CARDERY, APLICAÇÃO MOBILE PARA ESTUDO DA LÍNGUA INGLESA:
RELATÓRIO DE DESENVOLVIMENTO DO FRONT-END

Bertonni Thiago de Souza Paz
Orientador: Prof Me. Ivo Felix Gualberto de Sá

Resumo
Este trabalho consiste em um relatório sobre o desenvolvimento do front-end de uma aplicação mobile que tem a proposta de auxiliar o aprendizado de uma língua estrangeira, nesse caso, o inglês. O aplicativo tem como foco a criação de flashcards, que deverão auxiliar o usuário a expandir seu vocabulário, revisando seus cards periodicamente para que o conteúdo estudado seja relembrado frequentemente. No desenvolvimento, processo que terá linguagens, recursos utilizados e telas desenvolvidas apresentados neste trabalho, foram utilizadas tecnologias como o React Native, framework baseado em React, uma biblioteca JavaScript que possibilita a criação de aplicativos para Android e iOS sem a necessidade de escrever código nativo para essas plataformas. Visando facilitar o aprendizado do usuário, foram construídas telas simples, que não exigem muito esforço para serem utilizadas, focando no objetivo principal, que é o estudo da língua inglesa. Como resultado, foi criado um aplicativo, a ser disponibilizado de forma gratuita ao público, que se pretende ser de grande valia para aqueles que estudam a língua inglesa em todo o mundo.

Palavras-chave: Aprendizagem de língua inglesa. Aplicação mobile. Front-end. JavaScript.

Abstract
This work consists of a report on the development of the front-end of a mobile application that aims to assist the learning of a foreign language, in this case, English. The application focuses on creating flashcards, which should help the user expand their vocabulary, reviewing their cards periodically so that the studied content is often remembered. Through the application development, a process that will have languages, resources and developed screens presented in this paper, one of the technologies used was React Native, a React-based framework, a JavaScript library that enables the creation of Android and iOS applications without the need to write native code for these platforms. In order to facilitate the user's learning process,

simple screens have been built, which do not require much effort to be used, focusing on the main objective that is the study of the English language. As a result, an application was created to be made available for free to the public, so that it can be valuable to those who study the English language around the world.

Keywords: English language learning. Mobile application. Front-end. JavaScript.

1. Introdução

Estudar a língua inglesa é importante por muitas razões; uma delas é que o ato de aprender algo novo, por si só, é ótimo para exercitar o cérebro e, além disso, aprender inglês, a língua mais falada no mundo (BERLITZ, 2023), pode dar a oportunidade de conhecer outros países e culturas, conseguir o emprego dos sonhos etc.

Quanto à importância da língua inglesa, Morato (2020, p. 4) aponta que

> é indiscutível a importância da LI (Língua Inglesa) e sua utilização em diversas áreas de atuação profissional em todo o mundo, o que lhe garante grande aderência social como pré-requisito para ser inserido na sociedade globalizada.

Para o profissional de TI, aprender inglês é quase obrigatório, pois as linguagens de programação, ferramentas, tecnologias e suas documentações são em inglês, além dos fóruns de ajuda que, em sua maioria, estão neste mesmo idioma.

Sobre o inglês para o profissional de TI, Franco *et al* (2016) afirma que “o inglês é a língua predominante no mundo globalizado; a área de Tecnologia da Informação (T.I.), por estar em constante evolução, não se faz uma exceção.”

As pessoas podem se questionar sobre como aprender um novo idioma se o tempo é curto e o recurso financeiro também, mas existem alternativas para solucionar esses problemas como o estudo de um idioma através de um aplicativo, por exemplo. Muitas pessoas têm um aparelho celular com acesso à internet e podem estudar em qualquer lugar, a qualquer hora, como na ida/vinda do trabalho, no metrô, na fila do banco etc.

Sobre o uso de aplicativos para o aprendizado, Borges (2016, p. 19) aponta que “estas ferramentas atuais podem ampliar e melhorar a educação, otimizar o tempo na sala de aula, facilitar a aprendizagem de alunos, melhorar a comunicação e a aprendizagem no ensino.”

Nesse contexto, este trabalho tem como objetivo principal desenvolver o front-end de uma aplicação mobile para auxílio no estudo da língua inglesa. Os objetivos específicos são os seguintes:

- Definir as funcionalidades presentes no aplicativo;
- Mapear os recursos necessários para o desenvolvimento da aplicação;
- Desenvolver as telas do aplicativo.

Neste relatório, serão apresentados o conceito de front-end utilizado por este trabalho, o processo através do qual foram desenvolvidas as telas do aplicativo, indicando as linguagens e recursos utilizados. Por fim, serão apresentadas as considerações finais, em que são relatadas, entre outros tópicos, as perspectivas futuras para o aplicativo desenvolvido.

