2020-12-05 16:34:12
85	Italiano	it	Boa tarde	Buon pomeriggio	0	2020-12-06	2.50	2020-12-05 16:38:46
86	Italiano	it	Boa noite (quando estiver indo dormir)	Buonanotte	2	2020-12-06	2.60	2020-11-28 16:20:13
87	Italiano	it	Até logo	Arrivederci	2	2020-12-06	2.60	2020-11-25 17:44:47
88	Italiano	it	Janta	Cena	2	2020-12-06	2.60	2020-11-25 18:00:06

Fonte: elaborado pelo autor.

Figura 34 – Teste Flashcard App: Italiano 02

id	category	lang	front	back	timesRepeated	nextRevision	ef	register
4	Italiano	it	Olá, meu nome é Vitor, sou brasileiro. Prazer!	Ciao, mi chiamo Vitor, sono brasiliano. Piacere.	2	2020-12-07	2.60	2020-11-28 18:20:09
60	Italiano	it	Quero uma passagem para a estação central de...	Vorrei un biglietto per Milano Centrale, solo and...	1	2020-12-07	2.60	2020-12-05 16:43:12
63	Italiano	it	Bom dia	Buongiorno	3	2020-12-12	2.70	2020-11-25 17:59:36
64	Italiano	it	Boa noite	Buonasera	3	2020-12-12	2.70	2020-11-25 17:43:23
69	Italiano	it	Posso provar?	Posso provare?	2	2020-12-07	2.60	2020-11-28 18:23:24
70	Italiano	it	Quanto custa?	Quanto viene?	3	2020-12-12	2.60	2020-11-25 17:42:30
77	Italiano	it	Que hora é o café da manhã? Obrigado.	A che ora è la colazione? Grazie.	0	2020-12-08	2.50	2020-12-05 07:49:59
80	Italiano	it	Oi e tchau	Ciao	3	2020-12-12	2.70	2020-11-25 18:01:56
81	Italiano	it	Muito obrigado	Grazie mille	3	2020-12-12	2.70	2020-11-25 17:44:34
82	Italiano	it	Por favor	Per favore	1	2020-12-07	2.60	2020-12-05 16:30:41
83	Italiano	it	Restaurante	Ristorante	3	2020-12-12	2.70	2020-11-28 16:19:56
84	Italiano	it	Não entendo italiano.	Non capisco italiano.	1	2020-12-07	2.60	2020-12-05 16:34:12
85	Italiano	it	Boa tarde	Buon pomeriggio	1	2020-12-07	2.60	2020-12-05 16:38:46
86	Italiano	it	Boa noite (quando estiver indo dormir)	Buonanotte	3	2020-12-12	2.70	2020-11-28 16:20:13
87	Italiano	it	Até logo	Arrivederci	3	2020-12-12	2.70	2020-11-25 17:44:47
88	Italiano	it	Janta	Cena	3	2020-12-12	2.70	2020-11-25 18:00:06

Fonte: elaborado pelo autor.

Comparando os dois registros (Figuras 33 e 34), tem-se que cartões repetidos (coluna `timesRepeated`) pela primeira vez (registros 4, 60, 69, 77, 82, 84 e 85, Figura 33), também adicionaram 1 dia após a sessão atual (coluna `nextRevision`, Figura 34), sendo que aqueles repetidos pela segunda vez (registros 63, 64, 70, 80, 81, 83, 86, 87 e 88, Figura 33), receberam um acréscimo de 6 dias (Figura 34), mantendo o intervalo espaçado previsto pelo método SRS. Vale salientar que os valores obtidos no fator de facilidade (`ef`) em cartões de Geografia foram maiores do que os associados à categoria de Italiano, por não terem sido classificados com qualidade inferior a 3 (Rever ou Difícil), o que fez com que o valor não retornasse em 2,5 e repetição 1. Cita-se, como contraexemplo, o cartão com `id` igual a 4 da categoria Italiano, cuja etapa de revisão iniciou em 2020-11-28 (coluna `register`, Figura 33). Como a resposta desse cartão (coluna `back`) não foi recordada com facilidade, foi classificado com qualidade inferior a 3 em sessões de revisão anteriores a 2020-12-06, obtendo valores de `ef` e `timesRepeated` estagnados em 1 e 2,5 respectivamente (Figura 33).

