

DESENVOLVIMENTO DO APLICATIVO PORTAL DO ALUNO - UNI-FACEF

Pedro Henrique Galetti Pires

Graduando em Sistemas de Informação - Uni-FACEF

pedrogaletti12@gmail.com

Thales Henrique Joazeiro

Graduando em Sistemas de Informação - Uni-FACEF

thalesjoazeiro@gmail.com

Me. Ely Fernando do Prado

Mestre em Ciência da Computação - Uni-FACEF

elyfprado@facef.br

RESUMO

Com o surgimento de novas tecnologias e ideias que trazem a possibilidade de melhoria, surgem novos objetivos e desafios. E este projeto teve como objetivo entregar aos alunos do Centro Universitário Municipal de Franca (Uni-FACEF) um novo Portal do Aluno, com o aprimoramento da interface, usabilidade, segurança, velocidade e performance da aplicação, e contendo a implementação de novas funções como: a visualização de notas do curso de medicina; a opção ‘fale conosco’ que possui informações para contato junto à universidade; e também um sistema de autenticação agregado a persistência de token limitado por tempo. Este artigo apresenta técnicas utilizadas no desenvolvimento de aplicações computacionais móveis, junto aos processos de documentação e modelagem de software.

Palavras-chave: Aprimoramento. Interface. Usabilidade. Segurança. Velocidade. Performance.

ABSTRACT

With the emergence of new technologies and ideas that bring the possibility of improvement, new objectives and challenges arise. And this project aimed to deliver to the students of the Centro Universitário Municipal de Franca (Uni-FACEF) a new Student Portal, starting with the improvement of the interface, usability, security, speed, and performance of the application, and containing the implementation of new functions such as: viewing medical school notes; the 'contact us' option that has contact information with the university; and also an authentication system added to time-limited token persistence. This article presents techniques used in the development of mobile computing applications, along with the documentation and software modeling processes.

Keywords: *Enhancement. Interface. Usability. Safety. Velocity. Performance.*

1 INTRODUÇÃO

Este trabalho de conclusão de curso possui como tema e referência, a aplicação das técnicas relacionadas ao curso de sistemas de informação, e a gestão de alunos e disciplinas, para o desenvolvimento de uma aplicação móvel.

O antigo aplicativo do Portal do Aluno da Uni-FACEF disponibilizava aos usuários a possibilidade de acesso à informações como notas, faltas e biblioteca, porém, por possuir um layout pouco usual e não contar com uma performance correspondente às tecnologias disponíveis atualmente, fez com que surgisse a seguinte questão: é possível realizar a refatoração do aplicativo, junto a implementação de novas funções e utilizando novas tecnologias? A partir de então, nos reunimos para tratar do assunto e propor melhorias que viriam a ser realizadas, como a visualização de notas do curso de medicina; a opção 'fale conosco' contendo informações para contato junto à universidade; e também um sistema de autenticação adjunto a persistência de token limitado por tempo, fazendo com que não seja necessário que o usuário passe pelo processo de login novamente, até que o token se expire. Utilizando tecnologias atuais, pode-se aprimorar a interface, usabilidade, segurança, velocidade e performance da aplicação.

Levando em consideração o problema proposto, o objetivo deste trabalho é desenvolver um aplicativo para que o usuário (aluno) tenha uma melhor experiência na utilização do Portal do Aluno, e que lhe seja proporcionado rapidez na busca por dados, e fluidez de informações, sem que sejam necessários muitos cliques.

A partir de uma análise do atual aplicativo do Portal do Aluno, foi observado que se tratava de um layout pouco usual e não correspondente à performance desejada, levando em consideração o avanço que a tecnologia teve com o passar do tempo. Foi aí que surgiu a necessidade de realizar a refatoração do mesmo, juntamente a implementação de novas tarefas. Para que com poucos cliques, o usuário tenha acesso à biblioteca, notas/faltas, financeiro, fale conosco e mais informações, com a tranquilidade de saber que seus dados serão mantidos em segurança.

Para o processo de desenvolvimento foram utilizadas diversas ferramentas tecnológicas. Na gerência do projeto foi empregada a metodologia Kanban, para designar, dividir e organizar as tarefas, fazendo com que fossem definidos e seguidos os prazos para conclusão de cada uma delas.

Na documentação foram usadas ferramentas para gerar o BPMN (*Business Process Model and Notation*), Diagramas UML (*Unified Modeling Language*) e Matrizes de Rastreabilidade.

Pensando na interface e usabilidade do aplicativo, demos andamento à prototipação de telas, onde foi utilizada a ferramenta Adobe XD, que deixou mais dinâmicos os processos de criação, desde alguns simples wireframes, até a definição das telas finais do projeto.

O React Native foi a linguagem de programação utilizada no desenvolvimento do aplicativo, o mesmo também conta com sistemas de segurança de autenticação e a persistência de token limitado por tempo, fazendo com que os dados dos usuários(alunos) possam ser mantidos em segurança.

Para o estudo e desenvolvimento do trabalho, na seção dois são apresentados os conceitos teóricos a cerca do tema proposto.

Na seção três foi abordado quais foram os objetivos do projeto feito, visando sempre a comodidade do usuário (aluno).

Na seção quatro foi apresentado todas as motivações para que esse trabalho fosse realizado, a fim de que o usuário tenha praticidade em seu uso.

Na seção cinco foi feita uma análise do que foi apresentado no artigo, o que foi usado para o desenvolvimento do trabalho e a apresentação do resultado obtido.

2 REFERENCIAL TEÓRICO

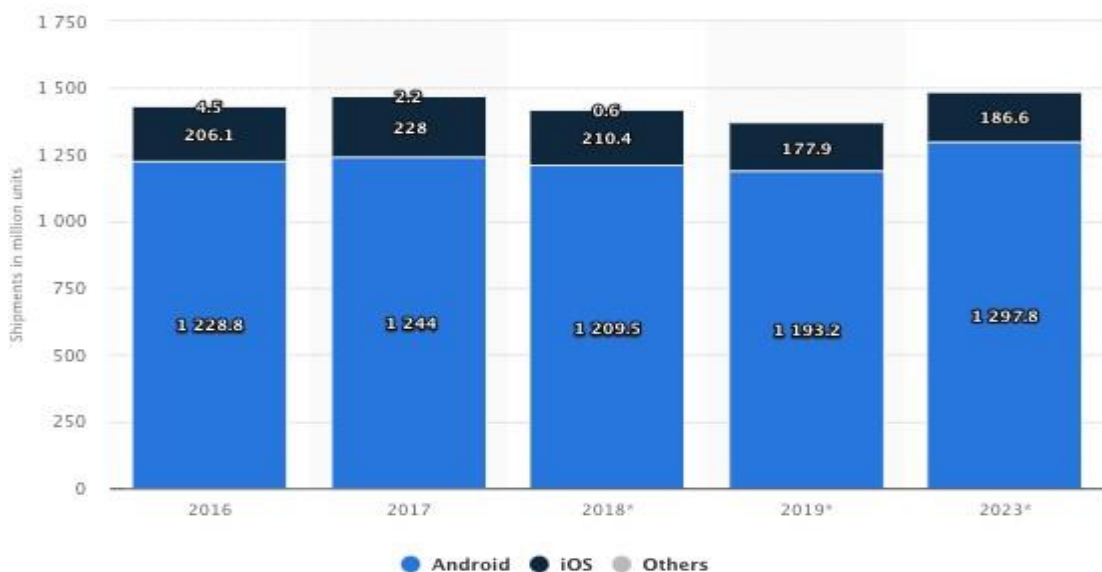
Esta seção aborda temas relacionados aos sistemas de informação utilizados no desenvolvimento de aplicações móveis, sistemas operacionais e à tecnologia de desenvimento React Native.