2. O front-end do Cardery

Nesta seção, serão apresentados os recursos utilizados no desenvolvimento, além da definição de front-end e uma explicação sucinta sobre as funcionalidades e telas construídas.

2.1 Conceito de front-end

Mario Souto (2023), sobre o que é front-end, pontua que “podemos classificar como a parte visual de um site, aquilo com que conseguimos interagir”.

O processo de desenvolvimento do front-end de uma aplicação web ou, no nosso caso, de um aplicativo mobile, consiste no desenvolvimento das telas e funções com as quais o usuário final vai interagir, a fim de alcançar o seu objetivo ao utilizar a aplicação. Normalmente esse processo se dá através da transformação do

protótipo criado pelo design em telas funcionais, utilizando as tecnologias apropriadas.

Desenvolver uma aplicação moderna, simples de usar e rápida é fundamental para que a aplicação seja bem aceita pelos usuários, pois se o aplicativo demora muito para realizar determinada tarefa, o usuário, impaciente, pode acabar deixando de usar a aplicação. Segundo Pedro Souza (2016, p. 22):

> o estilo de desenvolvimento moderno trabalha de forma assíncrona, buscando não bloquear a UI (interface do usuário) durante as requisições. Para isto, são utilizadas linguagens client-side (do lado do cliente) nativas dos navegadores e extensamente suportadas, como o JavaScript.

2.2 Linguagens e Recursos Utilizados

O Javascript vem sendo uma das linguagens de programação mais utilizadas pelos desenvolvedores nos últimos anos e, em 2022, foi a linguagem de programação mais utilizada pelos desenvolvedores, segundo pesquisa realizada pelo Stack Overflow (fórum em desenvolvedores tiram suas dúvidas sobre problemas de desenvolvimento em geral). Por esse motivo, entre outros, foi escolhida para desenvolver o front-end da aplicação. O framework utilizado foi o React Native, um framework baseado em React (biblioteca Javascript de código aberto desenvolvida pelo Facebook), com o qual é possível escrever código React, que é “convertido” para as plataformas nativas como o Android e o iOS, por exemplo.

David Flanagan (2004, p. 1) pontua que “JavaScript é uma linguagem de alto nível, dinâmica, interpretada e não tipada, conveniente para estilos de programação orientados a objetos e funcionais”.

De acordo com MDN (Mozilla Developer Network) Web Docs (2022), “O JavaScript é uma linguagem baseada em protótipos, multi-paradigma e dinâmica, suportando estilos de orientação a objetos, imperativos e declarativos”. Por ser uma linguagem de tipagem dinâmica, o JavaScript permite que uma variável possa receber um valor de qualquer tipo e em qualquer momento do código. Isso é bom, mas pode atrapalhar o desenvolvimento em um certo momento, porque não se sabe exatamente qual é o valor que aquela variável pode ter ou que uma determinada função pode retornar.

Para facilitar o desenvolvimento, foi utilizado o Typescript, que é uma extensão do JavaScript e permite o desenvolvimento de aplicações JavaScript de larga escala. Segundo Gavin Bierman (2014, p. 1, tradução nossa)[1], “o TypeScript enriquece o JavaScript com um sistema de módulos, classes interfaces e um sistema de tipo estático”.

O editor de texto utilizado foi o VSCode (Visual Studio Code), que oferece ferramentas que auxiliam e agilizam o desenvolvimento. Para estilização, foi utilizada a biblioteca Native Base, que fornece componentes prontos, com um estilo padrão e que permitem a customização.

Por fim, foi utilizado o Expo para gerenciar o projeto. O Expo é uma ferramenta utilizada no desenvolvimento mobile que permite o acesso facilitado às APIs nativas do dispositivo como a câmera, microfone e gps, por exemplo, sem precisar instalar dependências extras.

Antes de descrever as telas desenvolvidas, será descrito como foi configurada a biblioteca de componentes utilizada para criar as telas, o Native Base. Essa biblioteca fornece diversos componentes e um tema padrão de cores, fontes, tamanhos etc., e, para atender melhor à proposta do app, foi criado um tema customizado, no qual foram inseridas as cores de acordo com o protótipo desenvolvido, além dos tamanhos, famílias e tipos das fontes. Para que esse tema customizado seja utilizado pela aplicação, foi preciso criar um arquivo, inserir os estilos customizados e disponibilizar esse arquivo para que pudesse ser utilizado em toda a aplicação.

2.3 Telas desenvolvidas

Foram desenvolvidas, inicialmente, oito telas, cada uma com sua respectiva função para que o usuário possa alcançar seu objetivo final. O foco principal foi criar telas objetivas, que não interferissem negativamente no processo de aprendizagem do usuário.

2.3.1 Landing page

[1] No original: TypeScript enriches JavaScript with a module system, classes, interfaces, and a static type system.