## 4 CONCLUSÕES

Neste trabalho de conclusão de curso foi apresentado o Flashcard App, uma aplicação web para a memorização de vocabulário, utilizando o método SRS. Para atingir os objetivos propostos, foi permitida a criação de cartões de conteúdo com vocábulos a serem estudados, podendo ser termos para estudar outro idioma, a exemplo dos cartões da categoria Italiano, descritos anteriormente, ou até mesmo conceitos diversos, a exemplo dos cartões de Geografia. Para tanto, foi disponibilizado o recurso de escolha do idioma ao efetuar o cadastro de uma categoria, que, posteriormente, é associado a cartões de conteúdo. Houve, contudo, por parte do autor do trabalho, a escolha por restringir os idiomas estrangeiros disponíveis na ferramenta, sendo eles apenas: Alemão, Espanhol, Francês, Inglês, Italiano e Português. Esta escolha se deu por conta da API de áudio usada no projeto não possuir suporte a todos os idiomas conhecidos e estar em fase de testes. Mesmo assim, ao criar e associar os cartões a um determinado assunto e seu respectivo idioma, além do texto, as respostas dos cartões têm também o áudio correspondente no idioma escolhido.

Por fim, para validar o uso da aplicação para memorização, foram realizados testes para a memorização de vocábulos e sentenças do idioma Italiano e de estados e suas respectivas capitais. Os testes apresentaram intervalos de tempo maiores para a revisão de palavras e sentenças menores, assim como para estados e capitais, visto que informações como estas são mais fáceis de serem recordadas, logo obtendo uma classificação mais alta nas revisões, ao contrário de sentenças mais longas, o que também é previsto pelo método SRS.

A contribuição tecnológica desse trabalho foi mostrar uma aplicação web responsiva de aprendizagem na área da computação, baseada em um método de estudo, usando tecnologias como: NodeJS, Redis, API SpeechSynthesisUtterance e MySQL.

As dificuldades enfrentadas no desenvolvimento deste trabalho envolveram a inexperiência no desenvolvimento do *back-end*, bem como no uso de um banco de dados relacional. Contudo, mediante ao que foi exposto, foi possível identificar extensões que poderão ser incorporadas no trabalho, a fim de aperfeiçoá-lo.

### 4.1 EXTENSÕES

Apesar do projeto possuir resultados positivos, alguns pontos podem ser melhorados permitindo explorar ainda mais a capacidade de quem utiliza a aplicação, bem como trazer inovações e funcionalidades pertinentes ao projeto. São elas:

- a) permitir que o usuário use recursos visuais como imagens, vídeos ou desenhos próprios em seus cartões de conteúdo;

- b) possibilitar que o usuário inverta frente e verso dos *flashcards* registrados durante lições/revisões, assegurando que o aprendizado seja feito nos dois sentidos e com o dobro de informação;
- c) incluir limite máximo na adição de cartões de conteúdo durante lições ou revisões, garantindo sessões pouco cansativas;
- d) permitir que o usuário acompanhe seu progresso, mostrando cartões memorizados a longo prazo após o período de revisões;
- e) reformular a área de lições da aplicação, atribuindo não só a classificação entre correto e incorreto, como também a possibilidade da escrita das respostas do verso, a fim de tornar mais assertivo o estudo dos cartões com maior dificuldade no aprendizado;
- f) validar a aplicação com pessoas interessadas na memorização de assuntos ou em aprender vocabulários estrangeiros;
- g) possibilitar que o usuário altere sua senha em caso de esquecimento.