2.1 ANDROID

Android é uma plataforma de software que permite criar aplicativos para dispositivos móveis, como smartphones e tablets, oferecendo um conjunto completo de softwares: um sistema operacional, middleware e aplicações chave. Seu desenvolvimento foi baseado no Kernel do Linux, no qual o código fonte open source tem a finalidade de possibilitar que a comunidade sempre incorpore novas funcionalidades e tecnologias, com intenção de sempre inovar as aplicações móveis (OLIVEIRA, 2020).

O Gráfico 01 a seguir ilustra as remessas globais de unidades de smartphones por sistema operacional 2016-2023, tendo como principais os sistemas operacionais móveis Android e IOS por serem os mais utilizados na atualidade.

Gráfico 01 - Remessas globais de unidades de smartphones por sistema operacional de 2016 a 2023.



Fonte: IDC (2019, online) .

Segundo O'DEA (2020), uma razão para o domínio do Android como mencionado

anteriormente, é sua plataforma de código aberto que permite que os fabricantes de aparelhos móveis usem e adaptem o sistema operacional para seus próprios dispositivos, algo muito realizado pelas líderes mundiais como Samsung e Huawei.

2.2 IOS

Desenvolvido e patenteado pela Apple, Iphone Operation System (IOS) é o sistema operacional móvel mais avançado do mundo. Apesar de não ser o mais vendido e mais utilizado, por seus concorrentes terem preços de vendas abaixo dos aparelhos móveis que a Apple produz, e seu código fonte ser fechado (O'DEA, 2020).

Segundo HOLST (2019), um sistema operacional de código fechado proíbe modificações; o desenvolvedor original do software tem controle total.

O IOS é um sistema operacional para iPhone, iPod, iPad e a sua primeira aparição foi com o lançamento da primeira geração de iPhone em 2008 revolucionando a maneira como os telefones celulares seriam criados a partir deste momento (MORISSEY e CAMPBELL, 2011).

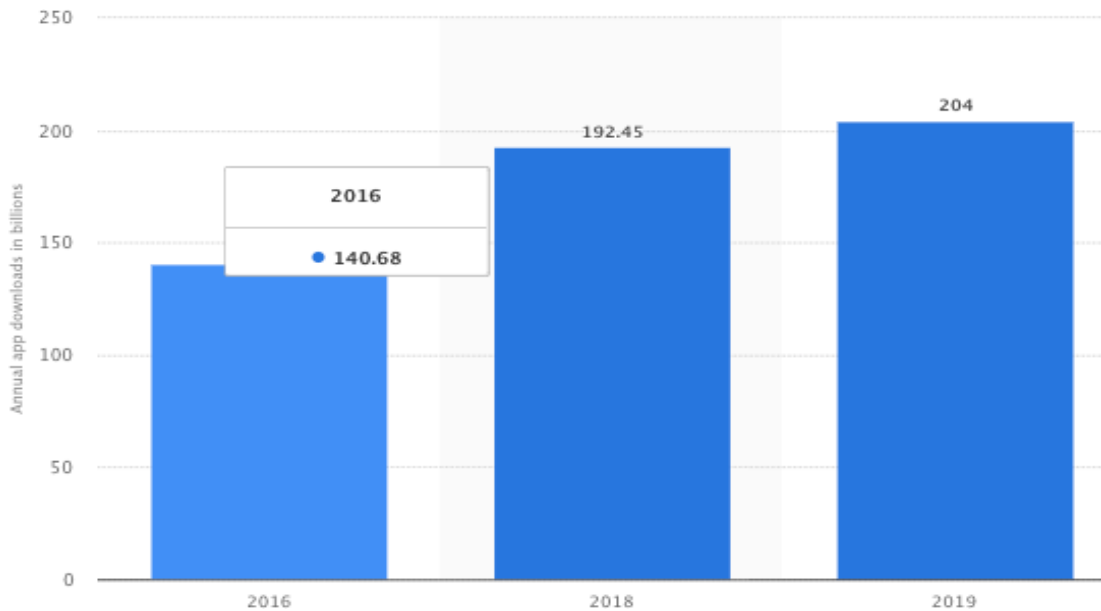
Segundo APPLE (2020), o IOS está na versão 13, foi desenvolvido para que a experiência do usuário seja de profundo contato com as tecnologias de pontas suportadas pelos aparelhos móveis, contando com o ARKit 3, Reality Composer e RealityKit para experiências de realidade aumentada. O Core ML 3 e o novo aplicativo Create ML ajudam a fornecer experiências personalizadas na utilização de sistemas, de maneira rápida e fácil com o aprendizado de máquina no dispositivo, líder do setor.

2.3 APLICAÇÕES MÓVEIS

Segundo CLEMENT (2019), aplicativos móveis, comumente chamados de aplicativos, são softwares desenvolvidos para uso em dispositivos móveis sem fio, como smartphones e tablets.

Os aplicativos são projetados com as limitações e os recursos dos dispositivos móveis em mente. Por exemplo, um jogo pode usar o acelerômetro de um smartphone ou um aplicativo de bloco de desenho pode usar a caneta de um tablet. Em comparação com os sistemas de software integrados em computadores, cada aplicativo móvel geralmente oferece uma funcionalidade específica.

Gráfico 02 - Número de downloads de aplicativos para dispositivos móveis em todo o mundo de 2016 a 2019.



Fonte: (CLEMENT, 2020).

No Gráfico 02 podemos analisar a quantidade de downloads realizados de aplicativos móveis em todo o mundo de 2016 a 2019.

2.4 REACT NATIVE

Na conferência React.js em 2015, o Facebook apresentou seu novo framework React Native, um framework que pensaram que revolucionaria a maneira como os aplicativos móveis são criados. Quando o React Native foi lançado, havia apenas suporte para o IOS, mas desde então o suporte para o Android foi adicionado e ainda está em expansão.

O principal objetivo do React Native é simples, o desenvolvedor não precisa gastar tempo supérfluo para criar um aplicativo móvel, pois não seria necessário que fossem codificados duas aplicações sendo uma para IOS e outra para Android. Cada plataforma conta com sua aparência, suas funcionalidades, seus recursos, sendo assim não é possível que um único aplicativo seja adequadamente instalado em diferentes plataformas.

No entanto, como é a interface gráfica que difere, o desenvolvimento pode se basear no mesmo idioma, mas ter os gráficos renderizados de maneira diferente, dependendo da plataforma de destino e se tornarem componentes nativos reais.

Abordagem chamada de “aprenda uma vez, escreva em qualquer lugar” pelo Facebook, que descreve o que é o React native, citação realizada na conferência React.js 2015

(ALPERT, 2015).

O React Native é executado em uma instância incorporada do JavaScriptCore (iOS) e V8 (Android) dentro dos aplicativos, e renderizada para componentes específicos da plataforma de nível superior. Os componentes JavaScript são declarados usando um conjunto de primitivas internas suportadas pelo IOS e pelos Componentes Android (DANIELSSON, 2016).

