

Sistema de Flashcards Inteligentes

Nome dos Alunos:

Arthur Barbosa Dornelas Silva (24.00813-3)

Bruno Binaghi Carpi (24.00246-0)

Eduardo Aguiar Leite da Silva (24.00380-8)

Leonardo Formagio Bicudo (24.00257-7)

Lucas Novaes de Sá Ferreira (24.00721-8)

Vitor Eiken Tanahara (24.00684-0)

Sumário

1. Descrição/Resumo do Projeto	3
2. Tecnologias Empregadas	3
3. Definição de Papéis no Projeto	3
4. Levantamento dos Requisitos do Sistema de Software - Product Backlog	3
5.1. Planejamento da Sprint 1 - Sprint Backlog	4
5.2. Implementação e Testes - Sprint 1	4
5.3. Review da Sprint 1	4
5.4. Retrospectiva da Sprint 1	4
6. Sprint 2	4
7. Registro da Apresentação ao Parceiro	4
8. Referências	5
9. Apêndice I	5

1. Descrição/Resumo do Projeto

- O programa se baseia na criação de flashcards, os quais auxiliarão na organização e no aprendizado dos estudos do aluno, possibilitando que ele monte sua rotina de revisão sobre assuntos de sua escolha. Tendo isso em vista, o aplicativo contará com implementações de manuseabilidade e uma interface amigável.
- O problema identificado é a dificuldade dos alunos em absorver e aprender os conteúdos abordados em aula, seguindo o método de ensino tradicional brasileiro.
- O propósito do aplicativo é auxiliar no aprendizado e na organização dos estudos do aluno, priorizando os assuntos de determinadas matérias com as quais ele tenha dificuldades. Espera-se que os flashcards contribuam para a memorização dos conteúdos e, conseqüentemente, para o desempenho acadêmico do aluno.
- Público-alvo e stakeholders envolvidos.
 - O stakeholder será a própria instituição de ensino Poliedro, e o público-alvo será composto pelos próprios alunos do Poliedro.
- O que será implementado:
 - Flashcards estarão disponíveis para a visualização dos alunos;
 - Níveis de dificuldade;
 - Informar se a resposta está certa ou errada;
 - O aluno pode selecionar as matérias que quer estudar;
 - Modo de perguntas aleatórias (não prioriza matérias específicas);
 - Modo por dificuldade (o aluno seleciona apenas as perguntas de uma certa dificuldade); (a confirmar)
 - Consultar respostas erradas;
 - Registrar quantos flashcards foram respondidos (certos e errados);
 - Interface intuitiva (botões interativos);
 - Sistema que priorize os flashcards que o aluno errou, do que os que eles acertaram várias vezes em seguida;
 - Modo desafio;
 - Aprimoramento da Interface;
 - Sistema de cadastro e login do usuário;
- O que não será implementado:
 - Ferramentas de segurança da informação

- o Estilização dos flashcards (ex.: mudança de cor, formato dos flashcards, brilho e efeitos, etc)

2. Tecnologias Empregadas

- As linguagens de programação utilizadas e suas justificativas para este projeto são:
 - o HTML e CSS, como linguagens de marcação para o desenvolvimento front-end.
 - o Java, como linguagem para o desenvolvimento back-end e conexão com o banco de dados.
 - o SQL, como linguagem para o CRUD.
 - o Javascript, como linguagem para o desenvolvimento de scripts front-end.
- Os frameworks e bibliotecas utilizadas durante o projeto são:
 - o React-native, para auxiliar o desenvolvimento front-end e multiplataforma.
 - o JDBC, para a comunicação entre programa e banco de dados.
- O banco de dados do projeto será local na máquina do usuário e para isso será utilizado o SQLite.
Ferramentas auxiliares (versionamento, colaboração, gerenciamento de tarefas).
 - o Git e Github
 - o Figma
 - o Trello

3. Definição de Papéis no Projeto

- Estrutura do time de acordo com o framework SCRUM.
- Nome e função de cada integrante:
 - o Product Owner (PO): Bruno Binarghi Carpi
 - o Scrum Master: Leonardo Formagio Bicudo
 - o Desenvolvedores: Arthur Barbosa Dornelas Silva, Eduardo Aguiar Leite da Silva, Lucas Novaes de Sá Ferreira e Vitor Eiken Tanahara

4. Levantamento dos Requisitos do Sistema de Software – Product Backlog

- Metodologia para levantamento de requisitos.
 - o Entrevistas
 - o Brainstorming
 - o Formulário/ Lista de Perguntas
- Descrição detalhada do Product Backlog:
 - Cartões de histórias representando requisitos.

N° do Cartão	Cartão de História	Prioridade
1	<p>Como cliente</p> <p>Quero que o sistema tenha cartões de diferentes níveis de dificuldade</p> <p>Para que os usuários possam treinar o conteúdo ensinado com perguntas mais desafiadoras.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • Criar uma interface para que o usuário possa selecionar este modo; • Este modo deve estar na mesma seção junto aos outros modos; • Ao selecionar este modo haverá duas opções, um para iniciar a sessão e outro que irá exibir as estatísticas do usuário; • A interface de estatísticas só ficará disponível para seleção após o usuário completar ao menos uma sessão, caso nenhuma sessão tenha sido realizada este botão estará bloqueado (terá um valor nulo); • Ao selecionar "exibir estatísticas", será exibido um gráfico, de pizza, que irá informar quais questões o usuário acertou ou errou da dificuldade selecionada; • Na tela de estatísticas, haverá uma interface que irá exibir os cards que o usuário errou, onde estes cards estarão destacados em vermelho, para o usuário saber quais foram os seus erros; • O usuário terá três opções de dificuldade que ele pode selecionar para uma pergunta, sendo elas: fácil, médio e difícil; • As perguntas deste modo serão organizadas com base na seleção do usuário ao 	<p>3</p> <p>Esforço: 3</p>