Após a configuração, foi desenvolvida a primeira tela, que é a tela inicial da aplicação (Figura 1). Essa tela possui a imagem de um globo com destaque para o mapa dos Estados Unidos, um dos países que têm a língua inglesa como língua nativa. Além disso, a tela contém o logo da aplicação na parte superior e, na parte inferior, o nome da aplicação, juntamente com o slogan e um botão de começar.

A imagem de fundo foi inserida através de um componente do React Native, que facilita a inserção de imagens de fundo para as telas. São passadas para esse componente algumas propriedades, como a fonte da imagem, o estilo e o modo de redimensionamento dessa imagem (o equivalente ao background-size do css), por exemplo.

Também foi aplicado um efeito de background linear por cima da imagem, para dar o efeito que vemos na imagem e, por fim, o conteúdo da página, que consiste no logo do app no topo da página, que foi inserido utilizando a biblioteca ‘react-native-svg’, que fornece componentes equivalentes às tags svg, path, circle etc., disponíveis no HTML padrão.

Os textos foram inseridos utilizando componentes da biblioteca Native Base. Para o nome do app, foi utilizado o componente Heading, que exibe o texto em negrito, para dar mais destaque. Já para o slogan, foi utilizado o componente Text, da mesma biblioteca, que é utilizado para exibição de textos, já que no React Native, diferentemente do React, não é possível inserir textos “soltos”, sem estarem envolvidos pelo componente Text.

O botão é um componente que emite um evento quando é pressionado. Esse botão em específico vai redirecionar o usuário para a tela de login do aplicativo.

Figura 1 – Landing page

Fonte: captura de tela de dispositivo móvel

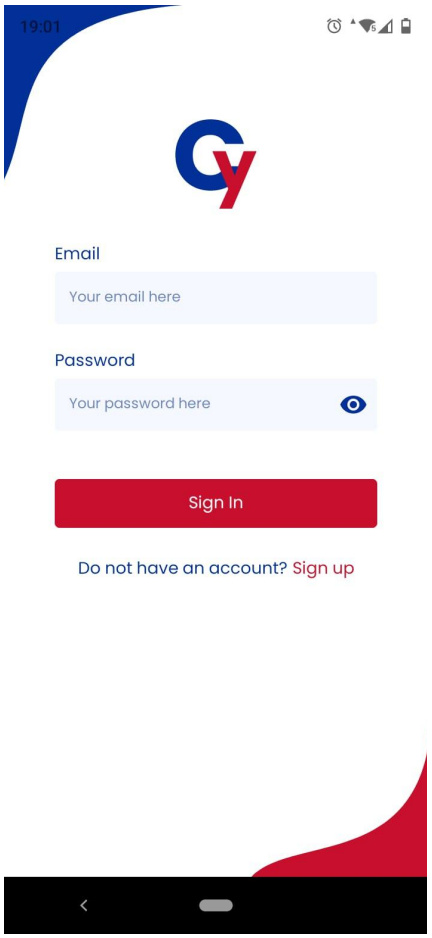
2.3.2 Tela de login

A tela de login (Figura 2) é composta por um formulário com dois campos, nos quais o usuário vai inserir seu e-mail cadastrado e sua senha, além do botão para submeter seus dados para verificação. Para esse e os demais formulários da aplicação, foi utilizada a biblioteca react-hook-form, que ajuda na performance da aplicação, pois evita re-renderizações desnecessárias e facilita a criação de formulários mais extensos.

Os Inputs são componentes do Native Base, que são personalizados pelo tema da aplicação e são renderizados pelo componente Controller, disponibilizado pela biblioteca react-hook-form.

Nessa tela também está disponível um link para o usuário se cadastrar, caso ainda não tenha uma conta. Assim que as credenciais do usuário são validadas, ele é redirecionado para a sua Home Page.

Figura 2 – Tela de login



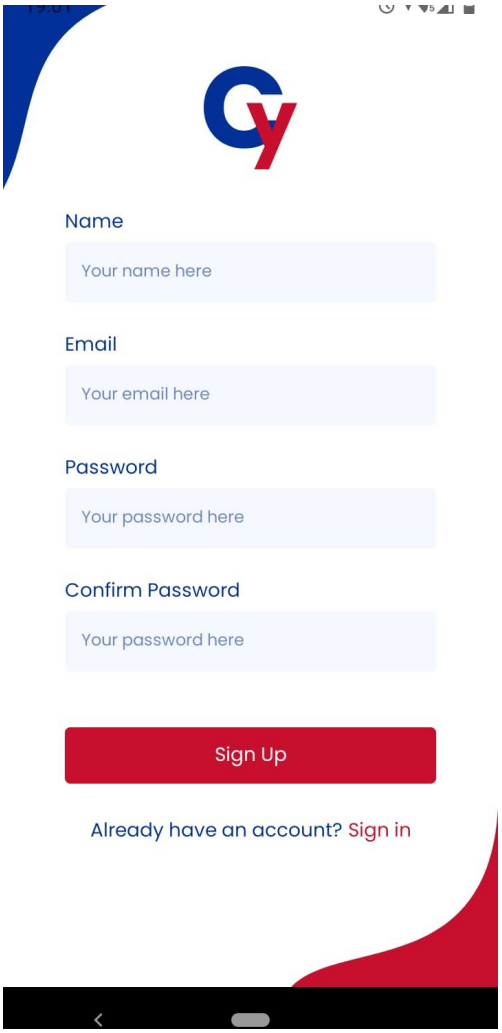
Fonte: captura de tela de dispositivo móvel