3 EMPREENDEDORISMO

O conceito de empreendedorismo vem sendo moldado e ganhando novas definições com o passar do tempo. Ser um empreendedor exige não somente criatividade e inovação, e sim uma ampla combinação de fatores pessoais, incluindo a forma em que pensamos, agimos e nos comunicamos.

Empreendedorismo é a capacidade que uma pessoa tem de identificar problemas e oportunidades, desenvolver soluções e investir recursos na criação de algo positivo para a sociedade. Pode ser um negócio, um projeto ou mesmo um movimento que gere mudanças reais e impacto no cotidiano das pessoas. (BUENO, 2019)

3.1 STARTUPS

Startup é um termo utilizado atualmente, voltado ao empreendedorismo e geralmente usado quando relacionado à pequenas e novas empresas no ramo de tecnologia, porém abrange todo e qualquer tipo de negócio desde que o mesmo utilize da inovação como meio de inserção no mercado.

Startup é uma empresa jovem com um modelo de negócios repetível e escalável, em um cenário de incertezas e soluções a serem desenvolvidas. Embora não se limite apenas a negócios digitais, uma startup necessita de inovação para não ser considerada uma empresa de modelo tradicional. (BICUDO, 2016)

3.2 CANVAS

O Business Model Canvas, mais conhecido apenas como Canvas é uma metodologia desenvolvida por volta dos anos 2000 por Alex Osterwalder e Yves Pingneur. O Canvas nos trás a possibilidade de criar modelos de negócios a partir dos nove elementos que toda empresa ou organização possui, descrevendo o modelo de negócio da organização em questão. A Figura 1 exhibe o modelo Canvas elaborado para este projeto.

Figura 1 - Modelo Canvas do Projeto



Fonte: Os autores.

3.2.2 DETALHAMENTO DO CANVAS DO PROJETO

Abaixo serão listados e detalhados os 9 tópicos que compõem o modelo Canvas do projeto aqui apresentado, junto a explicação sobre o que foi definido em cada um e o motivo para que isto acontecesse.

- 1) Proposta de valor: é o que a empresa vai oferecer ao mercado, aquilo que irá agregar valor e levar benefícios aos seus clientes ou usuários. E então seguindo este aspecto, nosso projeto oferece ao usuário a possibilidade de uma navegação com mais facilidade, rapidez, acessibilidade e segurança, para que estejam conectados à universidade independente de onde se localize, sendo necessário apenas o acesso à internet.
- 2) Segmento de mercado: tem como objetivo definir quais grupos de pessoas e organizações que uma empresa deseja atingir. Nosso projeto foi desenvolvido com o intuito de que todos os estudantes do Centro Universitário de Franca – Uni-FACEF contassem com um acesso facilitado, sendo necessário apenas um dispositivo mobile e acesso à internet.

- 3) Canais: os canais são os meios pelos quais o cliente ou usuário pode obter nosso aplicativo, e este foi disponibilizado nas lojas de aplicativos mais conceituais do mercado.
- 4) Relação com o cliente: É como a empresa se relaciona com os clientes, em nosso caso o usuário pode acessar a aba *Fale Conosco* que está disponível na tela de menu do aplicativo, e lá terá acesso aos contatos dos responsáveis pelo funcionamento do mesmo.
- 5) Atividade chave: são ações ou atividades essenciais que as empresas realizam para assim conseguirem manter o seu respectivo modelo de negócio. Neste projeto a principal atividade é o desenvolvimento da aplicação mobile, visto que o aplicativo só será disponibilizado á este modelo de plataforma.
- 6) Recursos chave: são recursos fundamentais para que o projeto possua características originais e exclusivas, onde então possa se diferenciar dos demais. Em nossa plataforma de auxílio ao aluno, trouxemos como diferenciais coisas simples, porém com extrema importância, dentre elas proporcionar ao usuário segurança, rapidez, acessibilidade, contando também com um design intuitivo.
- 7) Parceiros chave: são muitas vezes responsáveis por alguma atividade chave da empresa. Em nosso projeto temos como parceiros chave a Google com a Play Store, e a Apple com a Apple Store, ambos efetivando a hospedagem da aplicação e a disponibilizando aos usuários.
- 8) Fontes de renda: são meios para se obter uma renda, de acordo com o modelo de negócio proposto. Nossa aplicação foi desenvolvida especificamente aos alunos do Centro Universitário de Franca – Uni-FACEF tendo em vista que a faculdade irá arcar com todos os custos.
- 9) Estrutura de custos: nesta parte são descritos todos os custos relevantes e de máxima importância quando relacionados à estrutura e funcionamento da aplicação. Gastos com

infraestrutura e equipe são imprescindíveis, contudo, não se pode esquecer o investimento necessário para que a plataforma esteja disponível todos os dias.

4 ENGENHARIA DE SOFTWARE

A engenharia de software abrange um processo, um conjunto de métodos (práticas) e um leque de ferramentas que possibilitam aos profissionais desenvolverem software de altíssima qualidade (PRESSMAN, 2011).

Cria-se software para computadores da mesma forma que qualquer produto bem-sucedido: aplicando-se um processo adaptável e ágil que conduza a um resultado de alta qualidade, atendendo às necessidades daqueles que usarão o produto. Aplica-se uma abordagem de engenharia de software (PRESSMAN, 2011).