	<p>responder perguntas do modo normal, ou seja, ao selecionar perguntas de nível fácil, o sistema irá selecionar todos os cards marcados como fáceis pelo usuário, o mesmo será feito com as outras dificuldades.</p>	
2	<p>Como usuário Quero fazer login no aplicativo Para que eu possa ter acesso aos meus flashcards e estatísticas.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • Na página inicial deve ter opções de login/cadastro para alunos e professores; • O acesso dos professores deve ser feito pelo domínio "...@sistemapoliedro.com.br"; • O acesso dos alunos deve ser feito pelo domínio "...@p4ed"; • Deverá haver um layout que mostre se um email já cadastrado tente se cadastrar novamente; • Deverá haver um layout indicando que o login de aluno/professor foi bem sucedido, onde o professor será direcionado para uma página alternativa com mais funções do que a página dos alunos; • No banco de dados essas operações devem ser feitas utilizando um CRUD. 	<p>1</p> <p>Esforço: 5</p>
3	<p>Como cliente Quero que o usuário possa criar os cards das matérias que ele quiser estudar Para que o usuário possa ter controle sobre seu conteúdo e organizar melhor seus estudos.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • Devemos criar a interface do 	<p>2</p> <p>Esforço: 5</p>

	<p>nosso sistema de maneira intuitiva para o usuário;</p> <ul style="list-style-type: none"> • Os usuários devem ser capazes de criar suas próprias matérias e criarem seus cards com base nelas; • Devemos criar um botão para que o usuário possa criar um novo "deck" de cards; • Devemos deixar exemplos de como criar um flashcard e de como deve ser sua estrutura para o aluno seguir essa base, mantendo a padronização do sistema; • Os alunos podem adicionar quantos conteúdos e cards eles quiserem para determinada pasta de alguma matéria; • Os alunos devem ser capazes de compartilhar seus "baralhos de cards" criados com outros alunos. 	
4 (Feature)	<p>Como equipe de desenvolvimento</p> <p>Precisamos criar uma conexão com o banco de dados utilizando o SQL</p> <p>Para que os usuários possam ter todo tipo de informação que necessitam, armazenada.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • Devemos criar uma tabela que seja capaz de armazenar as perguntas criadas; • Devemos criar uma tabela que seja capaz de armazenar os dados de cadastro/login de alunos e de professores; • O sistema deve ser capaz de utilizar os dados armazenados na tabela de perguntas; • O sistema deve ser capaz de utilizar os dados armazenados na tabela de cadastro/login de alunos e de professores; • O sistema de banco de dados deve possuir a capacidade de criptografar as informações inseridas para garantir o 	<p>1</p> <p>Esforço: 8</p>

	mínimo de segurança em relação aos dados registrados.	
5	<p>Como equipe de desenvolvimento Precisamos criar um modo padrão para nosso sistema que deverá gerar os cards em ordem aleatória Para que o usuário possa responder todos os cards que ele necessita para aquela determinada sessão, sem que haja uma ordem específica e uma sequência padronizada dos cards.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • O aluno deve ser capaz de marcar quais perguntas ele acertou ou errou; • Os cards marcados como certos ou errados devem ser registrados pela tabela de perguntas, com uma indicação se a resposta estiver correta ou incorreta; • A tabela de perguntas deve registrar quais perguntas foram marcadas como corretas; • O sistema deve permitir que as informações alteradas (como quantidade de acertos ou erros) possam ser salvas manualmente a qualquer momento, ou quando o aluno responder todos os flashcards que correspondem ao modo de estudo escolhido por ele. 	<p>1</p> <p>Esforço: 3</p>
6	<p>Como equipe de desenvolvimento Precisamos criar sessões para o usuário responder as perguntas Para que ele possa encerrar seu período de resposta após realizar um certo número de perguntas.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • A sessão deve ser iniciada assim que o usuário começar a responder as perguntas e encerrada quando ele terminar de respondê-las; 	<p>2</p> <p>Esforço: 2</p>

	<ul style="list-style-type: none"> • Deve haver uma opção para que o usuário possa encerrar a sessão após responder qualquer número de perguntas; • O sistema deve informar ao usuário que ele encerrou a sessão através de uma mensagem de finalização; • O sistema deve registrar e informar ao usuário quantos cards ele respondeu naquela sessão através da mensagem de finalização; • O sistema deve informar quantos cards ele acertou e errou naquela sessão na mesma mensagem de finalização. 	
7	<p>Como equipe de desenvolvimento</p> <p>Precisamos que o sistema tenha um modo de desafio com perguntas mais desafiadoras</p> <p>Para que o usuário possa se desafiar e testar seus conhecimentos em determinadas matérias.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • Criar uma interface para que o usuário possa selecionar este modo; • Este modo deve estar na mesma seção junto aos outros modos; • Ao selecionar este modo haverá duas opções, um para iniciar a sessão e outro que irá exibir as estatísticas do usuário; • A interface de estatísticas só ficará disponível para seleção após o usuário completar ao menos uma sessão, caso nenhuma sessão tenha sido realizada este botão estará bloqueado (terá um valor nulo); • Ao selecionar "exibir estatísticas", será exibido um gráfico, de pizza, que irá informar quais questões o usuário acertou ou errou da dificuldade selecionada; • Na tela de estatísticas, 	<p>2</p> <p>Esforço: 3</p>