2.3.3 Tela de cadastro

Para utilizar o aplicativo, o usuário precisa criar uma conta, para que seus cards fiquem salvos. É na tela de cadastro (Figura 3) que isso é feito. Esta tela é bem similar à tela de login, que foi descrita anteriormente. Uma das poucas diferenças é que para o cadastro existe um campo extra, que é o de confirmação da senha. Esse campo é exigido para garantir que o usuário digitou corretamente a senha desejada. A exemplo da tela de login, existe um link na tela de cadastro para redirecionar o usuário para se autenticar caso ele já possua o registro.

Todos os formulários da aplicação foram validados utilizando a biblioteca yup, que traz vários métodos de validação, garantindo que as requisições sejam feitas conforme esperado.

Figura 3 – Tela de cadastro



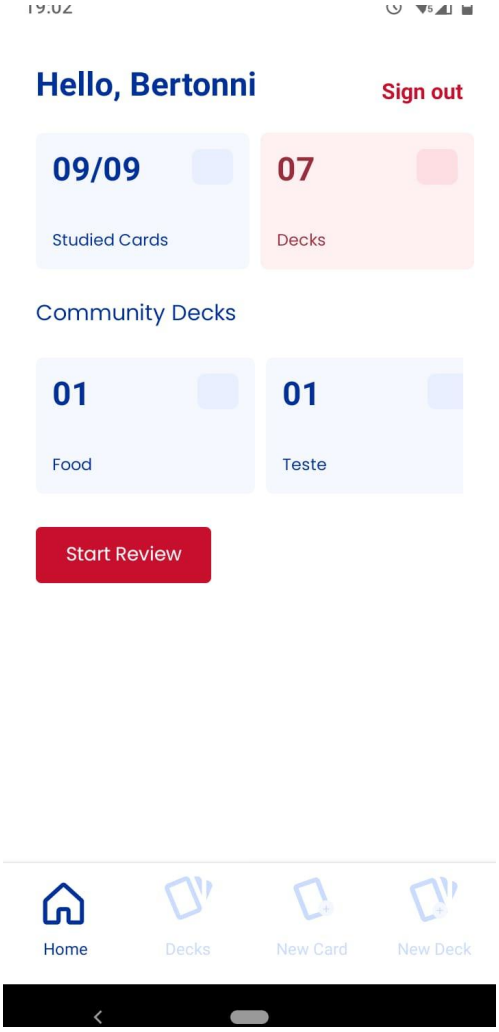
Fonte: captura de tela de dispositivo móvel

2.3.4 Home Page

Essa tela (Figura 4) exibe os cards que o usuário possui e os que ele já estudou, assim como seus próprios decks e os decks comunitários, que são decks criados por outros usuários e disponibilizados publicamente para que todos os usuários possam acessar. Também consta um botão para iniciar a revisão dos cards do usuário. Nessa tela, é introduzida a Bottom Tab (barra de navegação localizada na parte inferior da tela), que contém ícones que redirecionam o usuário para outras telas e ações.

A barra de navegação, assim como a navegação entre as telas, foi disponibilizada pela biblioteca “@react-navigation/native” e suas variantes, que são o sistema de navegação da aplicação.