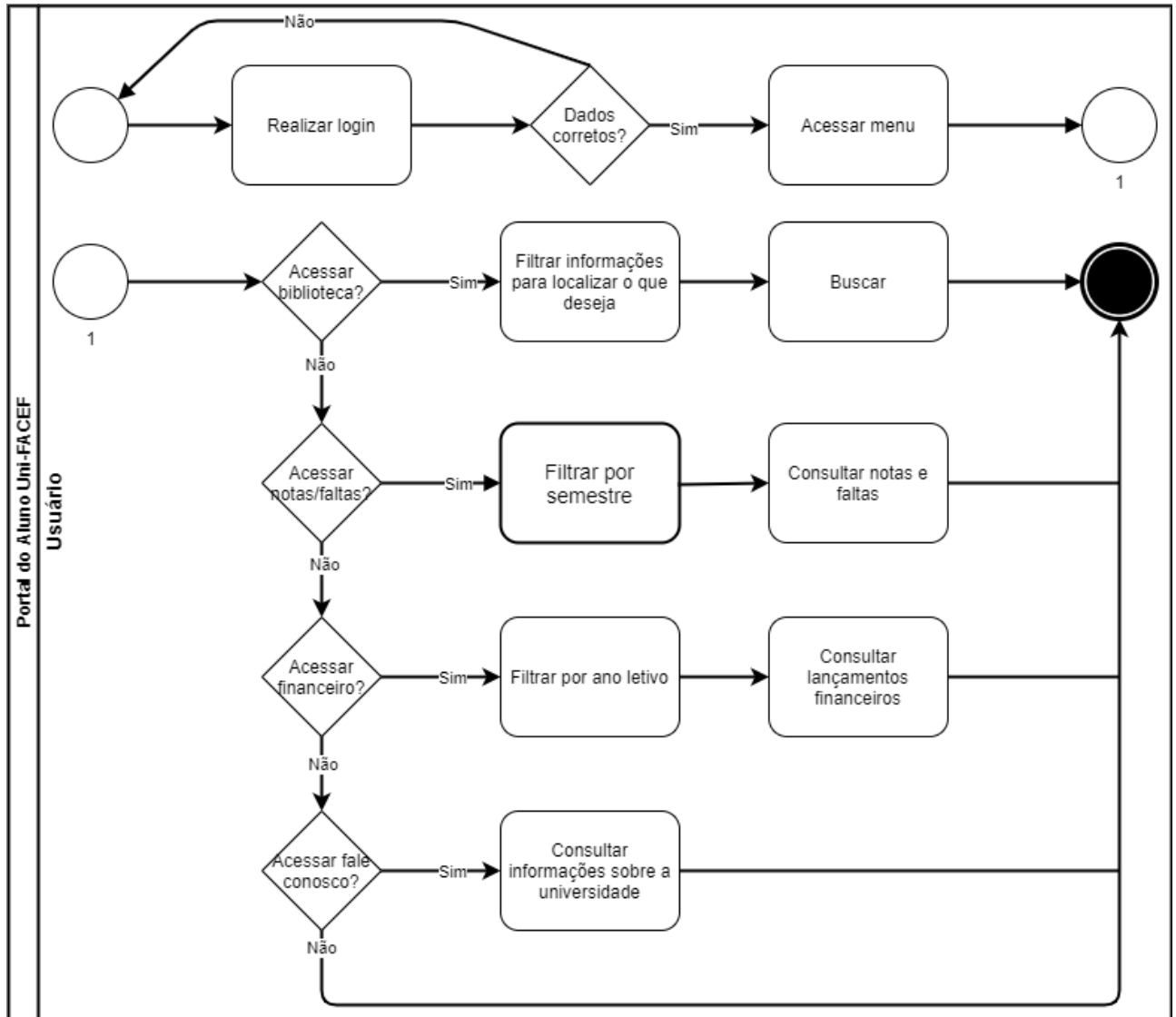
A seguir são demonstradas as documentações dos processos realizados no desenvolvimento deste projeto, assim como os diagramas BPMN (*Business Process Model and Notation*) e Diagrama de Caso de Uso, os quais facilitam o entendimento de todas as atividades da solução, e auxiliam em possíveis alterações ou correções, caso a aplicação venha à necessitar.

4.1 BPMN

BPMN é uma notação para modelagem de processos de negócio. Em outras palavras, o BPMN estabelece um padrão para representar os processos graficamente, por meio de diagramas. Esse padrão possui um conjunto de símbolos e regras que permite modelar diferentes fluxos de processos, com vários níveis de detalhamento (ALMEIDA, 2017).

O BPMN do projeto Portal do Aluno – Uni-FACEF, apresentado na Figura 2, representa todo o fluxo de processos da aplicação, desde a autenticação do usuário até a consulta de notas e faltas, financeiro, biblioteca e outras funcionalidades.

Figura 2 - BPMN



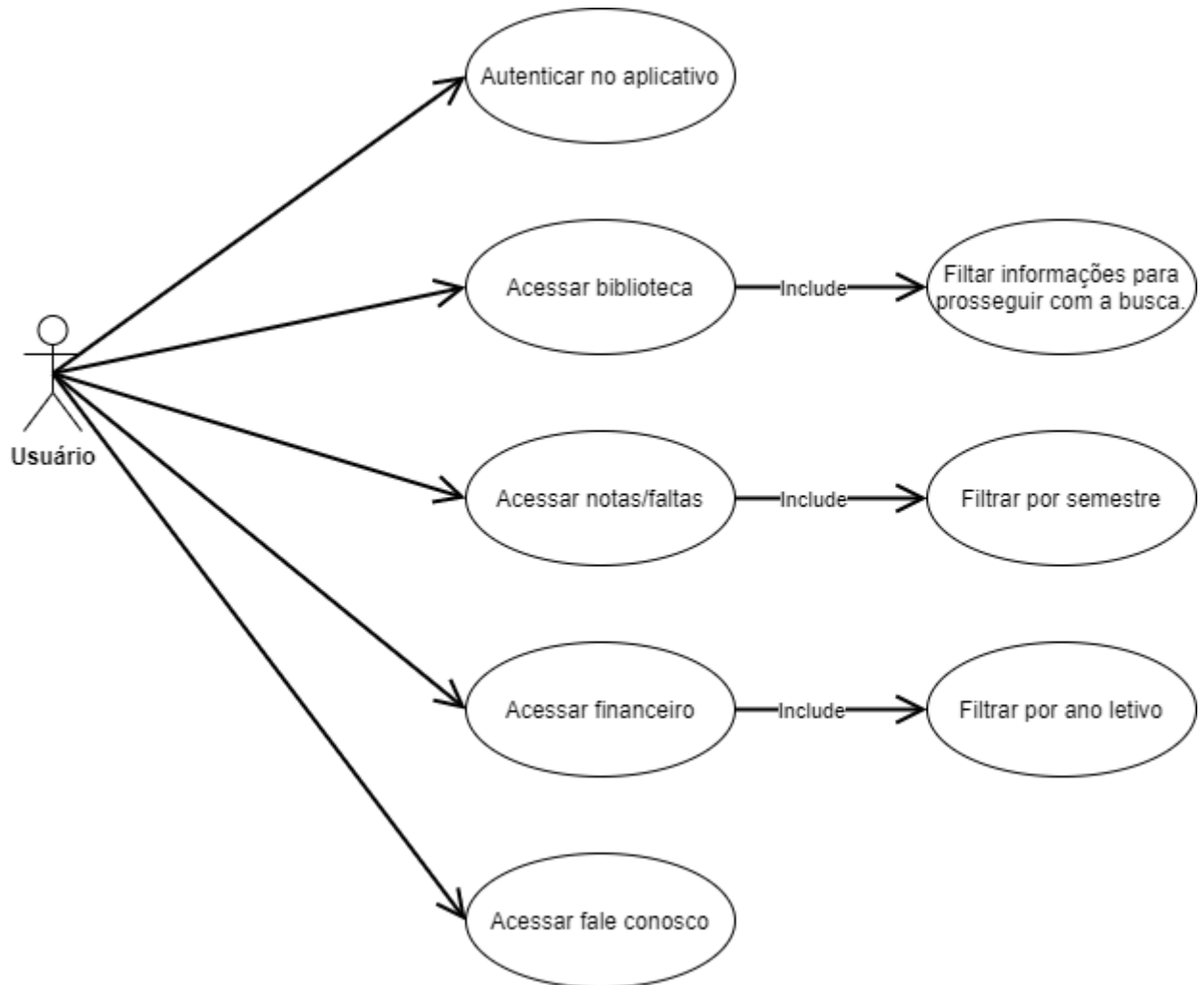
Fonte: Os autores.

4.2 DIAGRAMA DE CASO DE USO

Na Linguagem de modelagem unificada (UML), o Diagrama de Caso de Uso resume os detalhes dos usuários do seu sistema (também conhecidos como atores) e as interações deles com o sistema (LUCIDCHART, 2020).

O Diagrama de Caso de Uso do projeto Portal do Aluno – Uni-FACEF, apresentado na Figura 3, representa os fluxos de interações do usuário junto à aplicação.

Figura 3 - Diagrama de Caso de Uso



Fonte: Os autores.