## REFERÊNCIAS

- ANKI 2.1: user manual. 2.1. [S.l.], 2018. Disponível em: <https://apps.ankiweb.net/docs/manual.html>. Acesso em: 14 set. 2019.
- BARROS, T. **Como usar o Duolingo para estudar e aprender idiomas no celular**. [S.l.], 2016. Disponível em: <https://www.techtudo.com.br/dicas-e-tutoriais/noticia/2015/02/como-usar-o-duolingo-para-estudar-e-aprender-idomas-no-celular.html>. Acesso em: 22 set. 2019.
- CASSIMIRO, W. **A curva de esquecimento de Ebbinghaus**. [S.l.], 2018. Disponível em: <https://espresso3.com.br/a-curva-de-esquecimento-de-ebbinghaus/>. Acesso em: 21 set. 2019.
- GARCIA, M. **SRS: nunca mais esqueça o que aprendeu**. [S.l.], 2014. Disponível em: <http://aprenderlinguas.com.br/srs-nunca-mais-esqueca-o-que-aprendeu/>. Acesso em: 3 dez. 2020.
- LAMERI, D. C. **Ambiente de aprendizado de japonês com foco na memorização através de sistema de repetição espaçada**. 2017. 46 f. Trabalho de Conclusão de Curso (Curso de Tecnologia em Sistemas de Computação) - Universidade Federal Fluminense, Niterói.
- LOMB, K. **Polyglot: how I learn languages**. Berkeley: TESL-EJ, 2008.
- MANFRIN, F. **Como usar o Anki para aprender inglês ou passar em concursos públicos**. [S.l.], 2015. Disponível em: <http://www.loucosporconcursos.com/anki-como-usar-ingles-concursos/>. Acesso em: 16 set. 2019.
- MANFRIN, F. **O segredo de MosaLingua, por um especialista em aprendizagem**. [S.l.], 2018. Disponível em: <https://www.mosalingua.com/pt/segredo-mosalingua-um-especialista-aprendizagem/>. Acesso em: 16 set. 2019.
- MARTINS, H. F. A. **Mosalingua: aplicativo para aprender idiomas**. [S.l.], 2015. Disponível em: <https://www.meutedio.com.br/2015/04/mosalingua-aplicativo-para-aprender-idomas.html>. Acesso em: 22 set. 2019.
- MATHILDE, M. **Duolingo X MosaLingua: qual o melhor aplicativo para aprender idiomas?** [S.l.], 2018. Disponível em: <https://www.mosalingua.com/pt/comparativo-duolingo-mosalingua/>. Acesso em: 16 set. 2019.
- MEEDER, B. **What is the technology stack behind Duolingo?** Quora, 2013. Disponível em: <https://www.quora.com/What-is-the-technology-stack-behind-Duolingo>. Acesso em: 22 set. 2019.
- MICHELOT, S. **A curva do esquecimento: não perca seu tempo aprendendo pra nada**. [S.l.], 2018a. Disponível em: <https://www.mosalingua.com/pt/curva-do-esquecimento/>. Acesso em: 21 set. 2019.
- MICHELOT, S. **O Sistema de Repetição Espaçada (SRS): memorizar para jamais esquecer**. [S.l.], 2018b. Disponível em: <https://www.mosalingua.com/pt/o-sistema-de-repeticao-espacada-memorizar-para-jamais-esquecer/>. Acesso em: 21 set. 2019.
- PERANANDAM, C. **AI helps Duolingo personalize language learning: the learning behind the lingo**. Wired, 2018. Disponível em: <https://www.wired.com/brandlab/2018/12/ai-helps-duolingo-personalize-language-learning/>. Acesso em: 26 set. 2019.

SANTOS, J. R. A.; BARROS, D. S. A. Técnicas de estudos e gestão do tempo no auxílio à aprendizagem de fundamentos de algoritmo e lógica aplicada a computação. In: CONGRESS OF INDUSTRIAL MANAGEMENT AERONAUTICAL TECHNOLOGY, 5., 2018, São José dos Campos. **Anais eletrônicos...** São José dos Campos: FATEC, 2018. Não paginado. Disponível em: <https://publicacao.cimatech.com.br/index.php/cimatech/article/view/115/4>. Acesso em: 16 set. 2019.