	<p>haverá uma interface que irá exibir os cards que o usuário errou, onde estes cards estarão destacados em vermelho, para o usuário saber quais foram os seus erros;</p> <ul style="list-style-type: none"> • Neste modo, o usuário irá responder todos os cards que foram criados por ele, independente da matéria, onde esses cards estarão embaralhados e o usuário deverá marcar quais questões ele acertou e quais ele errou. 	
8 (Feature)	<p>Como equipe de desenvolvimento Precisamos criar um sistema de Flashcards funcional e intuitivo Para que possamos implementar os requisitos e funções neste sistema principal.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • Devemos criar uma base para o nosso sistema a partir das linguagens de programação: HTML, CSS, Java, Javascript e SQL para o nosso banco de dados; • O sistema deve ser intuitivo para vários usuários, com funções facilitadoras e botões para fácil entendimento; • O nosso sistema deve se adequar a todos os requisitos, sendo sua principal função, facilitar o aprendizado dos alunos e auxiliar na organização dos estudos. 	<p>1</p> <p>Esforço: 6</p>
9	<p>Como equipe de desenvolvimento Precisamos criar uma tela que exibe as estatísticas do usuário após uma sessão Para que o usuário possa receber um feedback do seu desempenho naquela sessão.</p>	<p>3</p> <p>Esforço: 5</p>

- Priorização dos requisitos.
- Estimativa de esforço.
- Sprint de implementação.
- Azure:

Projeto-Integrador-Flashcar... / Projeto-Integrador-Flashcar... / Boards / Backlogs

Search

Thank you for trying the new boards hub preview. If you experience any issues, please [report the bug](#). If your issue is blocking, you can disable the preview by following [these steps](#).

Projeto-Integrador-Flashcards Team + ☆ ⚙

New Work Item **View as Board** **Column Options**

Backlog Analytics **Backlog items**

Order	Work Item Type	Title	State	Effort	Value Area	Iteration Path
1	Product Backlo...	📄 Criar níveis de dificuldade para cards	● New		Architectural	Projeto-Integrador-Flashcards
2	Product Backlo...	📄 Login/Cadastro de usuários para acesso ao sistema	● New		Architectural	Projeto-Integrador-Flashcards
3	Product Backlo...	📄 Criação de Cards para categorias de matérias	● New		Architectural	Projeto-Integrador-Flashcards
4	Product Backlo...	📄 Criação do Banco de Dados	● New		Architectural	Projeto-Integrador-Flashcards
5	Product Backlo...	📄 Registro de acertos e erros	● New		Architectural	Projeto-Integrador-Flashcards
6	Product Backlo...	📄 Registro da pontuação	● New		Architectural	Projeto-Integrador-Flashcards
7	Product Backlo...	📄 Registro de dificuldade dos cards	● New		Architectural	Projeto-Integrador-Flashcards
8	Product Backlo...	📄 Criar sessões para responder as perguntas	● New		Architectural	Projeto-Integrador-Flashcards
9	Product Backlo...	📄 Criação do sistema de Flashcards	● New		Architectural	Projeto-Integrador-Flashcards

Planning
Drag and drop work items to include them in a sprint.

Projeto-Integrador-Flashcards Team

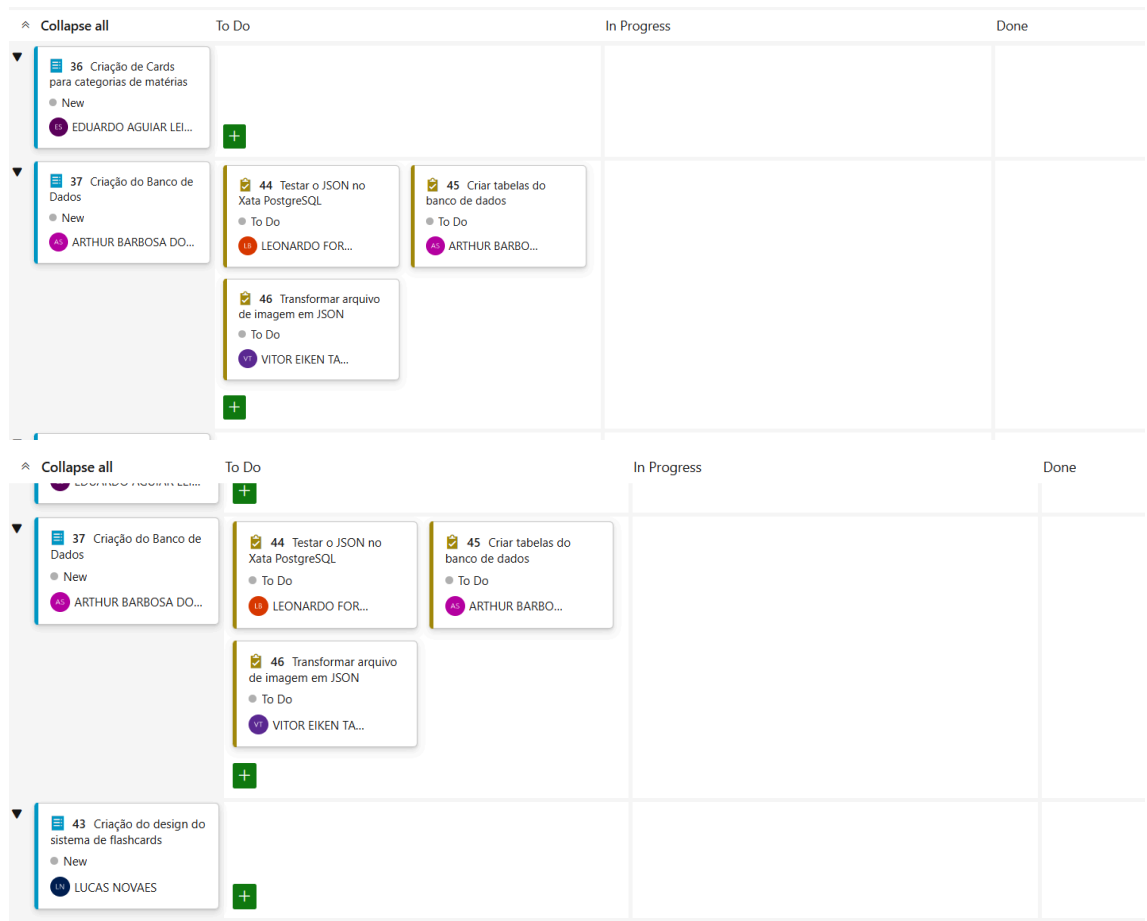
Sprint 1
No work scheduled yet

Sprint 2
No work scheduled yet

Sprint 3
No work scheduled yet

5.1. Planejamento da Sprint 1 - Sprint Backlog

- Ferramenta utilizada para gerenciamento: Azure.