5 DESENVOLVIMENTO

Um processo de desenvolvimento de software pode ser visto como um conjunto de atividades organizadas, usadas para definir, desenvolver, testar e manter um software (PINTO, 2007).

Nesta seção será retratado o desenvolvimento do aplicativo *mobile* do Portal do Aluno - Uni- FACEF e sua implementação, contando também com a prototipação das telas do projeto e com uma breve descrição do fluxo de cada uma delas.

5.1 IMPLEMENTAÇÃO DO FRONT-END

Para o desenvolvimento da aplicação móvel foi utilizada a linguagem de programação React Native.

A Figura 4 apresenta a configuração de endpoint principal da aplicação, ou seja, a url em que as requisições serão feitas (linha 3). Entre as linhas 5 e 28 é feita uma exportação de configuração do cliente do projeto Front-End, utilizamos da biblioteca “axios” para controlar as requisições, com uma configuração básica, passando ‘baseUrl’ que no caso é nossa url base para requisições, e responseType que diz que desejamos receber respostas em JSON. Nas “options” de maneira resumida, estamos configurando na linha 15 que o retorno com sucesso irá nos retornar a resposta da api para o front-end, na linha 16 iniciamos as configurações dos erros, o que acontece entre as linhas 17 e 23 é a verificação de autenticação do usuário, se ele não estiver autenticado o mesmo é deslogado do aplicativo.

Figura 4 - Fragmento da implementação da aplicação mobile

```
1  import axios from 'react-native-axios';
2
3  const DEVELOP_HOST = 'http://sga.unifacef.com.br/';
4
5  export default {
6    default: {
7      client: axios.create({
8        baseUrl: DEVELOP_HOST,
9        responseType: 'json',
10     }),
11     options: {
12       returnRejectedPromiseOnError: true,
13       response: [
14         {
15           success: (store, response) => response,
16           error: (store, error) => {
17             if ([error.response && error.response.status === 401]) {
18               return AsyncStorage.multiRemove(['@unifacef/user', '@unifacef/token'])
19                 .then(() => reset('Login'));
20             }
21             console.log('error', error);
22             return Promise.reject(error);
23           }
24         }
25       ]
26     },
27   },
28 };
29
```

Fonte: Os autores.

5.1.1 CONFIGURAÇÃO XML PARA JSON

A Figura 5 apresenta a configuração de conversão de XML para JSON e também a configuração para o envio das requisições para o back-end.

O back-end utilizado no projeto foi criado pela EddyData em WebServices/SOAP requisições, ou seja, o que é retornado nas respostas dos WebServices é um Envelope XML (Linha 8 à 15).

A seguir, as funções utilizadas nesta etapa de desenvolvimento:

toSoapArgs: Esta função recebe os parâmetros para criar os argumentos do envelope;

CreateEnvelope: esta função serve para criar um envelop que será enviado para o back-end;

Headers: função para setar os Headers das requisições;

getIsMatriculaValidaReturn: função que tem como objetivo checar se a matrícula do usuário é válida, pois a cada requisição é verificada, para que caso o aluno esteja logado e a matricula dele não seja mais válida o mesmo seja deslogado do sistema;

xmlToJson: com o auxilio de uma biblioteca chamada “x2js”, o “xmlToJson é responsável por retornar a resposta da requisição (xmlData) no formato JSON para que consigamos utilizá-la dentro da aplicação na linguagem JavaScript.

Figura 5 - Implementação das funções de configuração XML para JSON

```
5  const toSoapArgs = params => params.map((item, index) => `<arg${index}>${item}</arg${index}>`).join('');
6
7  const createEnvelope = (method, body) => {
8    <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
9      <S:Header/>
10     <S:Body>
11       <ns2:${method} xmlns:ns2="http://ws.eddydata.com.br/"
12         ${body}
13       </ns2:${method}>
14     </S:Body>
15   </S:Envelope>
16 }
17
18 const headers = urlMethod => ({
19   'Content-Type': 'text/xml; charset=utf-8',
20   SOAPAction: `http://sga.unifacel.com.br/${urlMethod}`,
21 });
22
23 const getIsMatriculaValidaReturn = xml => Number(RegExp(/<return>([0-9]+)</return>/gi.exec(xml)[1]);
24
25 const xmlToJson = (xmlData, method) => x2js.xml2js(xmlData).Envelope.Body[`${method}Response`].return;
```

Fonte: Os autores.

5.1.2 UTILIZAÇÃO DAS FUNÇÕES DE CONFIGURAÇÕES

É apresentado na Figura 6 uma requisição sendo realizada com as funções da figura 5, e todas elas são realizadas através do método POST. Para fazer as requisições utilizamos o auxílio da biblioteca `redux`¹, `redux-thunk`, `redux-middleware`, entre outras que trabalham e organizam os dados dentro da aplicação.

Na linha 14 nós passamos os headers para a requisição, olhando para a figura 5 na linha 20, identificamos que esse header é referente a ação que queremos executar no back-end.

Já na linha 15 chamamos a função “`createEnvelope`” para criarmos nosso envelope e assim mandar os dados para o back-end. Notamos também o uso da função “`toSoapArgs`”, para passar o “`semester`” para o envelope sendo nosso argumento. Da linha 18 em diante é nossa tratativa para caso de erro na requisição.

Figura 6 - Exemplo de requisição utilizando funções da configuração.

```
6 export function getNotes(semester) {
7   return (dispatch) => {
8     dispatch({
9       type: types.GET_NOTES,
10      payload: {
11        request: {
12          url: "NotaWS/NotaWS",
13          method: "POST",
14          headers: headers("NotaWS/getNotasPorMatricula"),
15          data: createEnvelope("getNotasPorMatricula", toSoapArgs([semester])),
16        },
17      },
18    }).catch(() => {
19      return dispatch(
20        showVisibilityModal({
21          visibility: true,
22          text:
23            "Opa! Não conseguimos recuperar suas notas neste momento. Tente novamente, por favor.",
24          buttonText1: "Ok!",
25        })
26      );
27    });
28  };
29 }
```

Fonte: Os autores.

¹ <https://redux.js.org/>

5.1.3 CHAMADA DA FUNÇÃO XMLTOJSON

É apresentado na Figura 7 o reducer do Container “Notes”, trabalhando com o Redux em que temos o seguinte processo: Action > Reducer > Index. Primeiramente é feita a requisição como na figura 6, o resultado da requisição nos retorna um tipo de action, ou seja, como a requisição é feita pelo type: “types.GET_NOTES” o resultado dessa action será: “GET_NOTES_SUCCESS” ou “GET_NOTES_FAIL”, isso se dá pela configuração da lib redux.

Sendo assim no nosso arquivo reducer.js esperamos receber o success, já que no “catch” da figura 6 nós tratamos o type “GET_NOTES_FAIL”. Quando então é retornado sucesso da requisição nós pegamos o “data” retornado da requisição (xml) passando para a função “xmlToJson” e ela nos retorna o data que antes era XML em JSON.