SANTOS, J. R. A.; OLIVEIRA JUNIOR, J. A. Técnicas de estudo e gestão do tempo na aprendizagem de fundamentos de matemática e algoritmos nos cursos técnicos integrado ao ensino médio do IFPA – Campus Itaituba. In: COLÓQUIO INTERNACIONAL “EDUCAÇÃO E CONTEMPORANEIDADE”, 12., 2018, São Cristóvão. **Anais eletrônicos...** Aracajú, Educon, 2018. p. 1-11. Disponível em: <https://ri.ufs.br/bitstream/riufs/9461/25/24.pdf>. Acesso em: 20 set. 2019.

WOZNIAK, P. **The true history of spaced repetition**. [S.l.], 2018. Disponível em: <https://www.supermemo.com/pl/articles/history>. Acesso em: 4 dez. 2020.

## APÊNDICE A – Descrição dos Casos de Uso

Este Apêndice apresenta a descrição dos seis casos de uso deste projeto: UC01: Cadastrar usuário (Quadro 8), UC02: Efetuar login/logout (Quadro 9), UC03: Manter categorias (Quadro 10), UC04: Manter flashcards (Quadro 11), UC05: Fazer lições (Quadro 12) e UC06: Realizar revisões (Quadro 13).

Quadro 8 – UC01: Cadastrar usuário

UC01 – Cadastrar usuário	
<b>Descrição</b>	Permitir que o usuário efetue seu cadastro para utilização do Flashcard App.
<b>Autor</b>	Usuário
<b>Cenário Principal</b>	1. Usuário preenche nome, e-mail e senha e pressiona o botão <i>Cadastrar</i> .
<b>Pré-condições</b>	Conexão com o banco de dados.
<b>Pós-condições</b>	Usuário cadastrado.

Fonte: elaborado pelo autor.

Quadro 9 – UC02: Efetuar login/logout

UC02 – Efetuar login/logout	
<b>Descrição</b>	Permitir que o usuário efetue login/logout no Flashcard App.
<b>Autor</b>	Usuário
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. Usuário preenche os campos de login com e-mail e senha e clica no botão <i>Entrar</i>.</li> <li>2. Se não houver erro no preenchimento dos campos, o usuário é redirecionamento para o <i>dashboard</i> contendo uma visão geral das informações na aplicação.</li> <li>3. Usuário clica em seu nome no canto superior direito e em seguida no botão <i>Sair</i> do <i>drop-down</i> de perfil.</li> <li>4. Usuário é redirecionado para tela de login da aplicação.</li> </ol>
<b>Pré-condições</b>	Conexão com o banco de dados. Usuário cadastrado.
<b>Pós-condições</b>	Usuário logado/deslogado.

Fonte: elaborado pelo autor.

Quadro 10 – UC03: Manter categorias

UC03 – Manter categorias	
<b>Descrição</b>	Permitir que o usuário crie, altere ou exclua categorias de assuntos no Flashcard App.
<b>Autor</b>	Usuário
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. Usuário seleciona a opção <i>Categorias</i> no menu esquerdo e é redirecionado para página de categorias.</li> <li>2. O sistema apresenta a lista de categorias existentes, juntamente com opções para pesquisar, editar, excluir e criar uma categoria. <ol style="list-style-type: none"> <li>2.1. Para criar uma categoria, usuário deverá informar o nome, a descrição e o idioma do assunto a ser estudado e clicar em <i>salvar</i>.</li> <li>2.2. Para alterar uma categoria existente, usuário deverá clicar no botão <i>editar</i> da linha onde encontra-se a categoria que deseja alterar, podendo mudar o nome, a descrição ou o idioma e clicar em <i>salvar</i>.</li> <li>2.3. Para excluir uma categoria existente, usuário deverá clicar no botão <i>excluir</i> da linha onde encontra-se a categoria que deseja excluir e confirmar a ação.</li> </ol> </li> </ol>
<b>Pré-condições</b>	Conexão com banco de dados. Usuário logado.
<b>Pós-condições</b>	Categorias atualizadas no banco de dados.