5.2. Implementação e Testes - Sprint 1

- Link do repositório contendo a implementação.

<https://github.com/Memoria-de-Elefante/flashcard.git>

- Algoritmos ou estruturas de dados obrigatórias implementadas.

Foi utilizada a estrutura do Hash-Map.

- Abordagem de testes adotada (BDD e TDD).

Fizemos em BDD (Behavior-Driven Development).

- Testes de BDD documentados para os principais cartões de história.

Feature: Cadastro e login de usuários

Como usuário

Quero fazer login no aplicativo

Para que eu possa ter acesso aos meus flashcards e estatísticas.

Cenário: Cadastro bem-sucedido de professor

Dado que não existe um usuário com o e-mail "prof@sistemapoliedro.com.br"

Quando eu tento me cadastrar como "Carlos" com o e-mail "prof@sistemapoliedro.com.br" e senha "senha123"

Então o sistema deve exibir a mensagem "Usuário cadastrado com sucesso."

Cenário: Cadastro bem-sucedido de aluno

Dado que não existe um usuário com o e-mail "aluno@p4ed"

Quando eu tento me cadastrar como "Ana" com o e-mail "aluno@p4ed" e senha "senha456"

Então o sistema deve exibir a mensagem "Usuário cadastrado com sucesso."

Cenário: Cadastro com e-mail duplicado

Dado que já existe um usuário com o e-mail "prof@sistemapoliedro.com.br"

Quando eu tento me cadastrar com esse mesmo e-mail

Então o sistema deve exibir a mensagem "E-mail já cadastrado."

Cenário: Cadastro com domínio inválido

Dado que não existe um usuário com o e-mail "joao@gmail.com"

Quando eu tento me cadastrar como "João" com o e-mail "joao@gmail.com" e senha "123"

Então o sistema deve exibir a mensagem "Domínio de e-mail inválido para cadastro."

Cenário: Login bem-sucedido como aluno

Dado que existe um aluno cadastrado com o e-mail "aluno@p4ed" e senha "senha456"

Quando eu tento fazer login com "aluno@p4ed" e "senha456"

Então o sistema deve exibir a mensagem "Login bem-sucedido como aluno."

Cenário: Login bem-sucedido como professor

Dado que existe um professor cadastrado com o e-mail "prof@sistemapoliedro.com.br" e senha "senha123"

Quando eu tento fazer login com "prof@sistemapoliedro.com.br" e "senha123"

Então o sistema deve exibir a mensagem "Login bem-sucedido como professor."

Cenário: Login com senha incorreta

Dado que existe um aluno com o e-mail "aluno@p4ed" e senha "senha456"

Quando eu tento fazer login com "aluno@p4ed" e "senhaErrada"

Então o sistema deve exibir a mensagem "Senha incorreta."

Cenário: Login com usuário inexistente

Dado que não existe um usuário com o e-mail "desconhecido@p4ed"

Quando eu tento fazer login com "desconhecido@p4ed" e "qualquerSenha"

Então o sistema deve exibir a mensagem "Usuário não encontrado."

Feature: Criação e gerenciamento de flashcards

Como equipe de desenvolvimento

Precisamos criar um sistema de Flashcards funcional e intuitivo

Para que possamos implementar os requisitos e funções neste sistema principal.

Cenário: Criação de uma nova matéria

Dado que estou logado no sistema

Quando eu informo o nome da nova matéria "Física"

E cliço no botão "Criar Matéria"

Então o sistema deve exibir a matéria "Física" na lista de categorias

Cenário: Criação de um novo deck para uma matéria

Dado que já existe uma matéria chamada "Matemática"

Quando eu cliço no botão "Criar novo deck"

E dou o nome "Trigonometria"

Então o sistema deve criar o deck "Trigonometria" dentro da matéria "Matemática"

Cenário: Visualização de exemplo de flashcard

Dado que estou na tela de criação de flashcard

Então o sistema deve exibir um exemplo de estrutura com frente e verso preenchidos

Cenário: Criação de flashcards dentro de um deck

Dado que o deck "Trigonometria" já existe

Quando eu adiciono um flashcard com frente "O que é seno?" e verso "Razão entre o cateto oposto e a hipotenusa"

Então o sistema deve salvar o flashcard no deck "Trigonometria"

Cenário: Adição de múltiplos flashcards em um deck

Dado que estou no deck "História - Brasil Colonial"

Quando eu adiciono 3 flashcards diferentes

Então o sistema deve exibir os 3 flashcards salvos corretamente dentro do deck

Cenário: Compartilhamento de deck com outro usuário

Dado que eu criei o deck "Geografia - Climas"

Quando eu cliço em "Compartilhar deck"

E informar o e-mail "colega@p4ed"

Então o sistema deve confirmar com a mensagem "Deck compartilhado com sucesso com colega@p4ed"

Cenário: Interface intuitiva para criação de conteúdo
Dado que sou um novo usuário acessando o sistema
Então a tela deve conter botões claros para criar matéria, criar deck e adicionar flashcard

- Evidências de testes unitários e de integração (prints ou links).