Figura 7 - Exemplo da chamada da função xmlToJson.

```
20 const reducer = handleActions(  
21   {  
22     [beginLoading]: state => ({  
23       ...state,  
24       isLoading: true,  
25     }},  
26     [stopLoading]: state => ({  
27       ...state,  
28       isLoading: false,  
29     }},  
30     [types.GET_NOTES_MEDICINE_SUCCESS]: (state, { payload: { data } }) => ({  
31       ...state,  
32       notes: xmlToJson(data, 'getNotasMedicina') || [],  
33     }},  
34     [types.GET_NOTES_SUCCESS]: (state, { payload: { data } }) => ({  
35       ...state,  
36       notes: xmlToJson(data, 'getNotasPorMatricula') || [],  
37     }},  
38     [types.SET_CURRENT_SEMESTER]: (state, { currentSemester }) => ({  
39       ...state,  
40       currentSemester,  
41     }},  
42   },  
43   INITIAL_STATE,  
44 );
```

Fonte: Os autores.

5.1.4 RENDERIZAÇÃO DE VETOR

É apresentado na Figura 8 a renderização dos cards de notas (em referência à tela de Notas e Faltas), o componente “NotesItem” na linha 66 que é responsável por criar os cards nos quais exibem as matérias, notas e faltas do aluno. As informações de cada matéria são passadas para este componente através de uma função nativa do JavaScript chamada “map”, na linha 63. Esta função executa um looping em um Array (Vetor) passando de objeto em objeto, neste caso o nosso vetor está armazenado na variável “notes”.

Figura 8 - Exemplo de renderização de vetor na aplicação.

```
63     {notes.map((item, index) => {
64         if (item.ativo) return (
65             <Card key={index} list margedHorizontal last={index === notes.length - 1}>
66                 <NotesItem
67                     last={index === (notes.length - 1)}
68                     title={toCapitalize(item.nomeDisciplina)}
69                     approved={item.situacao === 'APROVADO'}
70                     cursed={item.situacao === 'CURSANDO' || item.situacao === '-'}
71                     noteOne={item.nota1}
72                     noteTwo={item.nota2}
73                     noteThree={item.nota3}
74                     noteFour={item.nota4}
75                     sub={item.recuperar1}
76                     faultsOne={item.falta1}
77                     faultsTwo={item.falta2}
78                     frequency={item.frequencia}
79                     average={item.media}
80                 />
81             </Card>
82         )
83     }}}
```

Fonte: Os autores.

5.2 IMPLEMENTAÇÃO DO BACK-END

Dado que o projeto foi uma refatoração da aplicação já existente na universidade, a criação do código back-end não foi necessária, isto porque a instituição já tem um trabalho executado com a empresa EddyData localizada na cidade de Franca-SP que toma conta dos processamentos de dados da parte dos alunos.

A empresa é responsável pelos webservices que fornecem as notas e faltas dos alunos tanto de medicina quanto para os demais cursos prestados na instituição, acervo bibliotecário e

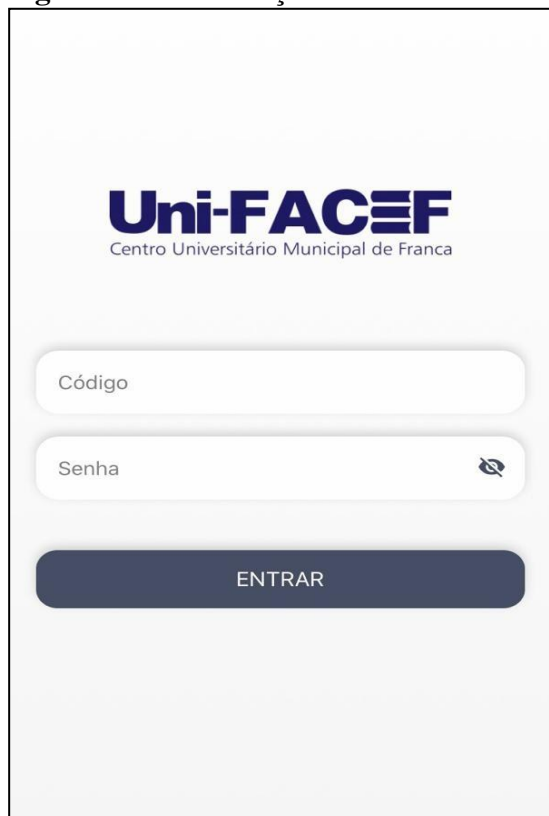
financeiro de cada aluno, dentre outros, porém os citados são os que o Portal do Aluno utiliza para o funcionamento.

6 ANÁLISE DE RESULTADOS

Nesta seção são apresentados os resultados, pela exposição das telas existentes no MVP (*Minimum Viable Product*) da aplicação móvel Portal do Aluno – Uni-FACEF, junto às caracterizações de suas funcionalidades.

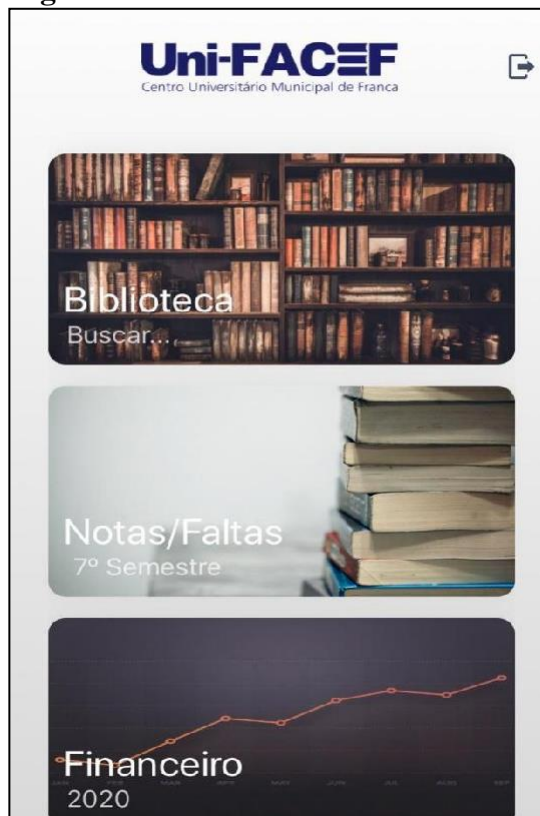
Para que o usuário tenha acesso às funcionalidades da aplicação, é necessário que a universidade (Uni-FACEF) o disponibilize um código e uma senha, e cada aluno recebe os seus, sendo todos divergentes. Já com as informações necessárias em mãos, o usuário deverá as inserir em seus respectivos campos (Figura 9). A figura 9 apresenta o aplicativo logo que iniciado, exibindo ao usuário, sua tela de autenticação.

Figura 9 - Autenticação



Fonte: Os autores.

Figura 10 - Menu



Fonte: Os autores.

Após sua autenticação, o Menu (Figura 10) da aplicação é apresentado ao usuário, e é onde o mesmo terá acesso à todas as funcionalidades do sistema, que consistem em: Consultas de livros na biblioteca virtual (Figura 11), Consulta de notas/faltas (Figura 12), Financeiro (Figura 13), Fale Conosco (Figura 14), e a partir daí poderá seguir a direção em que deseja.

Ao acessar a tela da Biblioteca (Figura 11) o usuário poderá buscar pelo livro ou qualquer que seja o tipo de material (cd, chave, disquete, dvd, fita cassete, monografia, vídeo, vídeo texto) em que necessita, informando seu título, autor ou assunto.