Figura 4 - Home Page

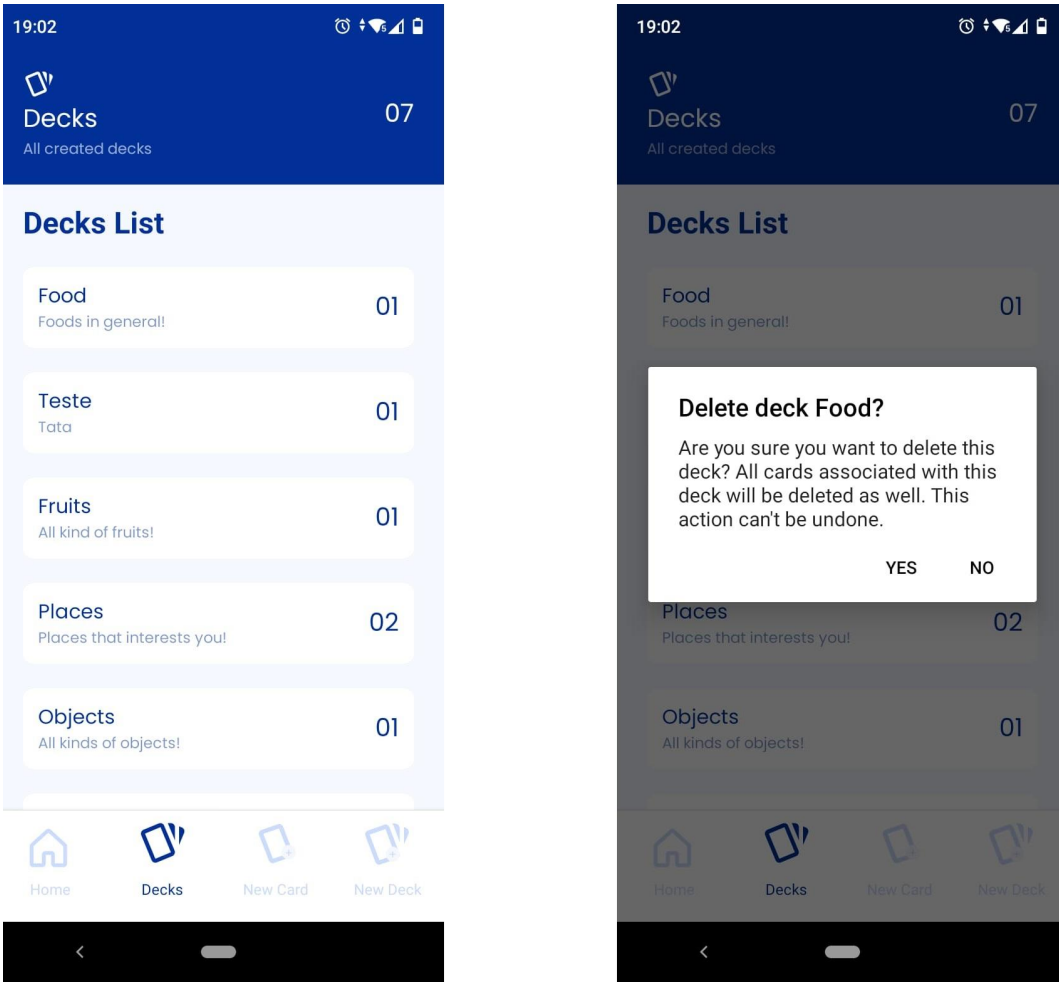


Fonte: captura de tela de dispositivo móvel

2.3.5 Tela de Decks

Aqui são exibidos os decks que o usuário possui. A tela (Figura 5) consiste de uma lista de decks pré-definidos, contendo o título do deck, uma descrição e quantos cards aquele deck possui. Se o usuário tocar e manter pressionado o card, será exibido um alerta de confirmação de remoção do card e, caso o usuário clique em “YES”, o card será removido, assim como todas as cartas atreladas a ele. Quando o usuário clicar em algum deck, serão exibidos os cards que aquele deck possui. Nessa tela, é introduzido um Header customizado, que exibe detalhes sobre o deck, o título do deck e um ícone de cartas acima do título.