TelaLogin.tsx

```
import React, { useState } from 'react';
import { Text, Image, SafeAreaView, StyleSheet, Dimensions,
TextInput } from "react-native";
import CustomButton from "../components/CustomButton";
import { Link, router } from 'expo-router';
import SenhaButton from "../components/SenhaButton";

const { width } = Dimensions.get('window');

export default function TelaLogin({ }) {
  const [email, setEmail] = useState('');
  const [senha, setSenha] = useState('');

  return (
    <SafeAreaView style={styles.container}>
      <Image
source={require('../assets/images/Poliedro.png')}
style={styles.image}
/>
      <Text style={styles.text}>LOGIN</Text>

      <Text style={styles.label}>E-mail</Text>
      <TextInput
style={styles.input}
placeholder="Digite seu e-mail aqui"
placeholderTextColor="#7C7C7C"
value={email}
onChangeText={setEmail}
/>

      <Text style={styles.label}>Senha</Text>
      <SenhaButton
title="Senha"
value={senha}
onChangeText={setSenha}
/>
    </SafeAreaView>
  );
}
```



```

        onPress={() => alert("Altera a visibilidade
da senha"))}
    />

    <Link href="/TelaCadastro"> {/* Alterar a rota
*/}

        <Text style={styles.link}>
            Esqueci a Senha
        </Text>
    </Link>

    <CustomButton
        title="Login"
        width={200}
        height={60}
        borderRadius={5}
        onPress={() => alert("Fazer Login")}
    />

    <Link href="/TelaCadastro">
        <Text style={styles.link}>
            Não tem uma conta? Cadastre-se
        </Text>
    </Link>

    <CustomButton
        title="Após Login"
        marginVertical={20}
        onPress={() => router.push('/TelaInicial')}
    />

    </SafeAreaView>
);
}

const styles = StyleSheet.create({
    container: {
        flex: 1,
        backgroundColor: "#000000",
        justifyContent: "flex-start",
        alignItems: "center",
        paddingTop: 10,
        paddingHorizontal: 0,
        paddingBottom: 0,
        flexDirection: 'column',
    },
    image: {
        width: width * 0.6,
        height: width * 0.5,
        resizeMode: 'contain',
        marginBottom: 0,
    }
});

```

```

    },
    text: {
      color: "#FFFFFF",
      fontSize: width * 0.1,
      fontWeight: "bold",
      textAlign: "center",
      marginTop: width * -0.05,
    },
    input: {
      backgroundColor: '#eee',
      borderRadius: 25,
      paddingHorizontal: width * 0.05,
      color: '#000',
      width: width * 0.7,
      height: width * 0.13,
      fontSize: 16,
    },
    label: {
      color: '#fff',
      fontWeight: "bold",
      fontSize: 18,
      marginTop: 10,
      padding: 10,
      alignSelf: 'flex-start',
      paddingLeft: width * 0.15,
    },
    link: {
      color: '#AAAAAA',
      textDecorationLine: 'underline',
      fontSize: width * 0.04,
    },
  },
});

```

TelaSelecaoCards.tsx

```

import React, { useState } from 'react';
import { TouchableOpacity, Text, Image, SafeAreaView, View,
StyleSheet, Dimensions, TextInput } from "react-native";
import CustomButton from "../components/CustomButton";
import Flashcard from "../components/Flashcard";
import { useRouter } from "expo-router";
import PagerView from 'react-native-pager-view';

const { width } = Dimensions.get('window');
const router = useRouter();

export default function TelaSelecaoCards({ }) {
  const [text, setText] = useState('Matéria');
  const [isEditing, setIsEditing] = useState(false);

```

```

return (
  <SafeAreaView style={styles.container}>
    <SafeAreaView style={styles.topRightIcon}>
      <Image
source={require('../../assets/images/IconCompartilhar.png')}
      > style={styles.iconCompartilhar} />
      { /* Configurar link para compartilhar ao
ser clicado */ }
    </SafeAreaView>

    <SafeAreaView style={styles.iconGroup}>
      {isEditing ? (
        <TextInput
          style={styles.input}
          value={text}
          onChangeText={setText}
          onBlur={() => setIsEditing(false)}
// Fecha o input ao perder o foco
          autoFocus
        />
      ) : (
        <Text style={styles.text}>{text}</Text>
      )}

      <TouchableOpacity style={{ marginLeft: 10
}} onPress={() => setIsEditing(true)}>
        <Image
source={require('../../assets/images/IconEditar.png')}
        > style={styles.image} />
      </TouchableOpacity>
    </SafeAreaView>

    <PagerView style={styles.pagerView}
initialPage={0}>
      {[1, 2, 3, 4].map((num) => (
        <View style={styles.page}
key={num.toString()}>
          <Flashcard
            frontText="Pergunta Aleatória"
            backText="Resposta"
            width={width * 0.8}
            height={400}
            borderRadius={10}
            editable={false}
            onPress={() =>
router.push('/TelaEdu')}
          />
        </View>
      ) )}
    </PagerView>

```

```

        <CustomButton
            title="+"
            onPress={() => alert("Criação de deck")}
            borderRadius={5}
            marginTop={50}
            textStyle={{ fontSize: 30 }}
        />
    </SafeAreaView>
  );
}

```

```

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "#000000",
    justifyContent: "flex-start",
    alignItems: "center",
    paddingTop: 10,
    paddingHorizontal: 0,
    paddingBottom: 0,
    flexDirection: 'column',
  },
  iconGroup: {
    flexDirection: 'row',
    alignItems: 'center',
    gap: 5,
  },
  text: {
    color: "#FFFFFF",
    fontSize: width * 0.08,
    fontWeight: "bold",
    textAlign: "center",
    marginVertical: width * 0.04,
    marginRight: 10,
  },
  image: {
    width: width * 0.09,
    height: width * 0.1,
    resizeMode: 'contain',
    marginBottom: 0,
  },
  iconCompartilhar: {
    width: width * 0.09,
    height: width * 0.1,
    resizeMode: 'contain',
  },
  topRightIcon: {
    width: '85%',
    alignItems: 'flex-end',
    paddingRight: 20,
  },

```

```

        marginTop: 30,
    },
    page: {
        justifyContent: 'center',
        alignItems: 'center',
        paddingHorizontal: width * 0.05,
    },
    pagerView: {
        flex: 1,
        width: '100%',
    },
    input: {
        height: 40,
        margin: 12,
        borderWidth: 1,
        padding: 5,
        fontWeight: "bold",
        color: 'FFFFFF',
        fontSize: width * 0.08,
        borderColor: 'FFFFFF'
    }
});