Figura 11 - Biblioteca

The screenshot shows a mobile application interface for a library. At the top, there is a back arrow and the title 'Biblioteca'. Below this is a search bar with a dropdown menu currently showing 'Material'. Underneath the search bar are three input fields labeled 'Título', 'Autor', and 'Assunto'. At the bottom of the search area is a dark blue button with the text 'BUSCAR' in white capital letters.

Fonte: Os autores.

Figura 12 - Notas e Faltas

The screenshot shows a mobile application interface for viewing grades and absences. At the top, there is a back arrow and the title 'Notas'. Below this is a list of courses. Each course entry includes a title, a table of grades and absences, and a frequency percentage.

Projetos em Desenvolvimento			
● Tecnológico e Inovação em Sistemas de Informação I			
Nota 1	Nota 2	Sub.	Média
0.0	0.0	0.0	0.0
Falta 1	Falta 2	Freq.	
6	0	93%	

● Sistemas Integrados I			
Nota 1	Nota 2	Sub.	Média
0.0	0.0	0.0	0.0
Falta 1	Falta 2	Freq.	
2	0	95%	

● Sociedade e Tecnologia			
Nota 1	Nota 2	Sub.	Média
0.0	0.0	0.0	0.0
Falta 1	Falta 2	Freq.	
2	0	95%	

Fonte: Os autores.

Na tela de Notas e Faltas (Figura 12), o aluno poderá consultar suas respectivas notas e faltas em cada disciplina que está cursando, incluindo as dependências caso possua.

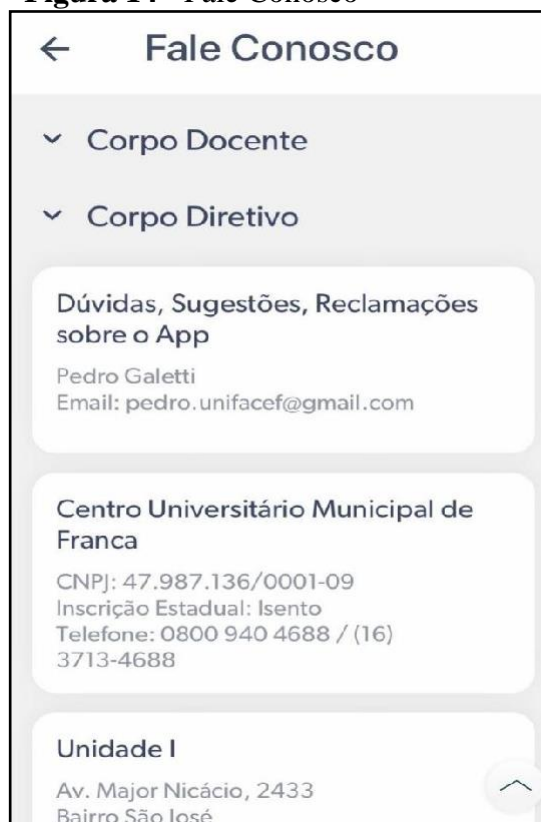
Ao acessar a tela do Financeiro (Figura 13), o aluno possui acesso à todos os lançamentos da faculdade, destinados que são filtrados por ano. Pode-se também observar os status destes lançamentos, tais como: em aberto, quitada, vencida, cancelada, renegociada.

Figura 13 - Financeiro



Fonte: Os autores.

Figura 14 - Fale Conosco



Fonte: Os autores.

A proposta da tela de Fale Conosco (Figura 14), é oferecer ao usuário acesso aos contatos da universidade, e informações referentes ao corpo docente e diretivo da mesma.

6.1 PUBLICAÇÃO E FEEDBACK

Para a realização de testes da aplicação, houve a liberação em fase beta e com entregas parciais a alguns dos alunos da universidade, como intuito de que fossem avaliados os seguintes pontos: usabilidade, consistência, segurança e integridade de dados. Após o desenvolvimento da primeira versão do aplicativo e a fase beta de testes terem sido finalizados, o aplicativo foi então disponibilizado nas lojas virtuais Google Play e App Store.

Na figura 15 são exibidas algumas informações disponibilizadas pela Google Play referentes ao aplicativo. Logo em seguida, na figura 16 são mostrados alguns dados sobre o número de instalações ativas da aplicação, na loja virtual App Store, e na Figura 17 é exibida a nota de avaliação dos usuários de smartphone com sistema operacional Ios. As informações foram coletadas em 16/09/2020.

Levando em consideração que a nota máxima de avaliação das lojas virtuais é 5 pode-se afirmar que os principais objetivos deste trabalho foram alcançados, fornecendo ao usuário final um aplicativo com total satisfação de uso, indicando que a experiência do usuário e interface foram aprovadas por boa parte daqueles que o utilizam, dando esperanças de que com as implementações de novas funcionalidades conseguiremos alcançar melhores resultados. Outro fator que indica uma aceitação positiva por parte dos usuários, é o fato de que cerca de 50% de todos os alunos matriculados nos cursos da Uni-FACEF mantém instalado o aplicativo em seu smartphone.

Empresas de sucesso utilizam o feedback, ou a chamada pesquisa de satisfação, de seus consumidores e clientes para construir pontes sólidas em direção a excelência (BUENO, 2018).

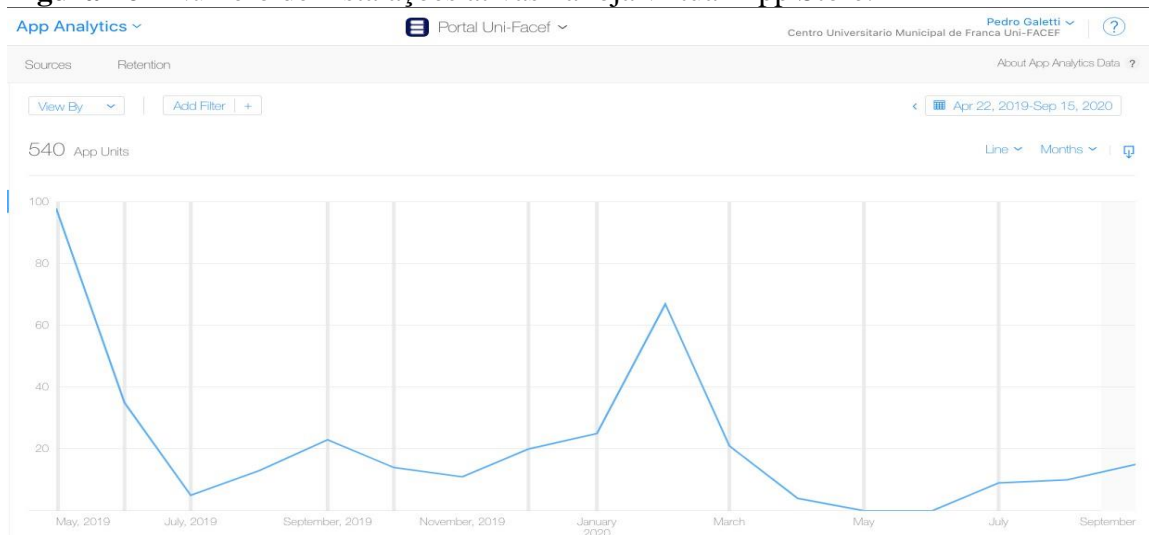
Figura 15 - Informações sobre o aplicativo na loja virtual Google Play.