Figura 5 - Tela de decks e remoção de deck

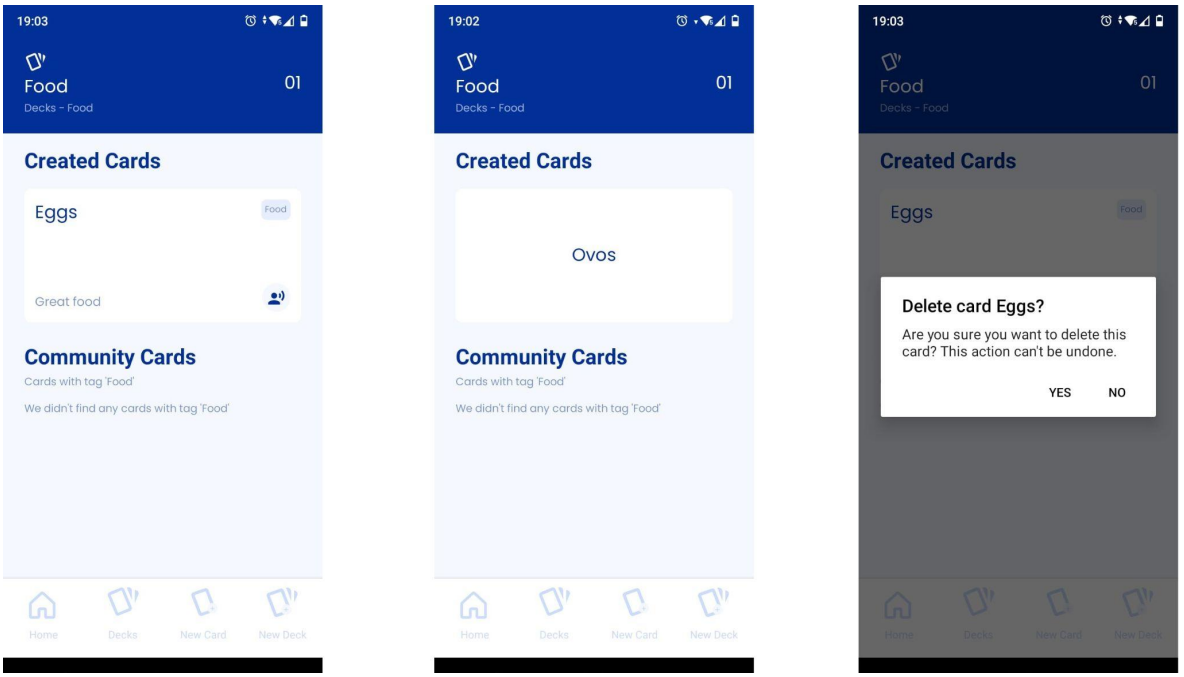


Fonte: captura de tela de dispositivo móvel

2.3.6 Tela de Cards

Essa tela (Figura 6) é acessada após o clique do usuário em qualquer deck de sua escolha. São exibidos os cards que o usuário criou com a tag referente àquele deck, além dos cards comunitários que possuem a mesma tag. Com um toque no card, será exibido o verso do card (Tela central da Figura 6), já ao manter o dedo pressionado no card, será executada uma ação que exibe um alerta perguntando se o usuário deseja remover o card (Tela da direita da Figura 6). Caso seja selecionada a opção “YES”, o card é removido e a interface é atualizada, caso contrário, nada acontece e o alerta é fechado.

Figura 6 – Tela de cards (Frente e Verso) e Remoção de Card

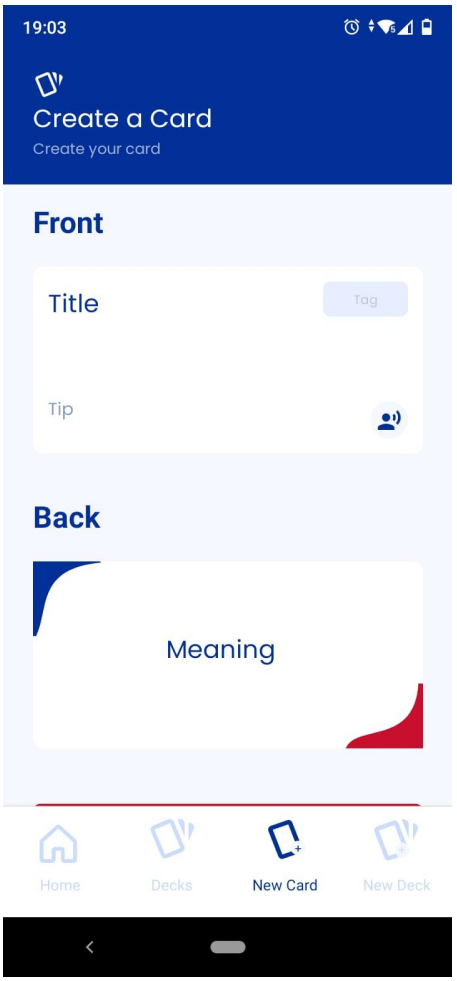


Fonte: captura de tela de dispositivo móvel

2.3.7 Tela de criação de card

Nessa tela (Figura 7) são exibidos dois cards, um representando a frente do card e o outro a parte de trás. Na parte da frente, o usuário deve clicar nos textos para inserir a palavra, dica e qual tag aquele card vai possuir. Já na parte de trás, será inserido o significado da palavra que deseja aprender. Após inserir essas informações, o usuário selecionará o deck em que vai salvar aquele card e vai clicar no botão “save” para concluir o processo.