```

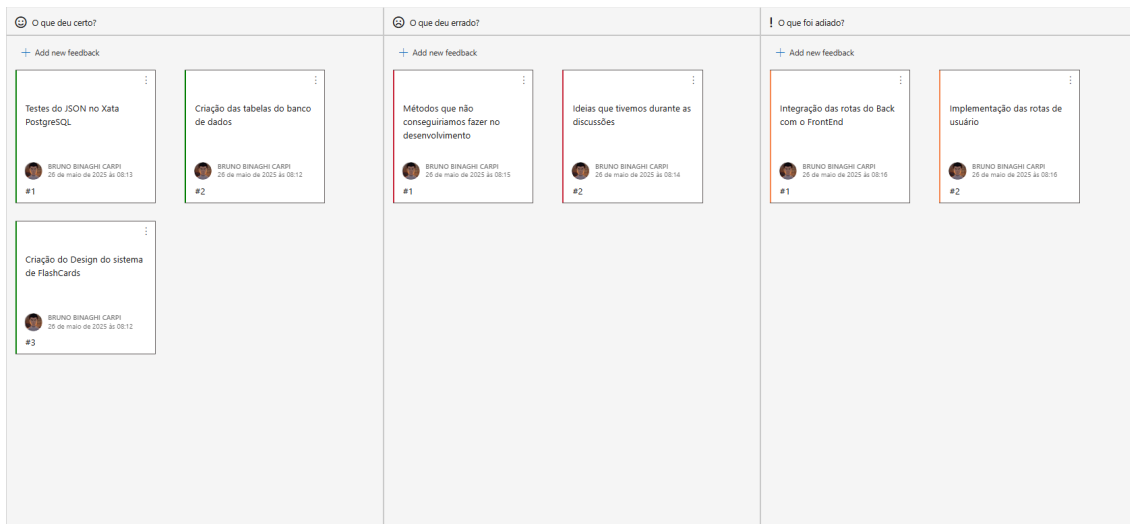
5.3. Review da Sprint 1

- Relato dos resultados obtidos na Sprint.
- O objetivo foi cumprido?
- Planejamento para a próxima Sprint.
- Feedback do parceiro testando essa Sprint.

Tivemos um ótimo resultado, nosso parceiro se agradou com as funcionalidades iniciais que implementamos na Sprint 1, o design principal foi aprovado, assim como os requisitos que iremos implementar. Agora para a próxima Sprint, iremos dar prioridade às funcionalidades do nosso sistema, como a criação das rotas de usuário, criação dos diferentes modos de jogo e integração das rotas do Back com o FrontEnd.

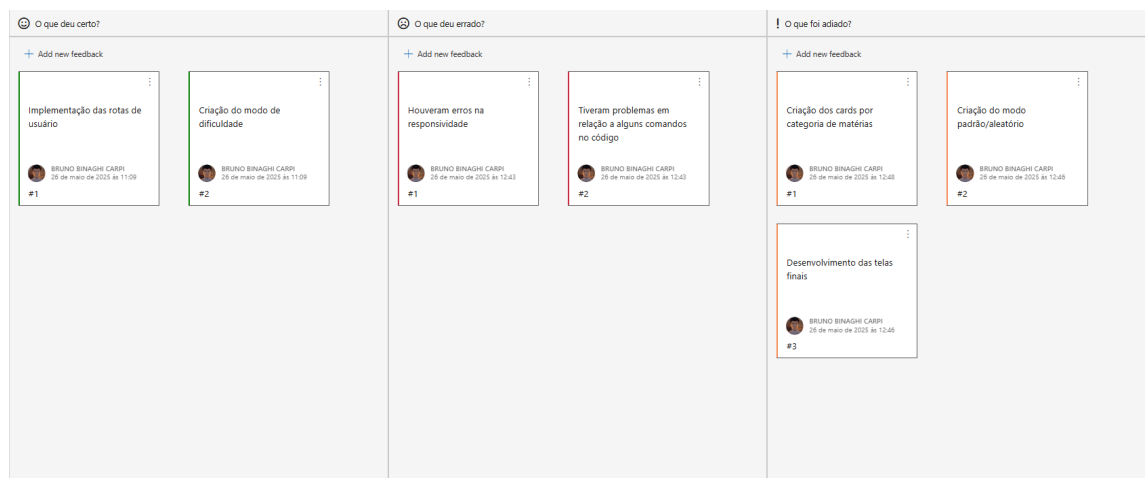
5.4. Retrospectiva da Sprint 1

- Avaliação do desempenho do time.
- Pontos fortes e fracos da Sprint.
- O que manter e o que melhorar para a próxima Sprint.



6. Sprint 2

- Repetição da estrutura da Sprint 1.
- Ajustes realizados com base no feedback da Sprint anterior.]



Após nossa primeira Sprint, fizemos uma divisão maior das tarefas, focando mais na etapa de desenvolvimento, já que o design das telas já foi concluído, já começamos a realizar os teste de usuário no banco de dado e eles foram bem-sucedidos, depois dividimos a equipe de Front e BackEnd, onde o Back irá se distribuir para realizar as rotas necessárias e o Front irá se dividir para cada um fazer um determinado número de telas, para que assim não

haja ninguém sobrecarregado e possamos concluir o projeto da forma mais rápida e eficiente possível.

- Testes de BDD documentados para os principais cartões de história.