Fonte: elaborado pelo autor.

Quadro 11 – UC04: Manter flashcards

UC04 – Manter flashcards	
<b>Descrição</b>	Permitir que o usuário crie, altere ou exclua cartões de conteúdo no Flashcard App.
<b>Autor</b>	Usuário
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. Usuário seleciona a opção <i>Flashcards</i> no menu esquerdo e é redirecionado para a página de cartões de conteúdo (<i>flashcards</i>).</li> <li>2. O sistema apresenta a lista de <i>flashcards</i> existentes, juntamente com opções para pesquisar, editar, excluir e criar um <i>flashcard</i>. <ol style="list-style-type: none"> <li>2.1. Para criar um <i>flashcard</i>, usuário deverá informar a categoria do assunto que será estudado, a pergunta que constará na frente do cartão e a resposta para o verso e clicar em <i>salvar</i>.</li> <li>2.2. Para alterar um <i>flashcard</i> existente, usuário deverá clicar no botão <i>editar</i> da linha onde encontra-se o <i>flashcard</i> que deseja alterar, podendo mudar a categoria, a pergunta ou a resposta e clicar em <i>salvar</i>.</li> <li>2.3. Para excluir um <i>flashcard</i> existente, usuário deverá clicar no botão <i>excluir</i> da linha onde encontra-se o <i>flashcard</i> que deseja excluir e confirmar a ação.</li> </ol> </li> </ol>
<b>Pré-condições</b>	Conexão com banco de dados. Usuário logado. Categoria previamente cadastrada.
<b>Pós-condições</b>	<i>Flashcards</i> atualizados no banco de dados.

Fonte: elaborado pelo autor.

Quadro 12 – UC05: Fazer lições

UC05 – Fazer lições	
<b>Descrição</b>	Permitir que o usuário estude lições no Flashcard App.
<b>Autor</b>	Usuário
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. Usuário seleciona a opção de Lições no menu esquerdo e é redirecionado para a página de lições.</li> <li>2. Usuário seleciona a categoria que deseja estudar e clica em iniciar.</li> <li>3. O sistema filtra os <i>flashcards</i> pela categoria e os apresenta um a um, juntamente com as opções <i>incorreto</i> e <i>correto</i>.</li> <li>4. Usuário classifica a resposta dada para cada cartão estudado como incorreta ou correta.</li> <li>5. À medida que o usuário avança na lição, o sistema apresenta a quantidade de <i>flashcards</i> restantes.</li> <li>6. Após o último cartão, o sistema informa que a lição terminou.</li> </ol>
<b>Pré-condições</b>	Conexão com banco de dados. Usuário logado. <i>Flashcards</i> previamente cadastrados.
<b>Pós-condições</b>	Lições atualizadas no banco de dados.

Fonte: elaborado pelo autor.

Quadro 13 – UC06: Realizar revisões

UC06 – Realizar revisões	
<b>Descrição</b>	Permitir que o usuário revise <i>flashcards</i> no Flashcard App.
<b>Autor</b>	Usuário
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. Usuário seleciona a opção Revisões no menu esquerdo e é redirecionado para a página de revisões.</li> <li>2. O sistema consulta os agendamentos de revisões existentes.           <ol style="list-style-type: none"> <li>3.1. Se não houver agendamento, o sistema informa que não há revisões agendadas.</li> <li>3.2. Se houver agendamento, o sistema apresenta os <i>flashcards</i> agendados para a data corrente, juntamente com as opções <i>rever</i>, <i>difícil</i>, <i>bom</i> e <i>perfeito</i>.               <ol style="list-style-type: none"> <li>3.2.1. O usuário classifica os <i>flashcards</i> de acordo com o grau de dificuldade para recordar o conteúdo em questão.</li> <li>3.2.2. À medida que o usuário avança na revisão, o sistema apresenta a quantidade de <i>flashcards</i> restantes.</li> <li>3.2.3. Após o último cartão, o sistema informa que a revisão terminou.</li> </ol> </li> </ol> </li> </ol>
<b>Pré-condições</b>	Conexão com banco de dados. Usuário logado. Lição previamente estudada, com agendamento para revisão.
<b>Pós-condições</b>	Revisões atualizadas no banco de dados.