Figura 7 – Tela de criação de card

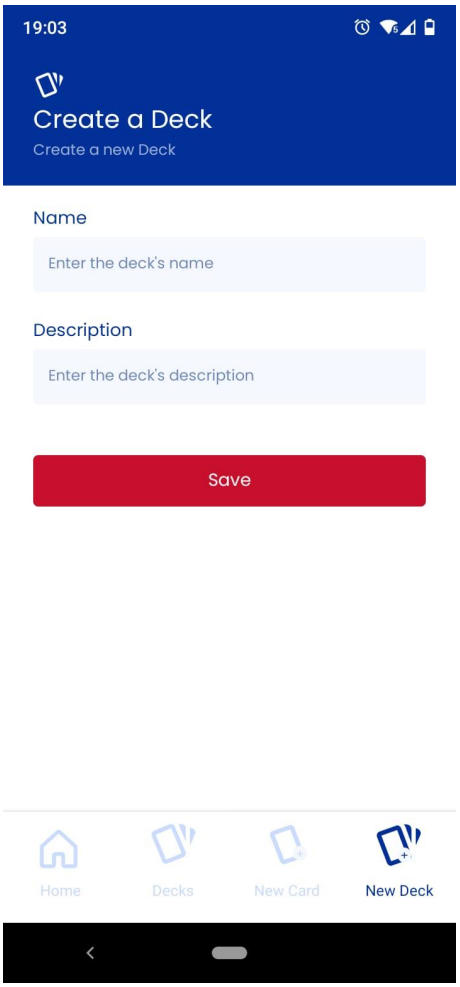


Fonte: captura de tela de dispositivo móvel

2.3.8 Tela de criação de deck

Esta tela (Figura 8) é bem simples, ela contém dois campos para inserção de texto, um para definir o nome do deck e o outro para a sua descrição. Ambos os campos são de preenchimento obrigatório e são validados no front-end, garantindo que os dados sejam salvos corretamente.

Figura 8 – Tela de criação de deck



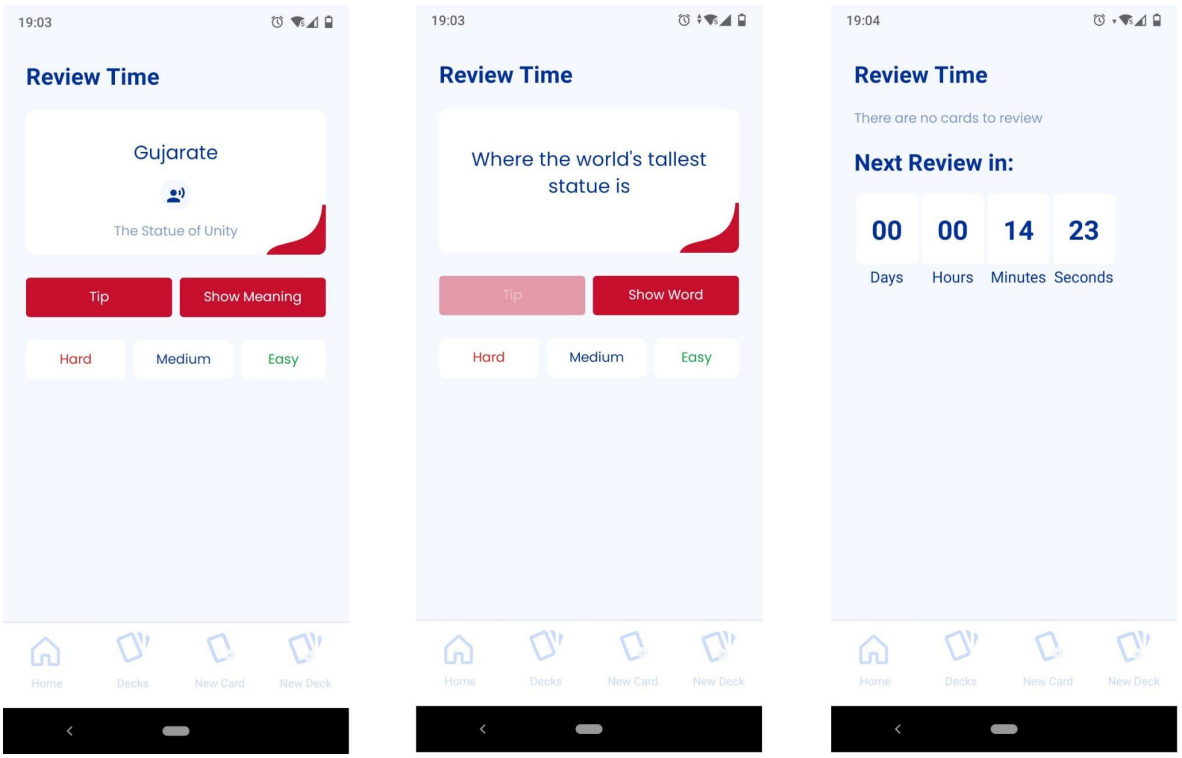
Fonte: captura de tela de dispositivo móvel

2.3.9 Tela de Review

Essa é a tela (Figura 9) em que os cards do usuário serão revisados. Aparecerão os cards criados, um a um, com a palavra que deseja aprender. Também constam na tela dois botões, um para exibir a dica cadastrada (Tela da esquerda da Figura 9) e outro para mostrar o significado da palavra (Tela central da Figura 9). Após revisar a palavra, o usuário vai atribuir uma dificuldade a ela. Essa dificuldade vai definir quando aquela carta ficará disponível novamente para revisão. Se o usuário achou a palavra difícil, por exemplo, aquele card ficará disponível para revisão novamente após 15 minutos, se ele escolheu a opção “medium”, 8 horas e, caso tenha achado fácil (opção “easy”), o card voltará para revisão após 3 dias. Quando não houver mais nenhum card a ser revisado, será exibido um contador

(Tela da direita da Figura 9), mostrando o tempo restante para que a próxima carta esteja disponível para revisão.