Funcionalidade: Modo de Dificuldade

Como usuário

Quero estudar flashcards com diferentes níveis de dificuldade

Para que eu possa melhorar meu aprendizado praticando com perguntas em níveis variados

Cenário: Selecionar modo de dificuldade

Dado que o usuário está na tela de modos de estudo

Quando o usuário seleciona "Modo de Dificuldade"

Então o sistema deve exibir as opções "Iniciar Sessão" e "Exibir Estatísticas"

Cenário: Exibir estatísticas desabilitado sem sessões realizadas

Dado que o usuário nunca concluiu uma sessão de dificuldade

Quando o usuário acessa o modo de dificuldade

Então o botão "Exibir Estatísticas" deve estar desabilitado

Cenário: Exibir estatísticas após sessão concluída

Dado que o usuário já concluiu pelo menos uma sessão de dificuldade

Quando o usuário acessa o modo de dificuldade

Então o botão "Exibir Estatísticas" deve estar habilitado

E um gráfico de pizza deve mostrar os acertos e erros por dificuldade

E os cards respondidos incorretamente devem estar destacados em vermelho

Cenário: Classificar perguntas por dificuldade

Dado que o usuário está respondendo perguntas no modo normal

Quando o usuário marca uma pergunta como "difícil"

Então o sistema deve associar esse card à dificuldade "difícil"

Cenário: Sessão com cards de uma dificuldade específica

Dado que o usuário selecionou a dificuldade "fácil" no modo de dificuldade

Quando a sessão for iniciada

Então o sistema deve exibir apenas os cards marcados como "fácil"

- Evidências de testes unitários e de integração (prints ou links).

TelaModoDesafio.tsx

```

import React from "react";
import { Image, SafeAreaView, StyleSheet, Dimensions } from
"react-native";
import CustomButton from "../components/CustomButton";
import { router } from "expo-router";

const { width } = Dimensions.get('window');

export default function TelaModoDesafio() {
  return (
    <SafeAreaView style={styles.container}>
      <Image
source={require('../assets/images/Poliedro.png')}
style={styles.image}
/>
      <CustomButton
title="Iniciar Sessão"
width={250}
borderRadius={10}
marginVertical={60}
onPress={() => router.push('./FlashcardDesafio')}
/>
      <CustomButton
title="Ver Estatísticas"
width={250}
borderRadius={10}
marginVertical={60}
onPress={() => console.log("Indo para
Estatísticas")}
/>
    </SafeAreaView>
    // width: width ?? screenWidth * 0.45,
    // height: height ?? screenWidth * 0.15,
    // borderRadius: borderRadius ?? 15,
    // marginVertical: marginVertical ?? 0.045,
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "#000000",
    justifyContent: "flex-start",
    alignItems: "center",
    paddingTop: 10,
    paddingHorizontal: 0,
    paddingBottom: 0,
    flexDirection: 'column',
  },
},

```



```
image: {  
  width: width * 0.6,  
  height: width * 0.5,  
  resizeMode: 'contain',  
  marginBottom: 0,  
  marginTop: width * 0.07,  
},  
));
```

7. Registro da Apresentação ao Parceiro

- Data e local da apresentação.
Participantes.
- Feedback recebido.
- Ajustes sugeridos.

Oi Bruno, boa tarde.

A ideia principal é o aluno utilizar o programa para estudar, então acho que uma funcionalidade em que o professor consegue acessar todos os flashcards não seria tão útil no dia a dia. Seria mais fácil um sistema de compartilhar um conjunto específico de flashcards, por exemplo. Imagine que você faz uma espécie de "baralho de 1ª guerra mundial" então todos os flashcards desse conjunto trabalham com esse tema e aí vc pode compartilhar só esse conjunto com alguém que poderia ser o professor.

12:34

A screenshot of a WhatsApp chat interface. The background is dark with a pattern of small, light-colored icons. The chat history shows several messages. At the top, a green bubble (sender) says "Oi Bruno, boa tarde." followed by "...". Below it, a grey bubble (receiver) says "Entendi, muito obrigado" with a timestamp of 12:39 and a double checkmark. Then, another green bubble (sender) asks a question about predefined flashcards, also with a timestamp of 12:39 and a double checkmark. Below that, a grey bubble (receiver) explains the idea of a customizable flashcard system, with a timestamp of 12:40. This is followed by another grey bubble (receiver) noting that it doesn't need to be pre-programmed, with a timestamp of 12:41. At the bottom, a green bubble (sender) says "Perfeito! Muito obrigado" with a timestamp of 12:44 and a double checkmark. A small white checkmark icon is visible in the bottom right corner of the chat area.

+55 11 99579-2377
Oi Bruno, boa tarde.
...

Entendi, muito obrigado 12:39 ✓✓

Outra dúvida que temos é em relação às matérias. Nós iremos inseri-las ou elas já estarão predefinidas? E, se estiverem predefinidas, quais são essas matérias? 12:39 ✓✓

+55 11 99579-2377
A ideia é que o aluno possa fazer de qualquer matéria, flashcards são uma metodologia de estudos, mas o importante é ser customizável, ele dar o nome para os baralhos, ele conseguir classificar o nível das perguntas etc. 12:40

Note que não é necessário vir programado até porque isso limitaria o uso. Em tese ele pode usar flashcards para memorizar coisas simples. A chave é ser customizável 12:41

Perfeito! Muito obrigado 12:44 ✓✓

A screenshot of a WhatsApp chat interface, continuing the conversation. At the top, a date separator shows "4/17/2025". Below it, a system message in a grey bubble states: "As mensagens e ligações são protegidas com a criptografia de ponta a ponta. Somente as pessoas que fazem parte da conversa podem ler, ouvir e compartilhar o conteúdo dessas conversas. Clique para saber mais." Another date separator shows "5/4/2025". The chat continues with a green bubble (sender) saying "Boa noite Cassiano, tudo bem?" (18:13 ✓✓), followed by a grey bubble (receiver) asking about requirements (18:13 ✓✓), and then a green bubble (sender) asking about difficulty and challenge modes (18:13 ✓✓). A date separator shows "5/5/2025". The chat continues with a grey bubble (receiver) saying "Oi bom dia, tudo bem? A dificuldade deve existir o aluno ao criar o flashcard classifica o nível de dificuldade da pergunta e o modo desafio não é necessário, mas seria interessante. Pense em um modo em que são sorteadas apenas questões difíceis" (10:59). This is followed by a green bubble (sender) saying "Por exemplo" (10:59), and then a grey bubble (receiver) saying "Mas o grupo pode pensar outras ideias de modo desafio" (10:59). At the bottom, a green bubble (sender) says "Perfeito então, era bem isso o que tínhamos em mente" (11:12 ✓✓), and finally a green bubble (sender) says "Muito obrigado!" (11:12 ✓✓).