Nome do app	Instalações ativas	Nota do Google Play	Última atualização	Status
 Portal Uni-Facef com.portalunifacef	531	★ 4,93	29/06/2020	Publicado

Página 1 de 1

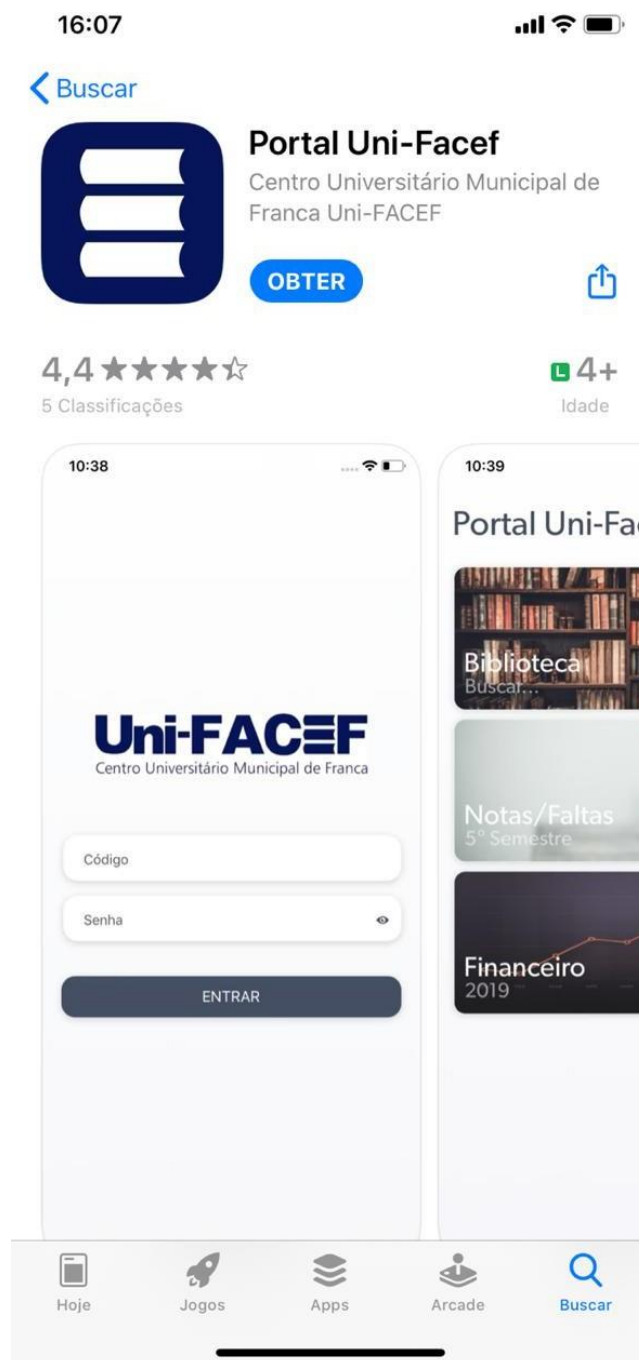
Fonte: Os autores.

Figura 16 - Número de instalações ativas na loja virtual App Store.



Fonte: Os autores.

Figura 17 - Nota de avaliação na loja virtual App Store



Fonte: Os autores.

7 CONSIDERAÇÕES FINAIS

Devido à possibilidade de melhorias na interface, implementação de novas funções e utilização de novas tecnologias, em relação ao antigo aplicativo do Portal do Aluno da Uni-FACEF, surgiu então o desafio de fazer com que os usuários (alunos) possuíssem uma melhor experiência ao utilizar o mesmo, e que lhe fosse proporcionado rapidez na busca por dados e fluidez de informações.

O propósito principal deste trabalho foi alcançado e concluído com sucesso, porém durante toda a trajetória de desenvolvimento foram identificadas algumas dificuldades, uma delas é relacionada à linguagem de programação utilizada, React Native, visto que de 2019 para 2020 haviam sido realizadas diversas atualizações, fazendo com que a aplicação tivesse que ser refatorada para que pudesse ir ao ar com a versão mais atual da linguagem. A usabilidade da aplicação também foi um grande desafio, pois foi necessário a realização de diversos estudos de casos, definições de jornada do usuário e utilização de personas.

Como projeto futuro, será desenvolvida e implementada a função de notificação de postagem de nota, que serão disparadas todas as vezes que algum docente divulgar a nota em questão. Melhorias constantes poderão ser implementadas no aplicativo de acordo com o feedback dos alunos, fazendo com que o app esteja sempre atualizado com as recentes necessidades dos usuários.

Referências

ALMEIDA, V. N. D. **euax**, 2017. Disponível em: <<https://www.euax.com.br/2017/02/o-que-e-bpmn-business-process-model-and-notation/>>. Acesso em: 12 Agosto 2020.

APPLE. **IOS**. [S.l.]: Apple, 2020.

BICUDO, L. **StartSe**, 2016. Disponível em: <<https://www.startse.com/noticia/startups/afinal-o-que-e-uma-startup>>. Acesso em: 12 Agosto 2020.

BUENO, J. R. **SEBRAE**, 2019. Disponível em: <<https://blog.sebrae-sc.com.br/o-que-e-empendedorismo/>>. Acesso em: 12 Agosto 2020.

BUENO, R. Feedback: O que é, Importância e Dicas. **SBCOACHING**, 2018. Disponível em: <<https://www.sbcoaching.com.br/blog/feedback/>>. Acesso em: 10 set. 2020.

CLEMENT, J. Uso de aplicativos para dispositivos móveis - Estatísticas e Fatos. **Statista**, 2019. Disponível em: <<https://www.statista.com/topics/1002/mobile-app-usage/>>. Acesso em: 15 set. 2020.

CLEMENT, J. Número de downloads de aplicativos móveis em todo o mundo de 2016 a 2019. **Statista**, 2020. Disponível em: <<https://www.statista.com/statistics/271644/worldwide-free-and-paid-mobile-app-store-downloads/>>. Acesso em: 15 set. 2020.

HOLST, A. Sistemas operacionais móveis - Estatísticas e Fatos. **Statista**, 2019. Disponível em: <<https://www.statista.com/topics/3778/mobile-operating-systems/>>. Acesso em: 15 ago. 2020.

IDC. [S.l.]: IDC, 2019.

LUCIDCHART. **Lucidchart**, 2020. Acesso em: 12 Agosto 2020.

MORISSEY; CAMPBELL. [S.l.]: [s.n.], 2011.

O'DEA, S. Remessas globais de unidades de smartphones por sistema operacional 2016 - 2023, 2020. Disponível em: <<https://www.statista.com/statistics/309448/global-smartphone-shipments-forecast-operating-system/>>. Acesso em: 10 set. 2020.

OLIVEIRA, V. A. D. J. **Introdução ao Desenvolvimento para Dispositivos Móveis**. 1a. Edição. ed. Porto Alegre: INF - UFRGS, v. 1, 2020.

PINTO, H. L. Atividades básicas ao processo de desenvolvimento de Software. **DevMedia**, 2007. Disponível em: <<https://www.devmedia.com.br/atividades-basicas-ao-processo-de-desenvolvimento-de-software/5413>>. Acesso em: 15 set. 2020.

PRESSMAN, R. S. **Engenharia de Software**. 7. ed. New York: The McGraw-Hill Companies, 2011.