Figura 9 – Tela de review (Frente, verso e contador para próxima revisão)



Fonte: captura de tela de dispositivo móvel

2.4 Testes

Foram realizados testes de interface, nos quais foi verificado se as telas e funcionalidades estavam funcionando como esperado. Esses testes consistem do preenchimento dos campos do formulário com diversos tipos de entrada, para garantir que apenas o tipo de dado correto seja inserido em cada campo. Além dos testes de interface, foram realizados, também, testes unitários nos componentes e funções do códig, nos quais é verificado, assim como no teste de interface, se uma requisição foi realizada com sucesso e trouxe a resposta esperada ou não, por exemplo.

Todos os testes foram feitos pelos próprios desenvolvedores. Infelizmente, não conseguimos realizar testes com potenciais usuários do sistema ou pessoas que não fizeram parte do desenvolvimento, para termos uma noção melhor se a interface seria intuitiva o suficiente, para que o usuário não enfrentasse problemas durante o

uso da aplicação, ou para que fossem reportados possíveis erros e sugestões de melhorias.

3. Considerações Finais

Desenvolver um aplicativo foi um desafio, entre outros motivos, por ter sido o primeiro contato com o desenvolvimento mobile. A utilização do React Native facilitou um pouco o processo pelo fato de termos uma familiaridade com o React - biblioteca JavaScript na qual o React Native é baseado - mas ainda assim foi desafiador.

Futuramente, pretendemos cobrir os testes que não conseguimos durante o desenvolvimento, adicionar mais recursos à aplicação, como possibilitar a inserção de imagens em vez do texto utilizado para o significado da palavra, que não foi possível adicionar durante este processo. Também pretendemos disponibilizar o aplicativo para usuários do iOS, uma vez que a aplicação atualmente só funciona em dispositivos Android e disponibilizar a revisão por deck, já que hoje o usuário não escolhe os decks que quer revisar.

Esperamos que o aplicativo realmente ajude as pessoas em seu processo de aprendizagem da língua inglesa. Assim, teremos atingido o nosso propósito.

REFERÊNCIAS

2022 Developer Survey. **Stack Overflow**, 2022. Disponível em: https://survey.stackoverflow.co/2022/#most-popular-technologies-language. Acesso em 11 jan. 2023.

BIERMAN, Gavin; ABADI, Martín; TORGERSEN, Mads. **Understanding TypeScript**. In: **Jones, R. (eds) ECOOP 2014 – Object-Oriented Programming. ECOOP 2014**. Lecture Notes in Computer Science, vol 8586. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-44202-9_11

BORGES, Kleiton. **O aprimoramento da aquisição de vocabulário de alunos de língua inglesa através do uso de aplicativos**. *In:* **CONGRESSO LATINO-AMERICANO DE FORMAÇÃO DE PROFESSORES DE LÍNGUAS**, nº 6, 2016, Florianópolis. Resumos. p. 411-431.

FLANAGAN, David. **JavaScript: o guia definitivo**. Bookman Editora, 2004.

FRANCO, José et al. **Um estudo de caso sobre a necessidade do conhecimento da língua inglesa para os profissionais de desenvolvimento de software**. *In:* **Jornada Científica e Tecnológica da FATEC de Botucatu**, nº 5, 2016, Botucatu. Resumos. p. 1.

JAVASCRIPT. In: **MDN Web Docs**. 2022. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/JavaScript. Acesso em: 19/03/2023

MORATO, Rosinadja; BARBOZA, Lara. A IMPORTANCIA DO ENSINO DA LINGUA INGLESA: uma experiência com alunos do ensino médio. **Scientia Plena Jovem**, v. 7, n. 1, 2020.

The most spoken languages in the world. **Berlitz**, 2023. Disponível em: https://www.berlitz.com/blog/most-spoken-languages-world. Acesso em: 10 jan. 2023.

SOUTO, Mario. Front-end, Back-end e Full Stack. **Alura**, 2023. Disponível em: https://www.alura.com.br/artigos/o-que-e-front-end-e-back-end. Acesso em: 12 jan. 2023.

SOUZA, Pedro. **Um estudo sobre padrões e tecnologias para o desenvolvimento WEB - front-end**. 121. Projeto de Graduação - Curso de Engenharia Eletrônica e de Computação da Escola Politécnica - Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2016.