Fonte: elaborado pelo autor.

## APÊNDICE B – Dicionário de Dados

Este Apêndice apresenta as definições e representações dos dados utilizados para o armazenamento das informações, juntamente com os tipos usados, quais sejam:

- a) `Date`: data civil sem considerar fuso horário;
- b) `Decimal`: tipo numérico para valores não-inteiros;
- c) `Int`: tipo numérico para valores inteiros;
- d) `String`: cadeia de caracteres com tamanho variável;
- e) `Timestamp`: data e hora levando em consideração o fuso horário.

Os quadros a seguir exibem a estrutura das tabelas `tb_users`, `tb_categories`, `tb_flashcards` e `tb_lessons`.

Quadro 14 – Tabela `tb_users`

Tabela: <code>tb_users</code>		
Descrição: Usada para armazenamento dos dados dos usuários.		
CAMPOS		
atributo	tipo	descrição
<code>id</code>	<code>Int</code>	identificador único do registro
<code>name</code>	<code>String</code>	nome do usuário
<code>email</code>	<code>String</code>	e-mail do usuário
<code>password</code>	<code>String</code>	senha do usuário
<code>register</code>	<code>Timestamp</code>	data/hora do registro na tabela

Fonte: elaborado pelo autor.

Quadro 15 – Tabela `tb_categories`

Tabela: <code>tb_categories</code>		
Descrição: Usada para armazenamento das categorias.		
CAMPOS		
atributo	tipo	descrição
<code>id</code>	<code>Int</code>	identificador único do registro
<code>name</code>	<code>String</code>	nome da categoria
<code>description</code>	<code>String</code>	descrição da categoria
<code>lang</code>	<code>String</code>	idioma da categoria
<code>register</code>	<code>Timestamp</code>	data/hora do registro na tabela

Fonte: elaborado pelo autor.

Quadro 16 – Tabela `tb_flashcards`

Tabela: <code>tb_flashcards</code>		
Descrição: Usada para armazenamento dos <i>flashcards</i> .		
CAMPOS		
atributo	tipo	descrição
<code>id</code>	<code>Int</code>	identificador único do registro
<code>category</code>	<code>String</code>	nome da categoria a qual o <i>flashcard</i> está associado
<code>lang</code>	<code>String</code>	idioma da categoria
<code>front</code>	<code>String</code>	frente do cartão de conteúdo
<code>back</code>	<code>String</code>	verso do cartão de conteúdo
<code>register</code>	<code>Timestamp</code>	data/hora do registro na tabela

Fonte: elaborado pelo autor.

Quadro 17 – Tabela `tb_lessons`

<b>Tabela:</b> <code>tb_lessons</code>		
<b>Descrição:</b> Usada para armazenamento dos dados de revisões e lições feitas.		
<b>CAMPOS</b>		
<b>atributo</b>	<b>tipo</b>	<b>descrição</b>
<code>id</code>	<code>Int</code>	identificador único do registro
<code>category</code>	<code>String</code>	nome da categoria a qual o <i>flashcard</i> está associado
<code>lang</code>	<code>String</code>	idioma da categoria
<code>front</code>	<code>String</code>	frente do cartão de conteúdo
<code>back</code>	<code>String</code>	verso do cartão de conteúdo
<code>timesRepeated</code>	<code>Int</code>	número de vezes que o cartão foi revisado
<code>nextRevision</code>	<code>Date</code>	data da próxima revisão
<code>ef</code>	<code>Decimal</code>	valor numérico do fator de facilidade do cartão
<code>register</code>	<code>Timestamp</code>	data/hora do registro na tabela

Fonte: elaborado pelo autor.