4/17/2025

As mensagens e ligações são protegidas com a criptografia de ponta a ponta. Somente as pessoas que fazem parte da conversa podem ler, ouvir e compartilhar o conteúdo dessas conversas. Clique para saber mais.

5/4/2025

Boa noite Cassiano, tudo bem? 18:13 ✓✓

Meu grupo estava com uma dúvida em relação a um dos requisitos 18:13 ✓✓

Sobre os flashcards, eles precisam obrigatoriamente ter um modo de dificuldade e um modo de desafio? E se sim, como esses modos iriam funcionar? 18:13 ✓✓

5/5/2025

Oi bom dia, tudo bem?
A dificuldade deve existir o aluno ao criar o flashcard classifica o nível de dificuldade da pergunta e o modo desafio não é necessário, mas seria interessante. Pense em um modo em que são sorteadas apenas questões difíceis 10:59

Por exemplo 10:59

Mas o grupo pode pensar outras ideias de modo desafio 10:59

Perfeito então, era bem isso o que tínhamos em mente 11:12 ✓✓

Muito obrigado! 11:12 ✓✓

8. Referências

- Listar as fontes utilizadas na documentação conforme normas ABNT.

Courier New e Arial

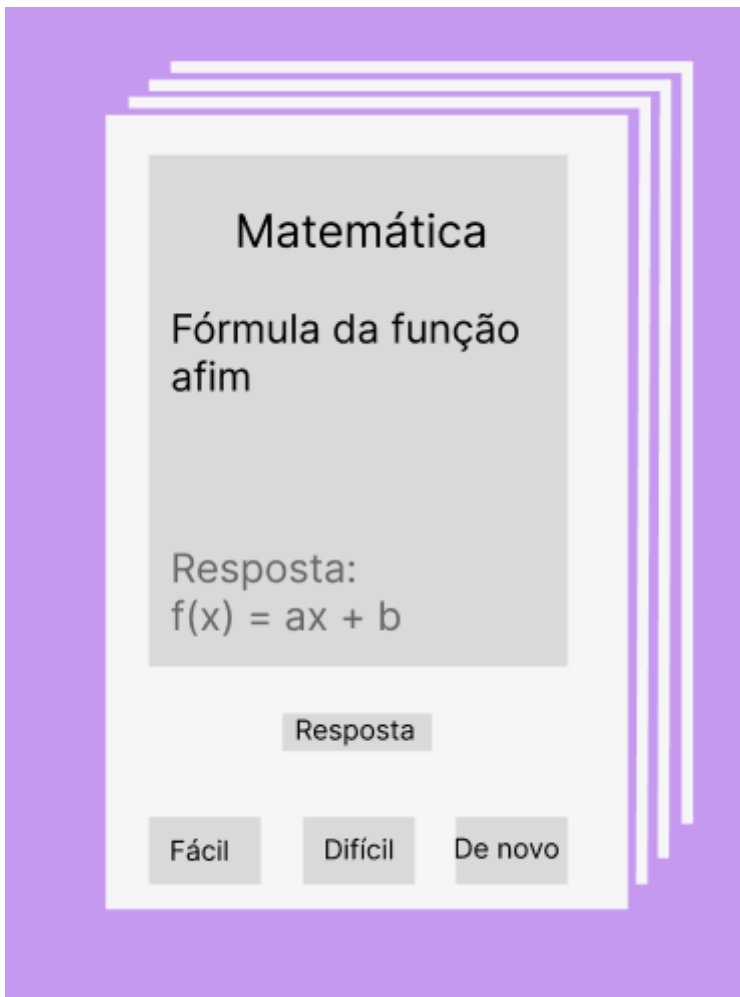
9. Apêndice I

- Registro das reuniões com o parceiro.
- Fotos, documentos de apoio e materiais relevantes.



Materiais:

https://drive.google.com/drive/folders/1kDQDO1kT4C4K7ddSwz5Cp555yDLyl_8h?usp=sharing



Site exemplo:

<http://ahoef.co/blog/2016/01/05/flashcard-app.html>

Figma: [Link](#)

- 1 - Prioridade
- 2 - Média prioridade
- 3 - Pouca prioridade
- 4 - baixa prioridade

perguntas:

Um modelo tradicional de flashcard (anki) possui níveis de dificuldade que calculam quando um determinado card aparecerá novamente (2 dias, 1 mês...). Deverá ser implementado esse sistema?

(2) ☒

Foi informado nos requisitos que o sistema deve informar se o aluno acertou ou errou. Então deverá ter uma caixa de resposta do aluno? Ou alternativas? (1 - o aluno que verificará se acertou ou não)

* Não seria melhor o aluno avaliar se ele acertou ou não? E não o sistema verificar? Se o cliente preferir o sistema, sugerir criar em modelo de alternativas ✓

Os flashcards devem ser pessoais ou compartilhados com os outros alunos? (pessoais)

* O que vocês acham de, no mesmo aplicativo, criamos 2 sessões.

1. O aluno terá os flashcard compartilhados com o restante dos alunos feito pelo professor

2. O aluno conseguirá criar seus próprios flashcard em uma sessão pessoal ✓

Quem irá criar os flashcards serão os alunos ou o professor? ✓

Haverá um cadastro/login para os alunos e o professor? Caso haja um cadastro ao professor, ele terá alguma superioridade que os alunos? Ele conseguirá fazer alguma ação a mais? (1- criar cadastro/login para o aluno primeiro) ✓

Quais são as matérias que irão ser abordadas? O aluno, o professor ou ambos podem conseguir adicionar matérias (crud)? ✓